Single Calculus Chain Documentation

Release 5.2.0

Ioannis Binietoglou, Holger Baars, Giuseppe D'Amico

Dec 12, 2022

Contents

Contents:

CHAPTER 1

Introduction

- The Single calculus chain is made of different modules. These modules don't interact directly but only change value in a database.
- This interface will allow Earlinet memebers to interact with parts of the database.
- One part of the interface (the "Station admin" section) permits registering a new station, registerring new lidar systems and configuration, fill in details for the channels that constitute the system and finnaly define the products (extinction, backscatter e.t.c.) that need to be calculated by the SCC.
- The second part of the interface is dedicated to the uploading of new measurement files, the configuration of the measurement specific paramters and, finally, the retrieval of the calculated products.
- Different types of users, with different level of access permissions can have access in the interface. In this way, higher level of flexibility and security can be achived.

CHAPTER 2

Tutorial

2.1 Introduction

- The Single calculus chain is made of different modules. These modules don't interact directly but only change value in a database.
- This interface will allow Earlinet members to interact with parts of the database.
- One part of the interface (the "Station admin" section) permits registering a new station, registering new lidar systems and configuration, fill in details for the channels that constitute the system and finally define the products (extinction, backscatter e.t.c.) that need to be calculated by the SCC.
- The second part of the interface is dedicated to the uploading of new measurement files, the configuration of the measurement specific parameters and, finally, the retrieval of the calculated products.
- Different types of users, with different level of access permissions can have access in the interface. In this way, higher level of flexibility and security can be achieved.

2.2 Adding a station

You can change all your settings throught the *admin section* of the website. To reach it, click on **Station admin** link at the main menu of the site.

Note: You will need to have an account with admin access privileges to access this part of the site. See *User management* for details.

The first you have to do to start using the single calculus chain is to register your station. To do this, go the the admin section and click on the **HOI stations** in the *System settings* panel. This will take you to a page with a list of all stations that you account has access to. This list should be empty if this is the first time you add a station.

To add a new station to the database click on **Add HOI station** at the top right of the screen. This will take you to a new page were you can fill in the needed information. The fields in **bold** are mandatory and you will need to fill them before you can save you new station.

For now you will need to fill in the following fields:

Name The name of the station

Id The Earlinet call sign with exactly 2 characters.

Institute name The name of the institute that own the system

Latitude In degrees north is the latitude of the station.

Longitude In degrees east is the longitude of the station.

Height asl The altitude of the station in meters above sea level.

PI The name of the Principal Investigator of the station.

You can leave all the other fields empty. When you are done, press the **save** button at the bottom right of the page. This will take you back to the list of your stations. If everything went OK your station you just added should appear in the list.

We don't have to make any more changes in this part, so you can click on **Home** on the top left of the page to return to the starting page of the *admin section*.

2.3 Adding a system

After adding a station to the database, we need to add a new system. To do this, click on the **HOI systems** in the *System settings* panel. This will take you to a page with a list of all available systems that are connected with your stations. This list should be empty if this is the first time you add a system.

Note: In the Single Calculus Chain, a *HOI System* represents a specific configuration of a lidar system. For example, if you are operating a lidar system and you use different channels during daytime and nightime, you will need to register *two different* systems in the database, one for each different configuration you use.

To add a new system to the database click on **Add HOI system** at the top right of the screen. This will take you to a new page were you can fill in the needed information. As before, the fields in **bold** are mandatory and you will need to fill them before you can save you new system.

Note: Not every field that is present in the database is used in the in the Single Calculus Chain. Many of them are part of the Handbook of Instruments.

For now you will need to fill in the following fields:

Name The name of your system.

Station (owner) From the drop-down list, select the station which this system belongs to.

Configuration The name of the specific configuration. For example you could specify "night time" if the system you are registering correspond to the night-time configuration of your system

Pi The principle investigator of this system

Height asl The altitude of the system above sea level (in meters).

You can leave all the other fields empty. When you are done, press the **save** button at the bottom right of the page. This will take you back to the list of your systems. If everything went OK your new system you just added should appear in the list.

We don't have to make any more changes in this part, so you can click on **Home** on the top left of the page to return to the starting page of the *admin section*.

2.4 Adding equipment

After adding a system to the database, we need to add at least one telescope and one laser before you add a channel.

2.4.1 Telescope

To add a new telescope, click on the **HOI telescopes** in the *System settings* panel. This will take you to a page with a list of all available telescopes that are connected with your station. This list should be empty if this is the first time you add a telescope.

To add a new telescope to the database click on **Add HOI telescope** at the top right of the screen. This will take you to a new page were you can fill in the needed information. Once again, the fields in **bold** are mandatory and you will need to fill them before you can save you new telescope.

The fields you need to add are:

Type The telescope type

Diameter The diameter of the primary mirror in mm

Focal length The equivalent focal length of the telescope in mm

Full overlap height The height where the full overlap is achieved.

When you are done, press the **save** button at the bottom right of the page. If no errors are present, you will return to the telescope list page. Your new telescope should appear in the list.

2.4.2 Laser

To add a new laser, click on the **HOI Laser** in the *System settings* panel. This will take you to a page with a list of all available lasers that are connected with your station. This list should be empty if this is the first time you add a laser.

To add a new telescope to the database click on **Add HOI laser** at the top right of the screen. This will take you to a new page were you can fill in the needed information. The fields in **bold** are mandatory and you will need to fill them before you can save you new telescope.

The fields you need to add are:

Manufacturer The manufacturer of the laser

Model The model of the laser

Repetition rate The repetition rate in Hz

Type The type of the laser (ex. Nd:YAG)

When you are done, press the **save** button at the bottom right of the page. If no errors are present, you will return to the laser list page. The new laser you added should be present there.

2.5 Adding a channel

After adding a system, a telescope and a laser to the database, you need to add a new channel. To do this, click on the **HOI channels** in the *System settings* panel. This will take you to a page with a list of all available channels that are connected with your lidar systems. This list should be empty if this is the first time you add a system.

To add a new channel to the database click on **Add HOI channel** at the top right of the screen. This will take you to a new page were you can fill in the needed information. As before, the fields in **bold** are mandatory and you will need to fill them before you can save you new system.

The fields you have to fill here are more, as many of these are used during the processing of measurements.

> **Warning:** There is a last step, different from the previous cases, when saving a new channel. You need to connect your channel with a lidar system before you save, or else all your entry will be lost. Read carefully through this document (or directly the *last section*) to avoid any problems.

2.5.1 Fill in the fields

To start using the single calculus chain you will need to fill the following fields:

Name The name of the channel ex. "355", "1064 analog" etc.

Telescope The telescope that is used for this channel

Laser The laser that is used for this channel

Interference filter center The center of the interference filter in nm

Interference filter FWHM The FWHM of the interference filter in nm

Emission wavelength The emission wavelength of the laser used for this channel

Field of view The field of view related to this channel in mrad

Raw range resolution The raw range resolution of the measured data in m

Dead time The dead of the detector in ns. You should fill in this in case of a photon counting detector.

Trigger delay The trigger delay value for the channel in ns. Fill in 0 if not needed.

Scattering mechanism The scattering mechanism that is involved in this channel. Select the appropriate value from the drop-down list.

Dead time correction type The dead time correction type to be applied. Select *Not defined* if none needs to be defined.

Background mode The way to calculate the signal background. Select *Not defined* if none needs to be defined.

Signal type The type of the signal that is measured, ex. "elT" for total elastic, "vrRN2" for vibrational-rotational Raman signal from Nitrogen molecules etc. See *Signal types* for details.

Detection mode The detection mode of this channel.

2.5.2 Connecting to a system

Before you finish, you need to attach your channel to one of your systems. To do this, go at the bottom of the page and select your system from the drop-down list in the **System channels** area.

When you are done, press the **save** button at the bottom right of the page. This will take you back to the list of your channels. If everything went OK your new channel you just added should appear in the list.

2.6 Adding products

After completing all the information related to the your system, we need to specify which product we need to acquire. To do this, click on the **Products** in the *Product settings* panel. This will take you to a page with a list of all available products that are connected with your stations. This list should be empty if this is the first time you add a product.

To add a new product to the database click on **Add Products** at the top right of the screen. This will take you to a new page were you can fill in the needed information. As before, the fields in **bold** are mandatory and you will need to fill them before you can save you new system.

For now you will need to fill in the following fields:

Product type Choose the type of this product. (Available products: Raman backscatter, Extinction only, Combined Raman backscatter and extinction, Elastic backscatter retrieval)

Usecase The use-case number (an integer) based on the descriptions given in the usecase section (ex. 0). See the *usecase list* for details.

Product channels Here you need to specify the channels that are involved in the calculation of the product. If you run out of empty spaces you can add a new blank line by clicking on **Add another Product Channel**.

System products Here you need to specify the system that this product is related to.

Next you will need to fill in various options in the **Product options** table. You will need to fill in the following:

Low range error threshold Specify the maximum acceptable error for altitudes **bellow** 2 km. (from 1% to 100%)

High range error threshold Specify the maximum acceptable error for altitudes **above** 2 km. (from 1% to 100%)

Detection limit The minimum value that your instrument can detect (in $m^{-1}sr^{-1}$ (backscatter) or in m^{-1} (extinction))

Min height The minimum height for for which the product should be calculated (in meters) e.g., 500 m for extinction calculations

Max height The maximum height for for which the product should be calculated (in meters) e.g., 10000 m for extinction calculations

Preprocessing integration time The integrated time of the preprocessed data (in seconds) ex. 900 sec

Preprocessing vertical resolution The vertical resolution of the preprocessed data (in meters) ex. 60 m

Next depending on the product type you selected you have to fill the values in the tables below.

When you are done, press the **save** button at the bottom right of the page. This will take you back to the list of your products. If everything went OK the new product you just added should appear in the list.

We don't have to make any more changes in this part, so you can click on **Home** on the top left of the page to return to the starting page of the *admin section*.

2.7 Processing data

Walk-through of how to upload a file and seeing the results.

CHAPTER 3

Station administration

Contents:

3.1 Overview

3.2 Stations

Note: You cannot delete a station that has associated systems. This is to prevent you from accidentally deleting all the systems you have set up. If you really want to delete a station, you will need to delete the related systems first, or move them to another station.

You can add the definition of new systems that belong to the station by clicking on the Hoi System blue line that appears bellow the main station fields. For more details on the filed you need to fill in see the *Systems* section. You can add more stations by clicking on the "Add another Hoi System" option.

Note: You need to have *Javascript* enabled to add a new station from this page.

3.3 Systems

3.3.1 Deleting systems

You cannot delete a system if it has either *channels* or *products* connected to it.

3.4 Channels

3.4.1 Signal types

(explain here all the signal type abbreviations).

3.4.2 Connection channels to systems

Each channels should be connected to at least one system.

3.5 Products

There are two ways that a product can be related to a system: *directly* or *indirectly*.

A **directly** connected product (or **primary** product) is linked to a system. When a measurement is processed, a file will be created containing the retrieved quantities of this product.

A **indirectly** connected product (or **secondary** product) is used in the definition of a composite, more complex, product. For example, an "extinction" product can be used to define a "lidar ratio and extinction" product. In this case the "lidar ratio and extinction" product is the *primary* product (directly linked to a system), and the "extinction" product is the *secondary* product (only liked to the "lidar ratio and extinction" product). The output of a secondary product is not stored in a file.

Warning: You should avoid linking a product both *directly* and *indirectly* to the same system. If you link a *secondary* product directly to a system, the output of the SCC could be wrong, as two products could attempt to write their output on the same file.

In the product list view, these information are summarized in two columns, labeled "Directly connected" and "Parent products".

Directly connected Will be green for **primary** products, i.e. if the product is directly linked to a system.

Parent products If the product is a **secondary** product (i.e. it is part of a composite product) this column will have a links to the related *primary* products.

Note: **For administrators**: These two columns can be used to quickly spot "orphan" products, i.e. products that are neither linked to a system nor to a composite product. These products will not be available to any user to use, and should be either connected to a system, or deleted. Orphan products will have *red* in the *Directly connected* column and "-" in the *Parent products* column.

3.5.1 Adding a product

Depending on the product you need to produce, some of the following tables need to be filled.

Note: This section is not up-to-date to the latest interface version.

Note: Each product should be connected to a system, either directly or indirectly, before being saved.

Backscatter products

The following tables need to be filed for the various backscatter products.

Backscatter calibrations

For backscatter (product types: Raman backscatter, Combined Raman backscatter and extinction, Elastic backscatter retrieval

LowestHeight in meters, lowest height for reference height search

TopHeight in meters, maximum height for reference height search

WindowWidth in meters, width of the reference height interval

calValue backscatter ratio, e.g. 1 or 1.01

calRangeSearchMethod ID: always 0

If done, you have to remember the ID for your backscatter calibration options an go the the specific tables below,

for product type 0 and 2: Raman_backscatter_options

Raman calculation method at the moment, only 1

bsc cal options insert ID from previous step

error method ID: at the moment, only 1

for product type 3: elast_backscatter_options

elast bsc method ID at the moment, only 1 (iterative approach)

bsc cal options insert ID from previous step

error method ID: at the moment, only 1

lr input id: at the moment, only 1 (fixed LR)

fixed_lr … in sr, value of the used lidar ratio

iter_bsc_options if, iterative approach is used, go to table iter_bsc_options and remember ID and insert here

iter_bsc_options

iter conv crit …e.g. 0.1 (=10%) The iteration is stopped when the RELATIVE difference between the actual and the previous column integrated backscatter coefficients is below this value

ram bsc method id at the moment, only 1

max iteration count: maximum number of iteration steps

for product type 1: Extinction options

Extinction method: Standard: 1, (0 weighted linear fit, 1 non-weighted linear fit, 2 difference quotient, 3 polynomial sec order fit, 4 quadratic function, 5 Savitzky-Golay filter, 6 Russo)

error method ID: at the moment, only 1

overlap file ID: if overlap file is available, enter ID here

angstroem: used angström value for the extinction calculation, e.g. 0,1, or 1.5

for product type 2: Ext bsc options

product ID ID of product type 2

extinction option product ID ID of the extinction product (see table products)

raman backscatter options product ID ID of the Raman backscatter product (see table products)

error method ID: at the moment, only 1

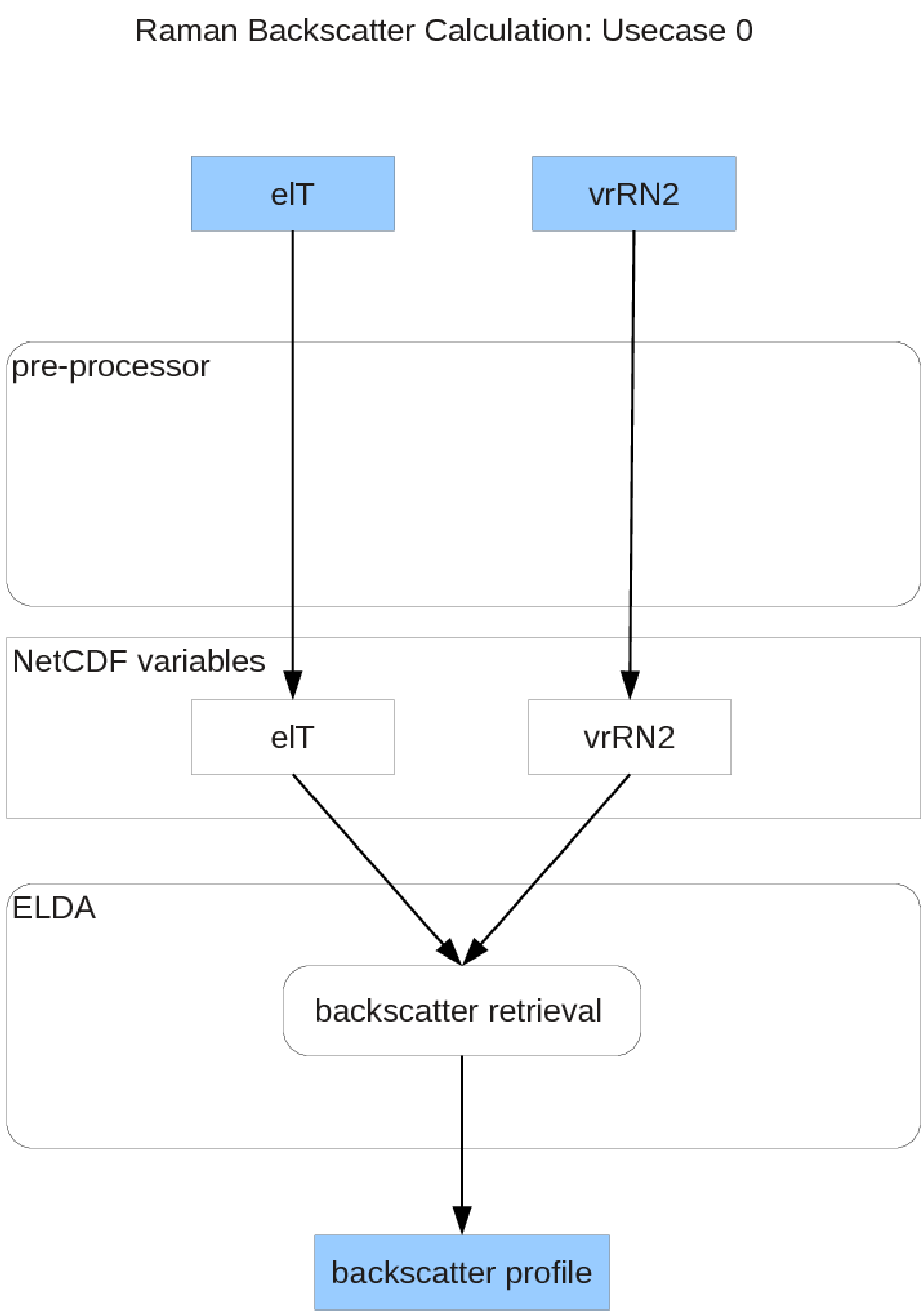
3.6 Usecases

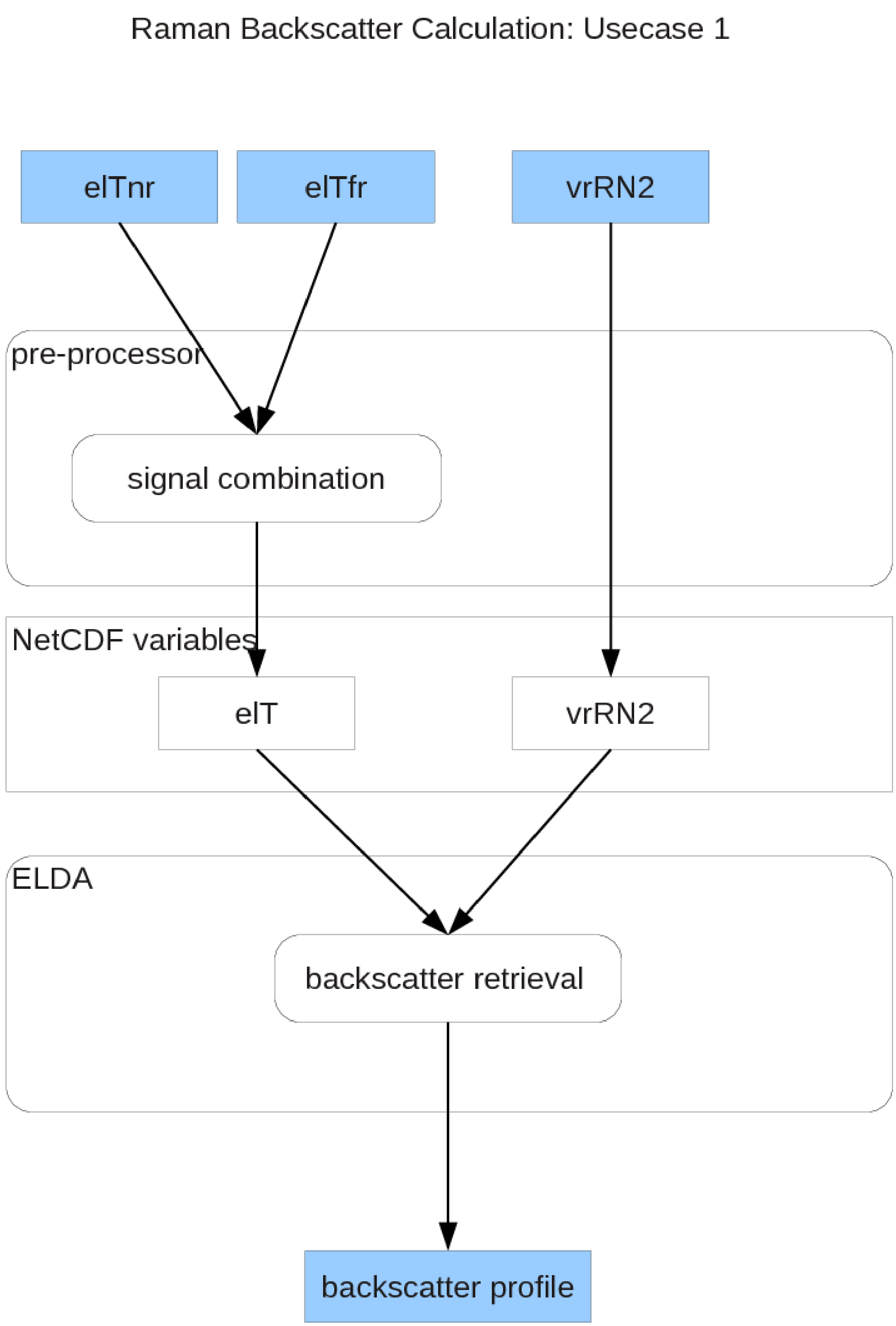
Each usecase corresponds to a way to analyze lidar data.The SCC support the following usecases.

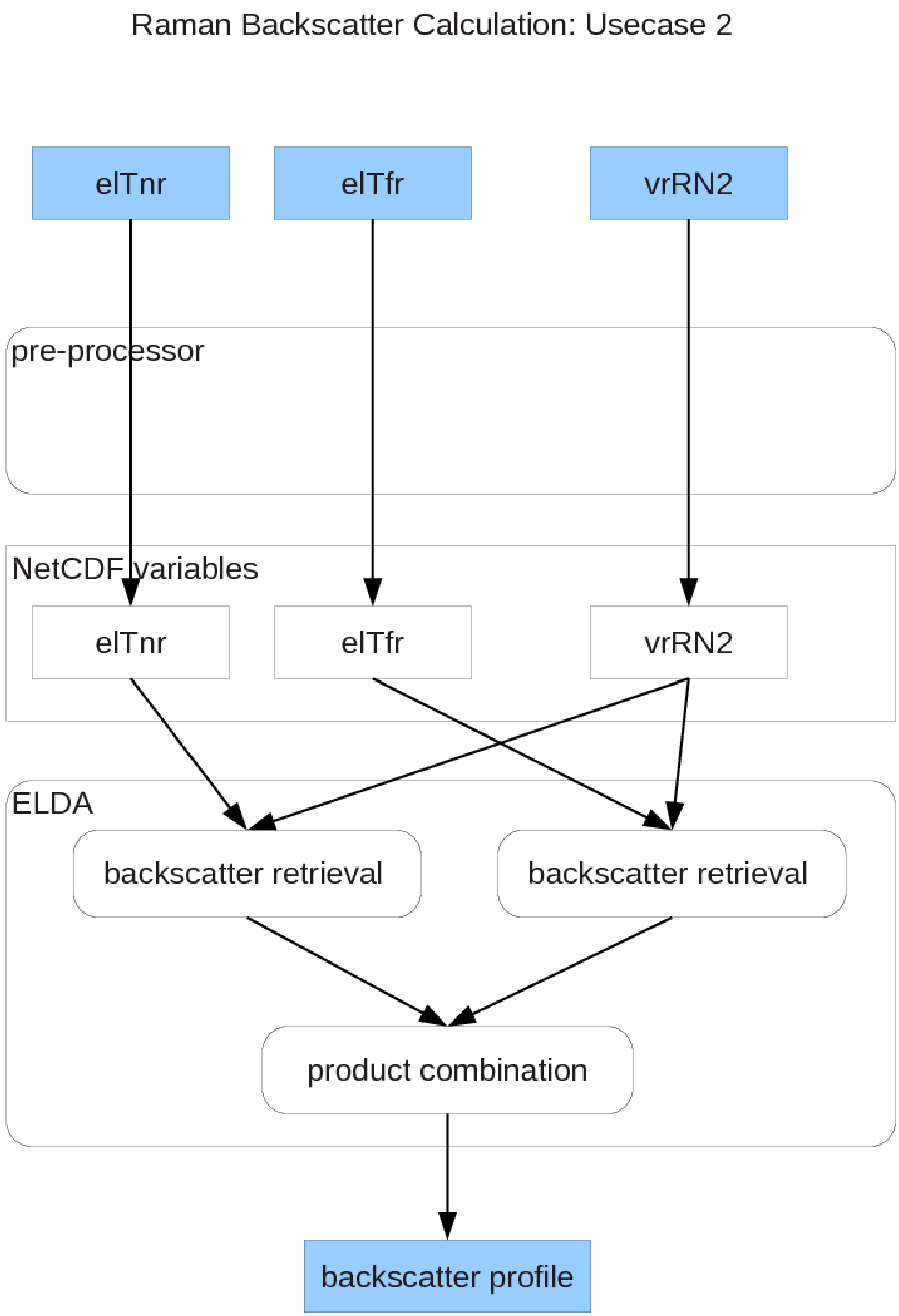
Product types

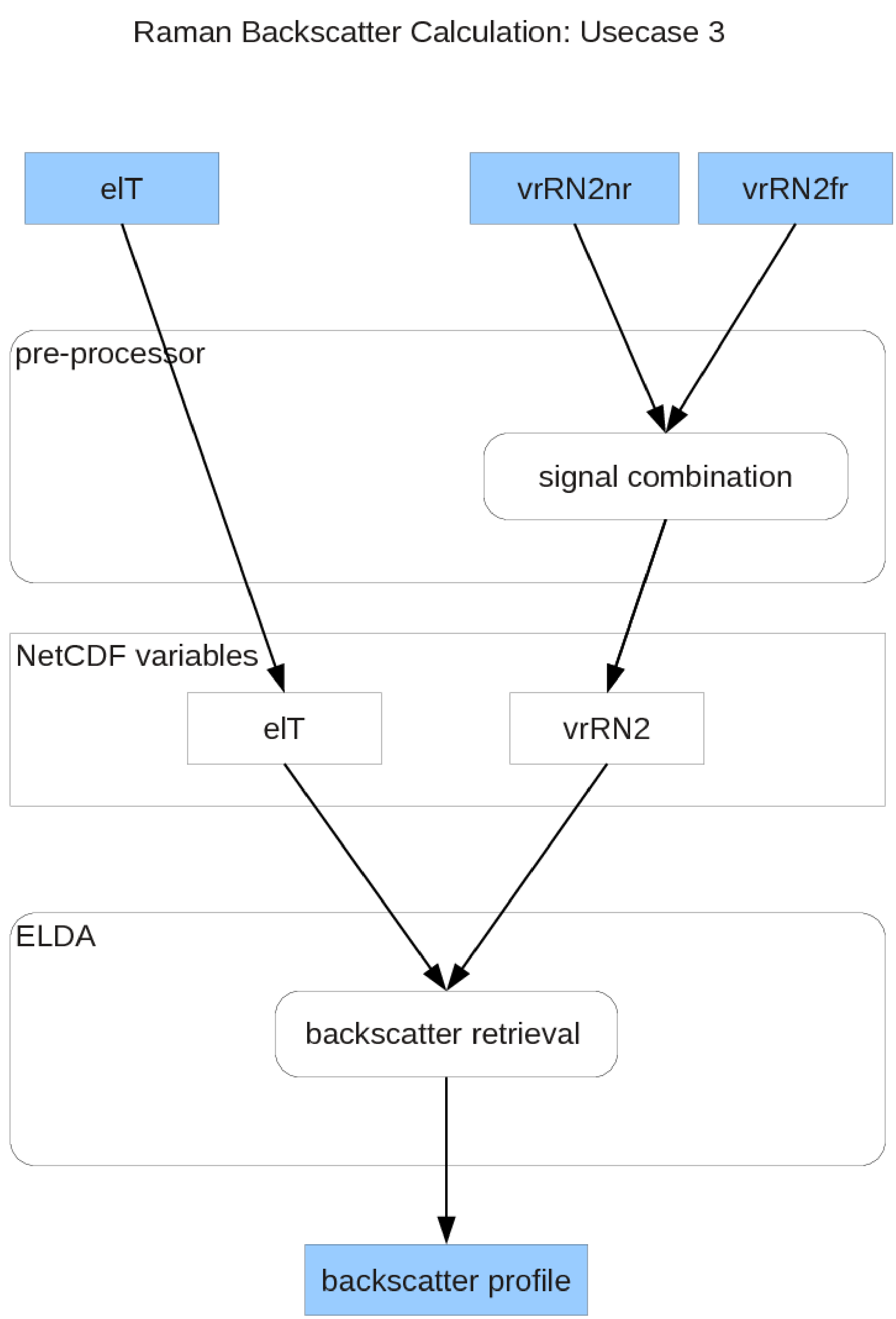
- *Raman backscatter*
- *Raman extinction*
- *Elastic backscatter*
- *Raman backscatter and depolarization*
- *Elastic backscatter and depolarization*
- *Depolarization calibration*

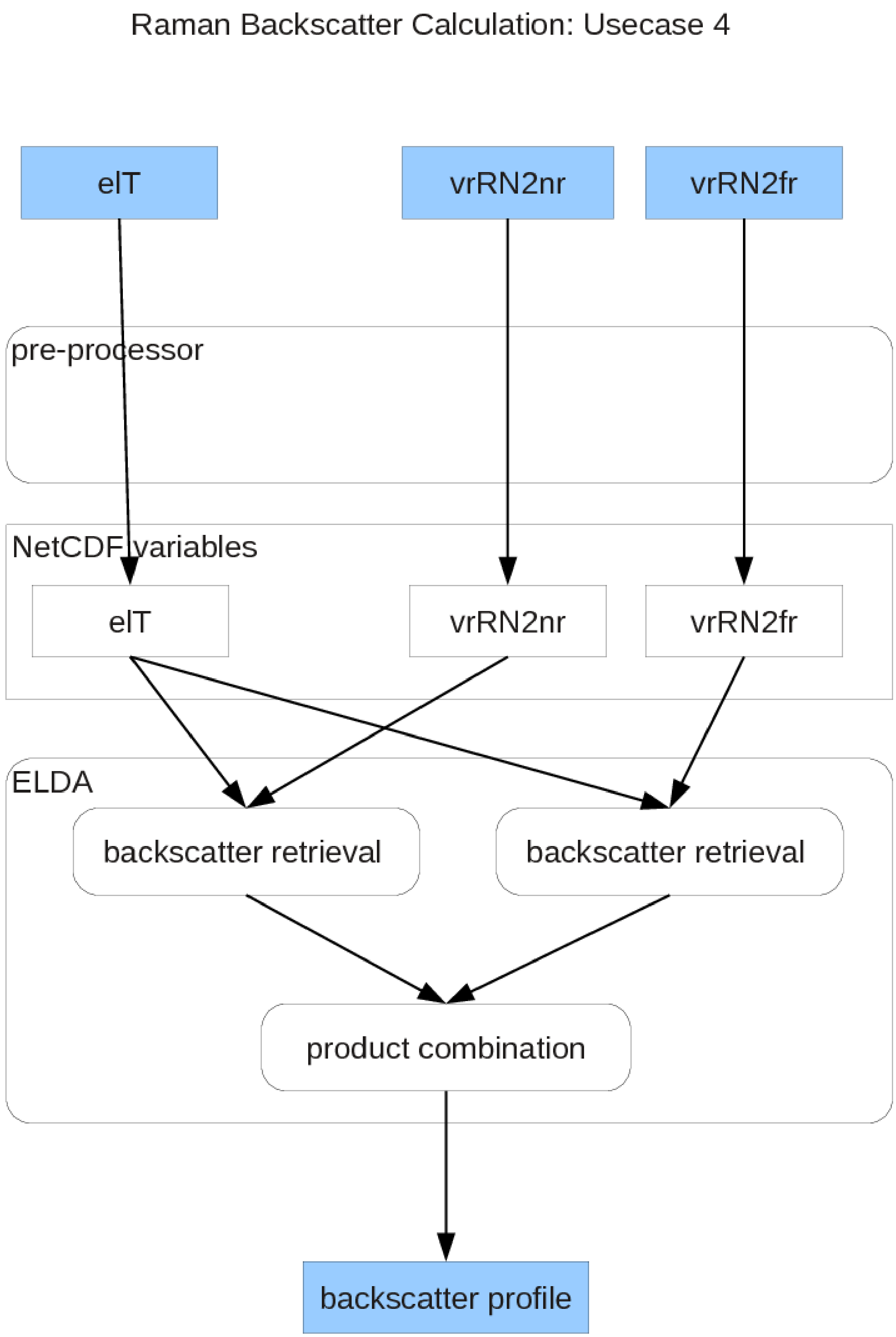
3.6.1 Raman backscatter

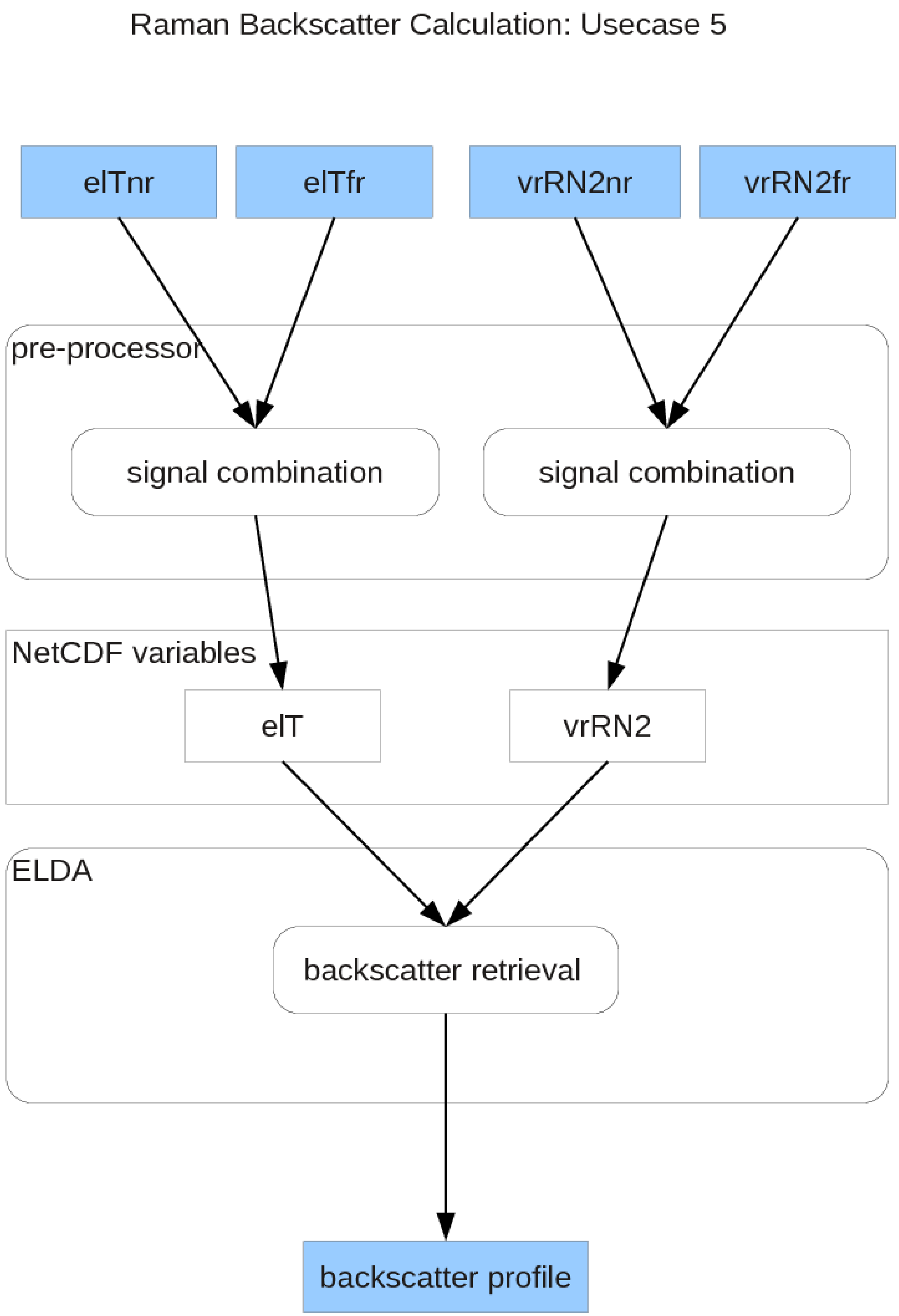


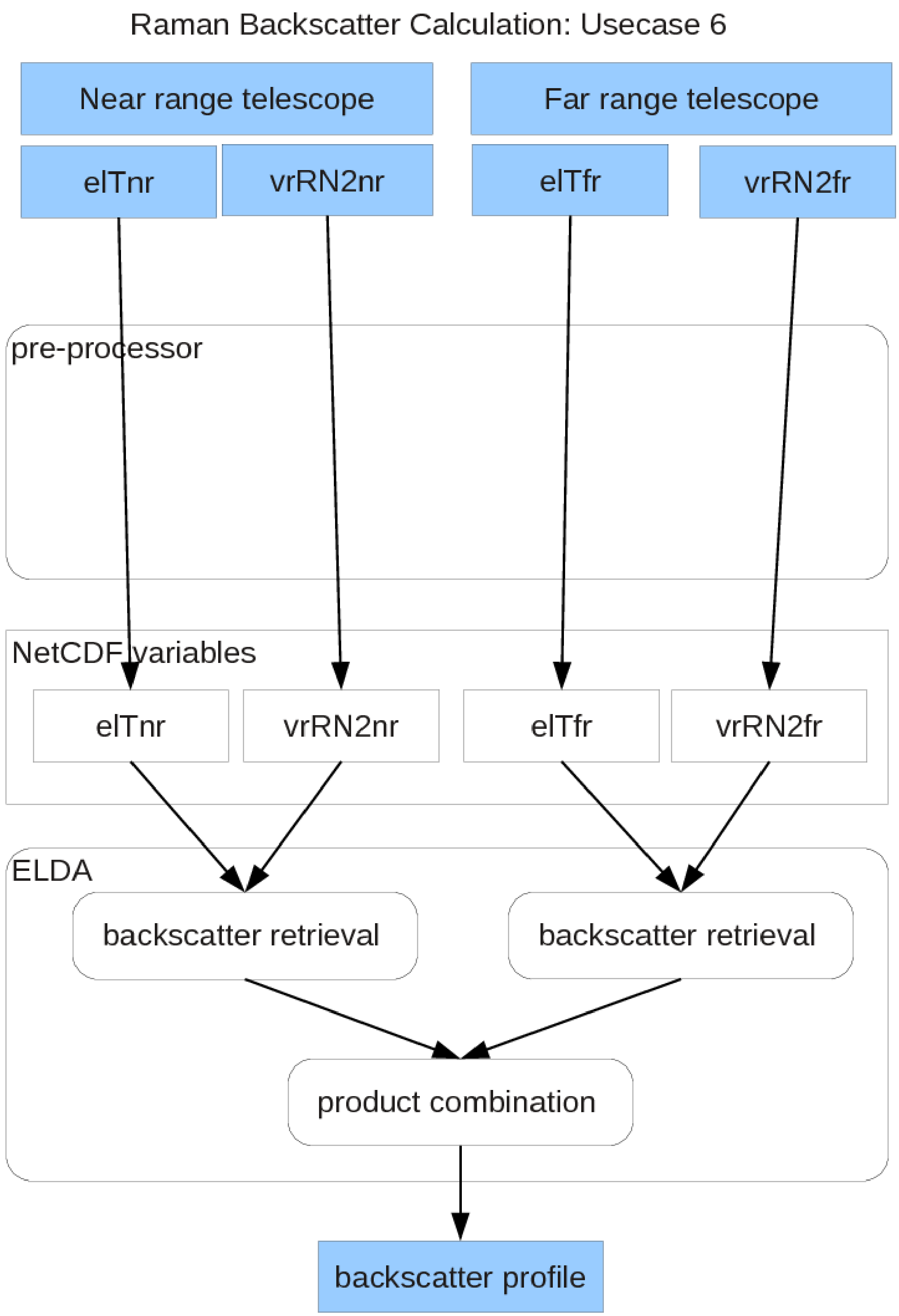

Raman Backscatter Calculation: Usecase 1
elTnr
elTfr
vrRN2
pre-processor
signal combination
NetCDF variables
elT
vrRN2
ELDA
backscatter retrieval
backscatter profile

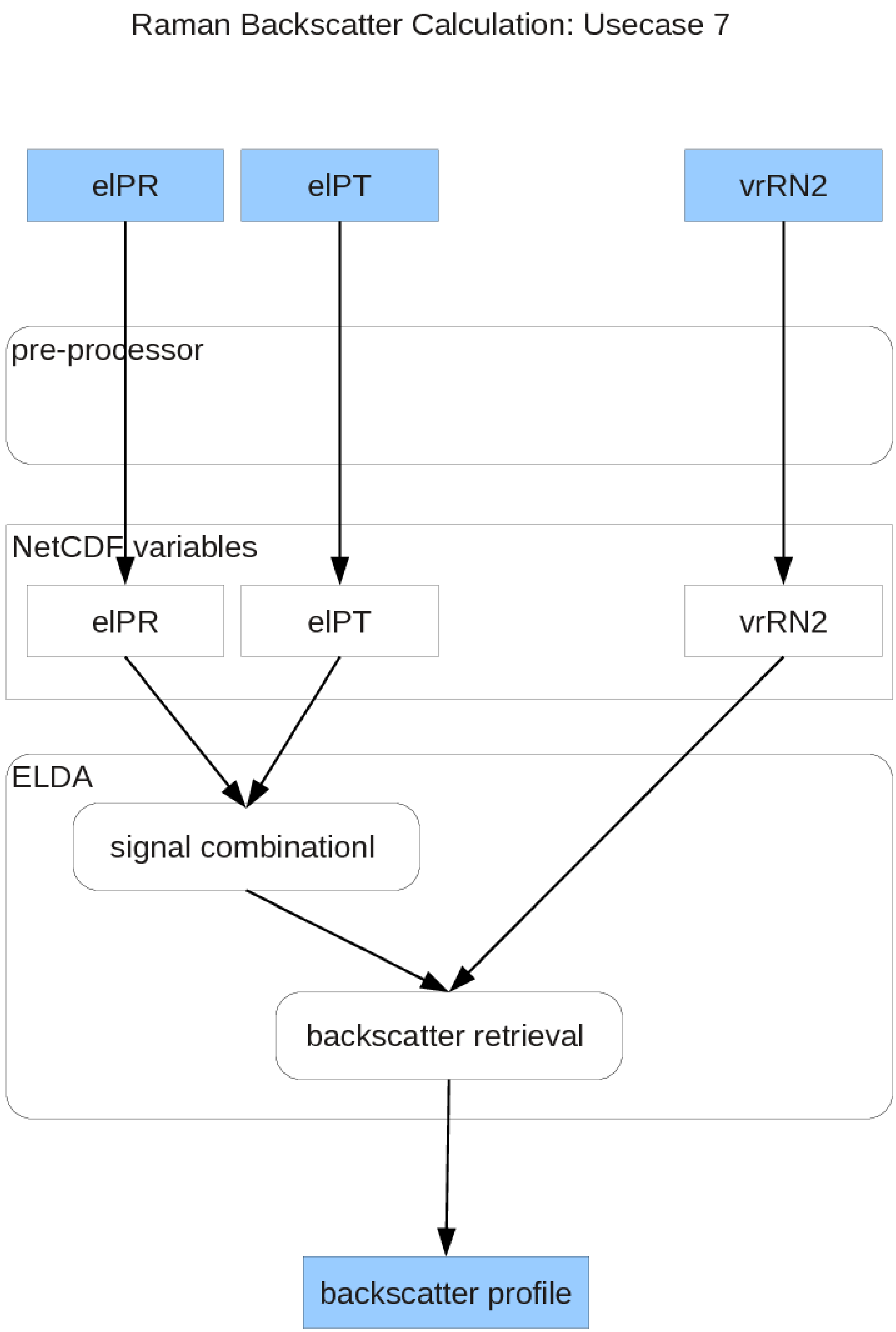

Raman Backscatter Calculation: Usecase 2
elTnr
elTfr
vrRN2
pre-processor
NetCDF variables
elTnr
elTfr
vrRN2
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

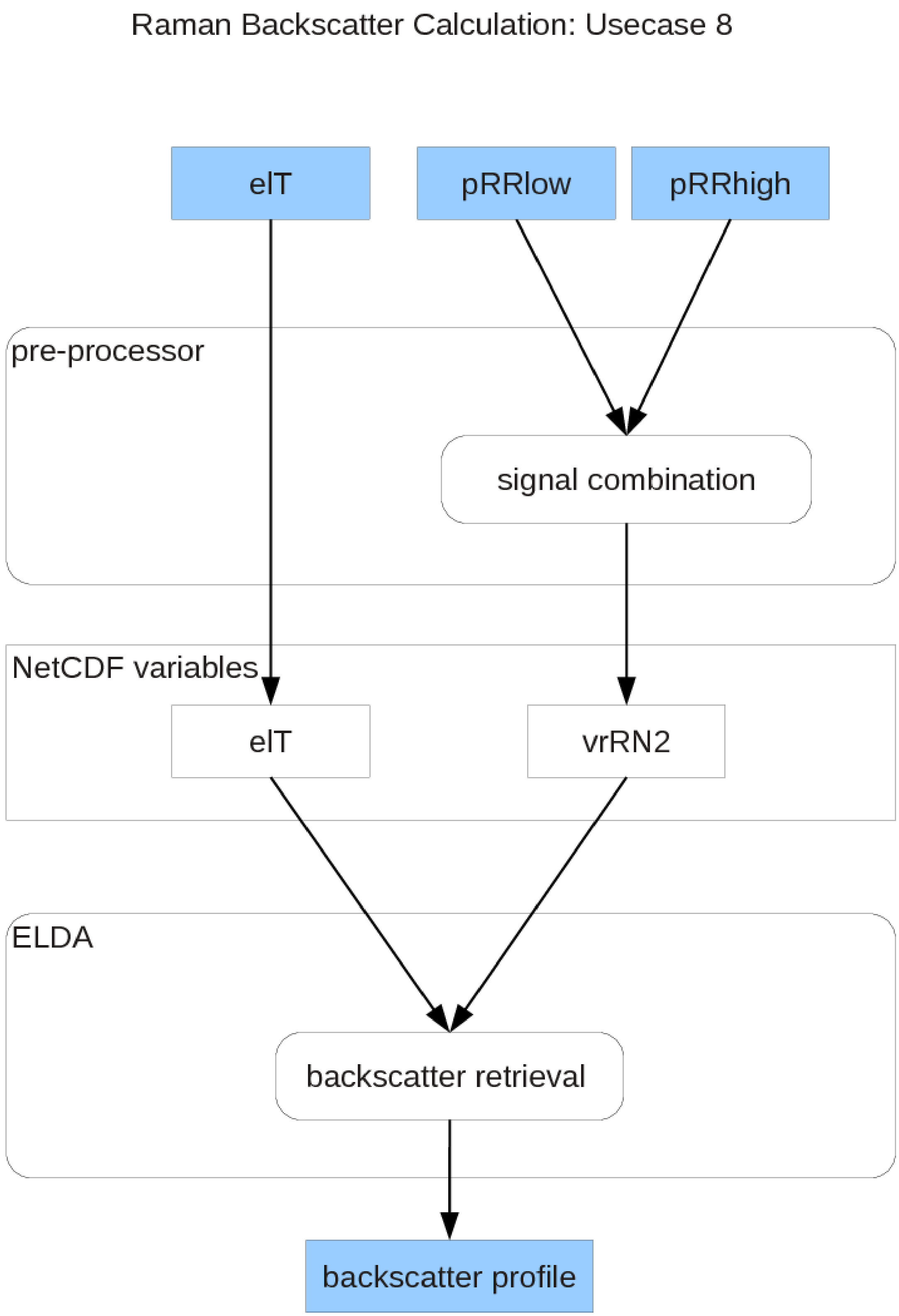

Raman Backscatter Calculation: Usecase 3
elT
vrRN2nr
vrRN2fr
pre-processor
signal combination
NetCDF variables
elT
vrRN2
ELDA
backscatter retrieval
backscatter profile

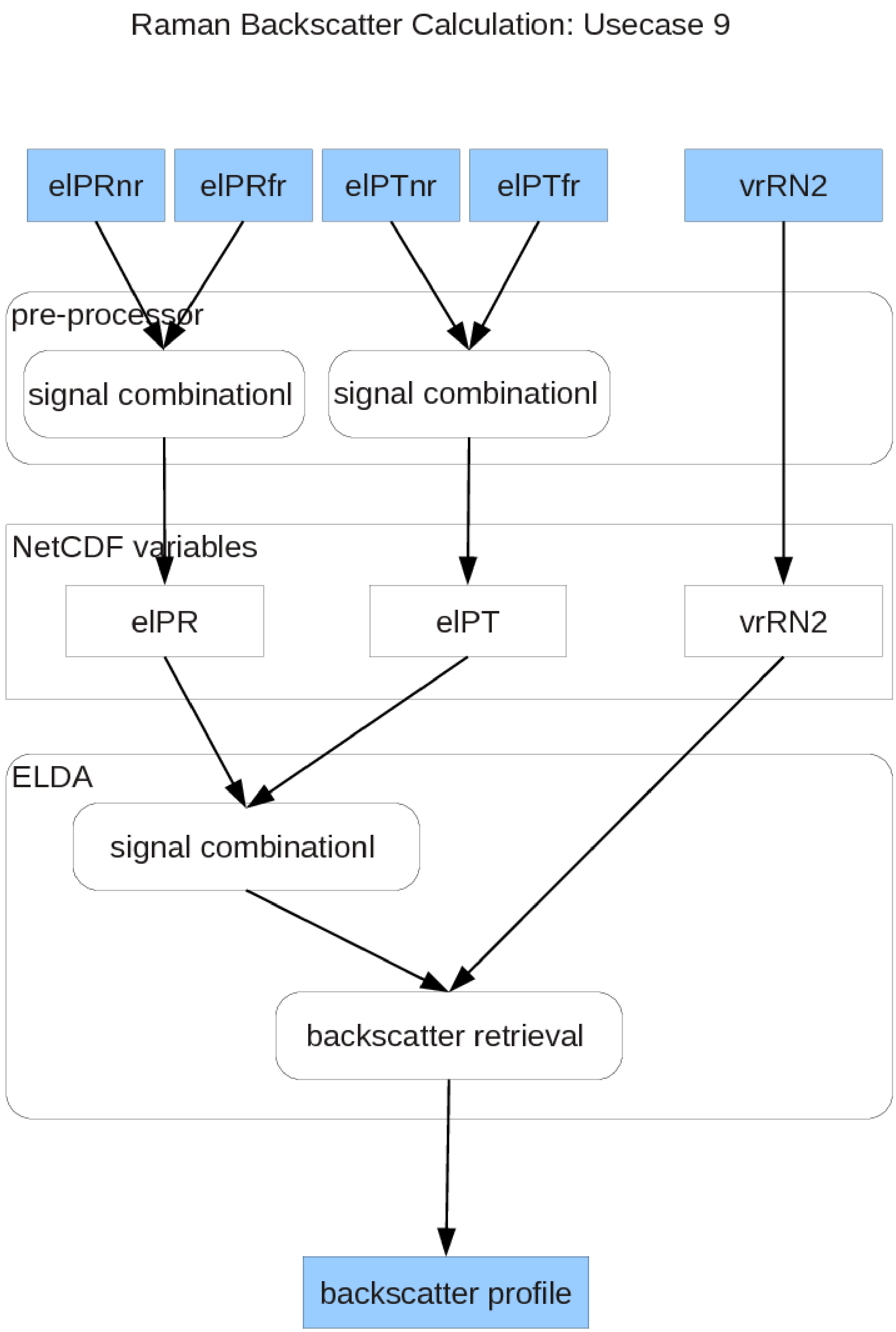

Raman Backscatter Calculation: Usecase 4
elT
vrRN2nr
vrRN2fr
pre-processor
NetCDF variables
elT
vrRN2nr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

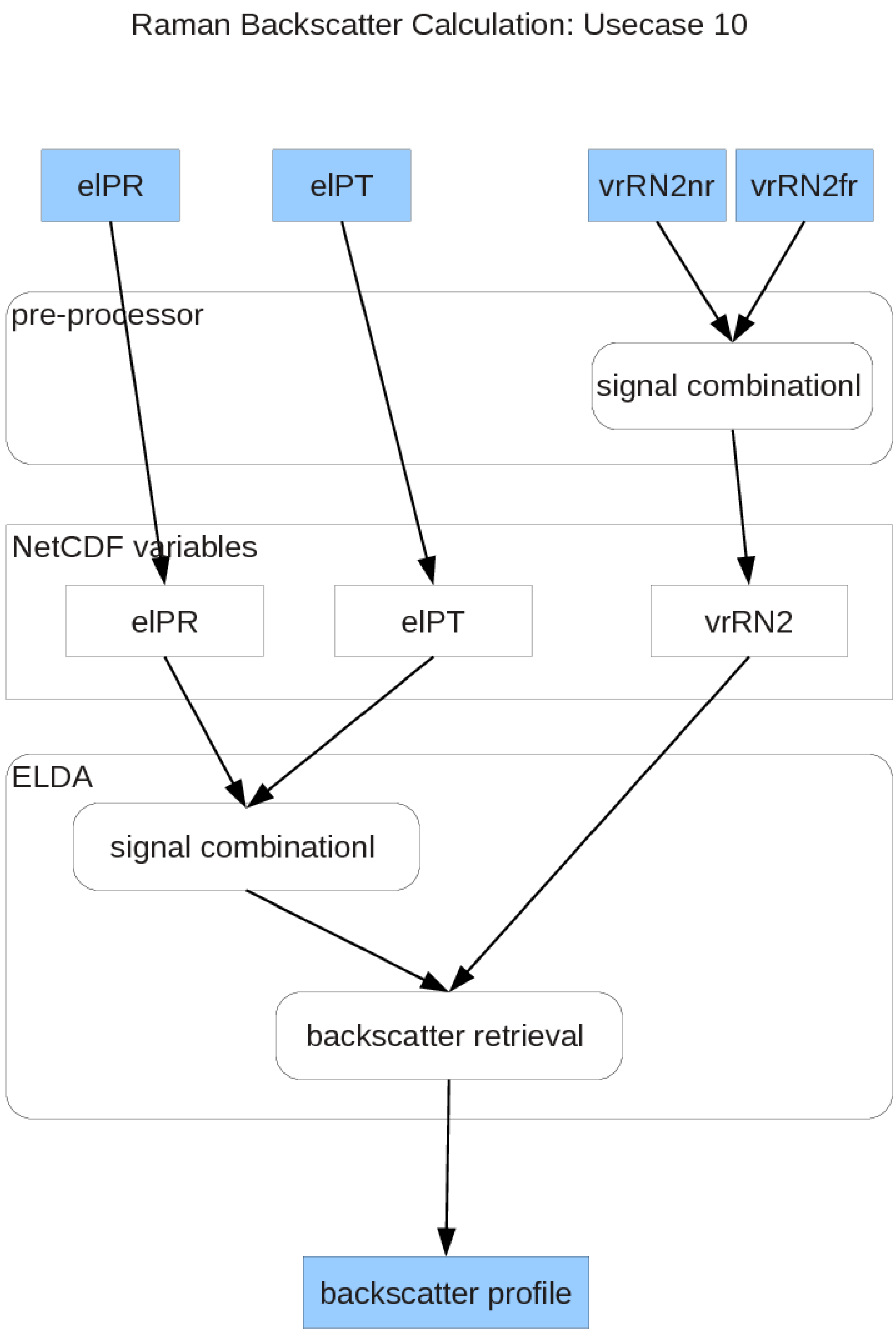

Raman Backscatter Calculation: Usecase 5
elTnr
elTfr
vrRN2nr
vrRN2fr
pre-processor
signal combination
signal combination
NetCDF variables
elT
vrRN2
ELDA
backscatter retrieval
backscatter profile

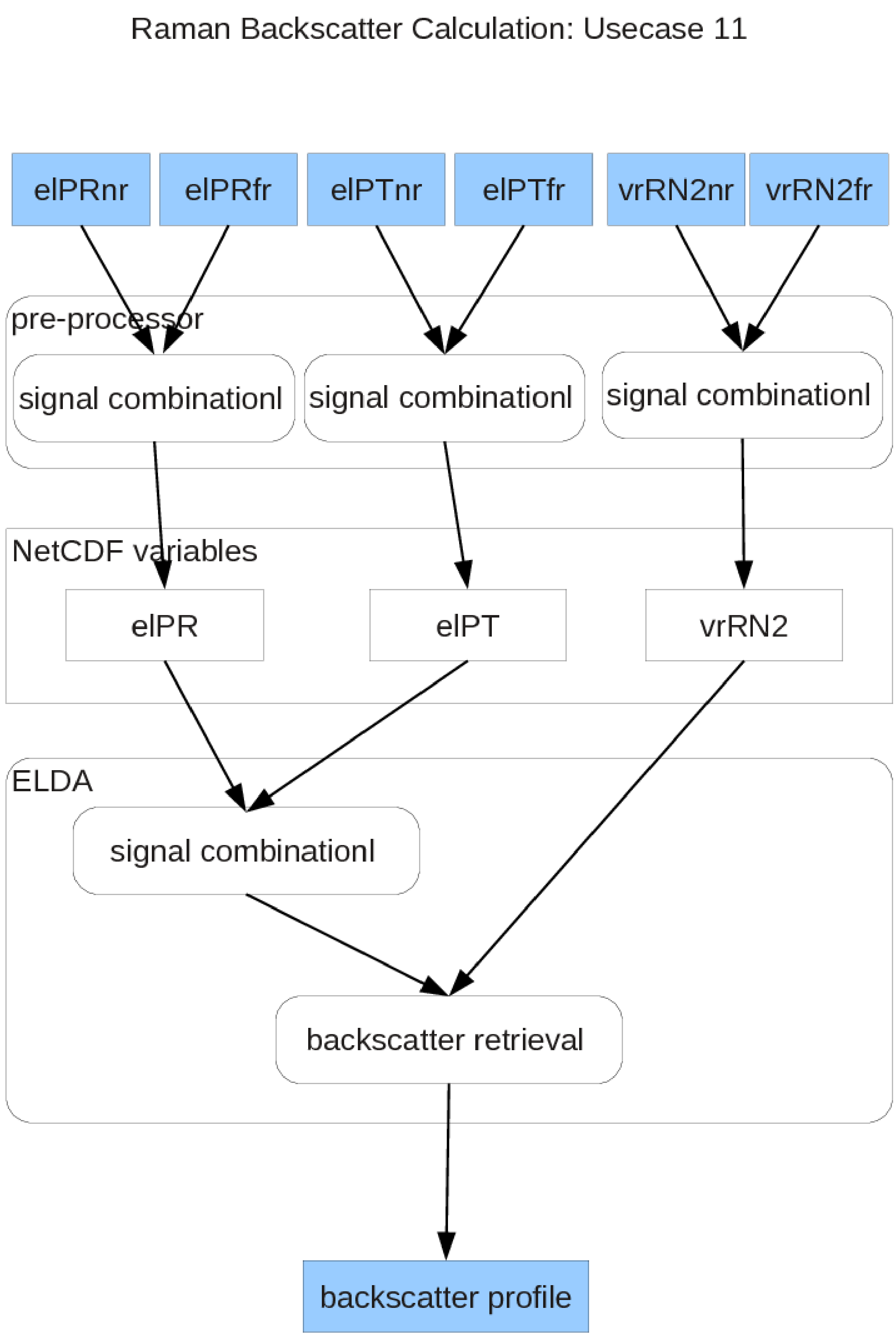

Raman Backscatter Calculation: Usecase 6
Near range telescope
Far range telescope
elTnr
vrRN2nr
elTfr
vrRN2fr
pre-processor
NetCDF variables
elTnr
vrRN2nr
elTfr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

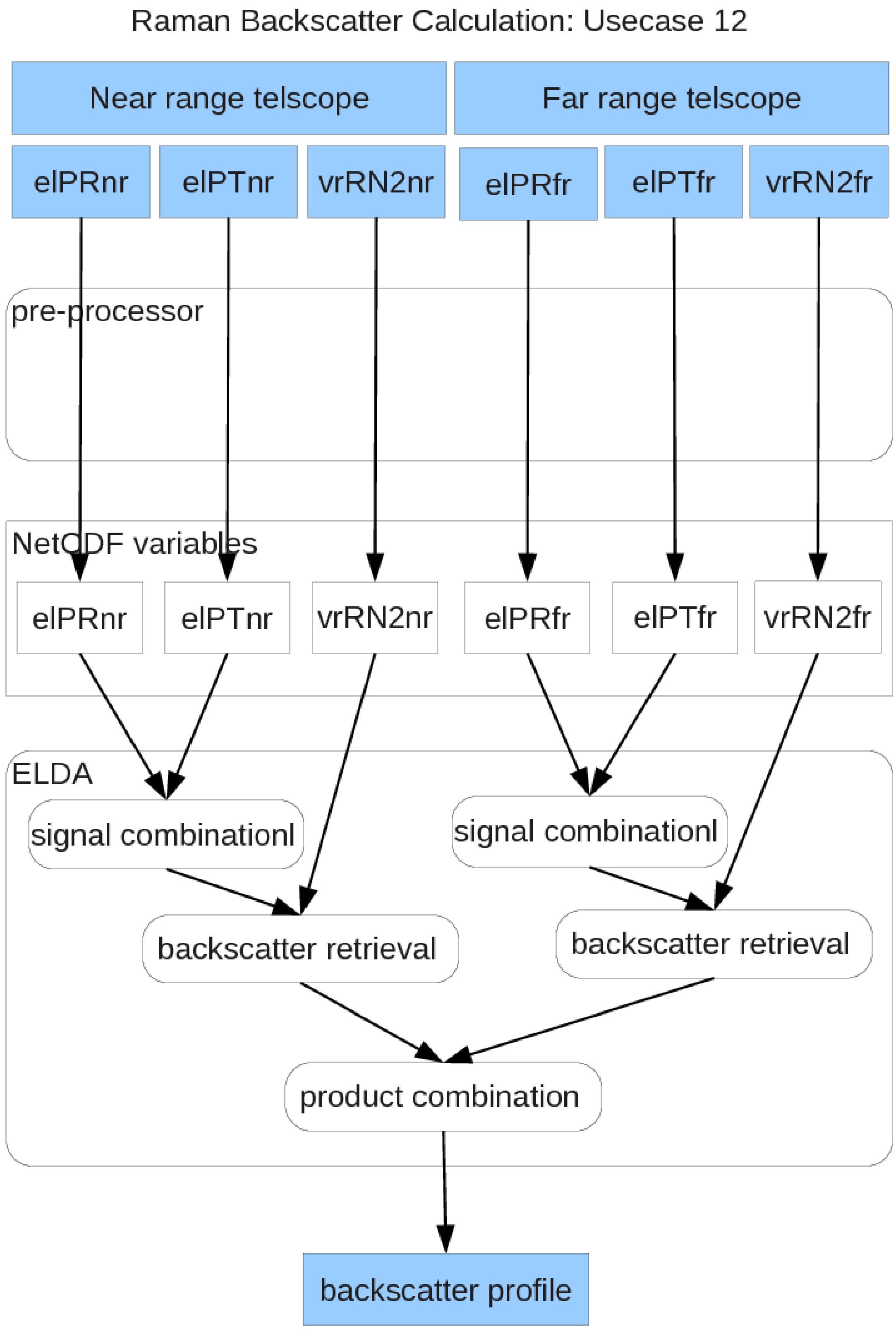

Raman Backscatter Calculation: Usecase 7
elPR
elPT
vrRN2
pre-processor
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

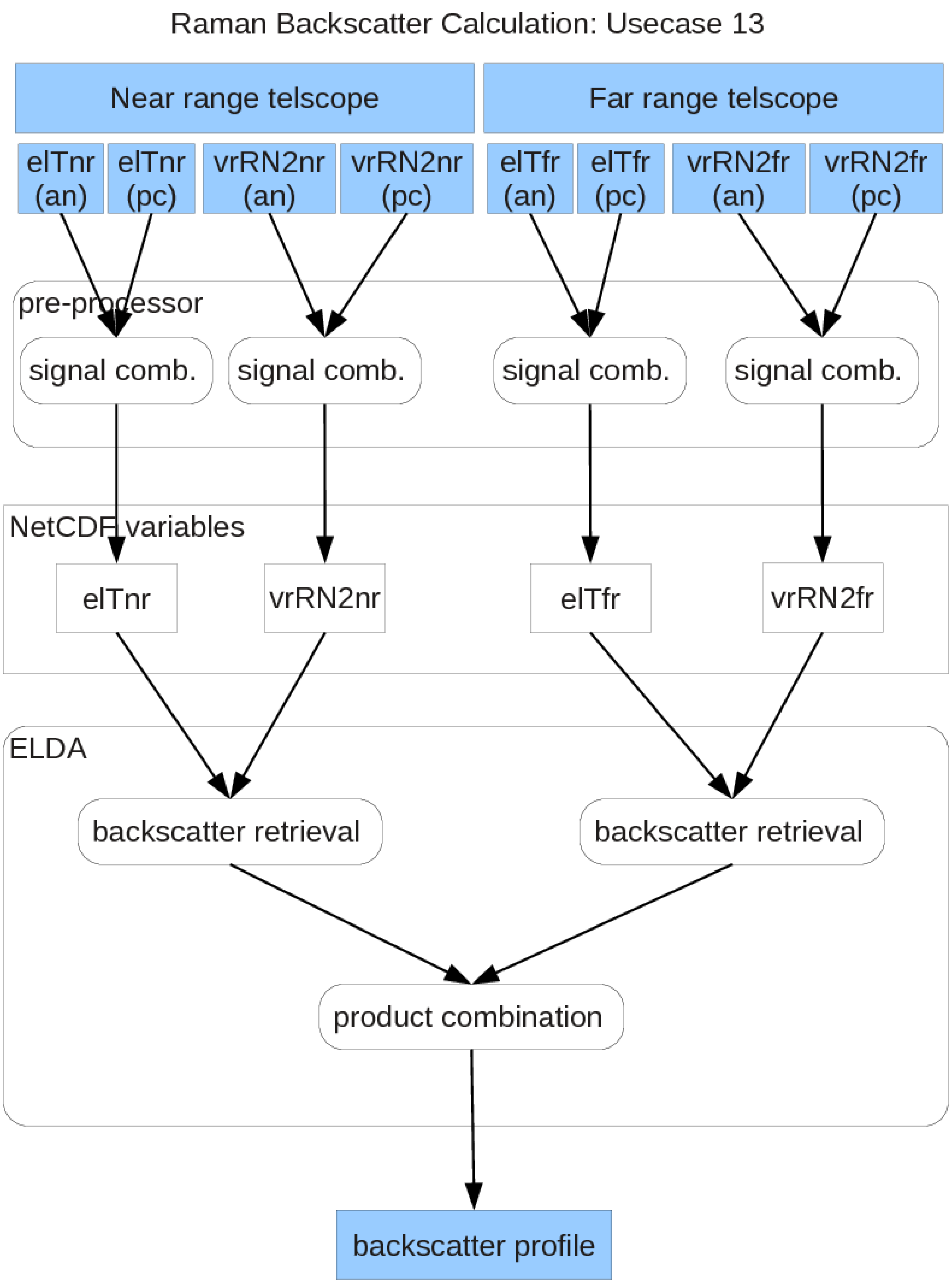

Raman Backscatter Calculation: Usecase 8
elT
pRRlow
pRRhigh
pre-processor
signal combination
NetCDF variables
elT
vrRN2
ELDA
backscatter retrieval
backscatter profile

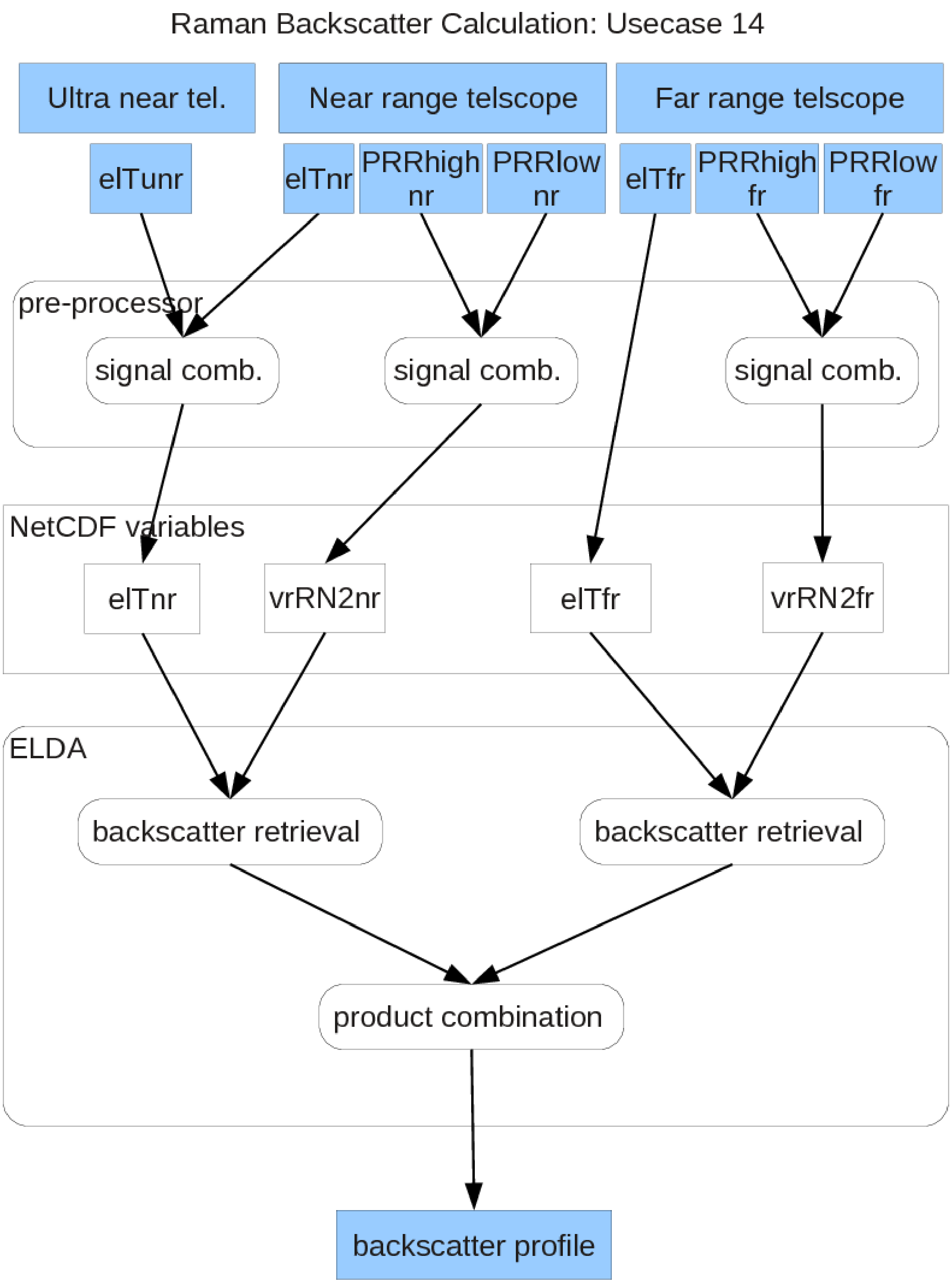

Raman Backscatter Calculation: Usecase 9
elPRnr
elPRfr
elPTnr
elPTfr
vrRN2
pre-processor
signal combinationl
signal combinationl
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

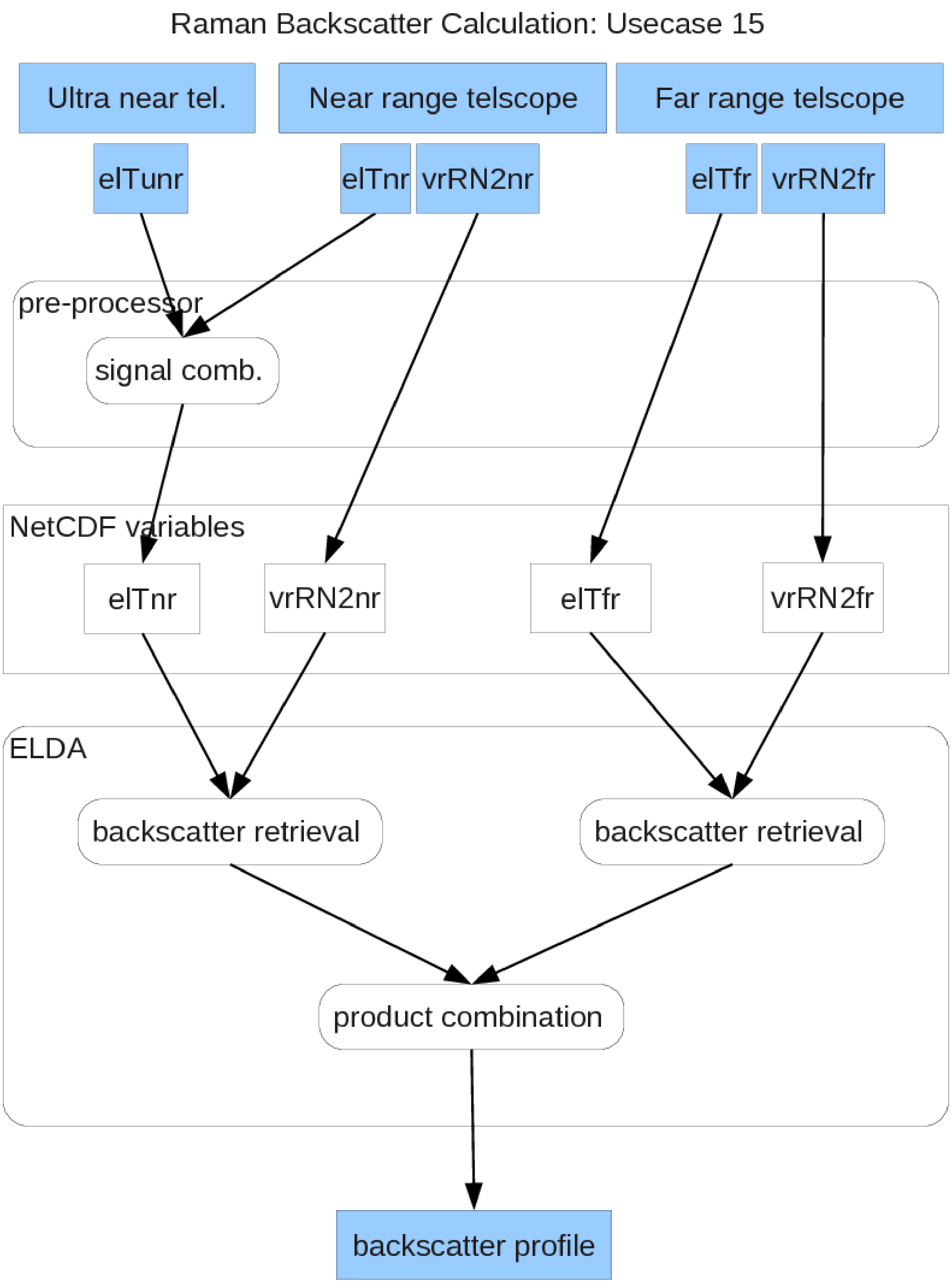

Raman Backscatter Calculation: Usecase 10
elPR
elPT
vrRN2nr
vrRN2fr
pre-processor
signal combinationl
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

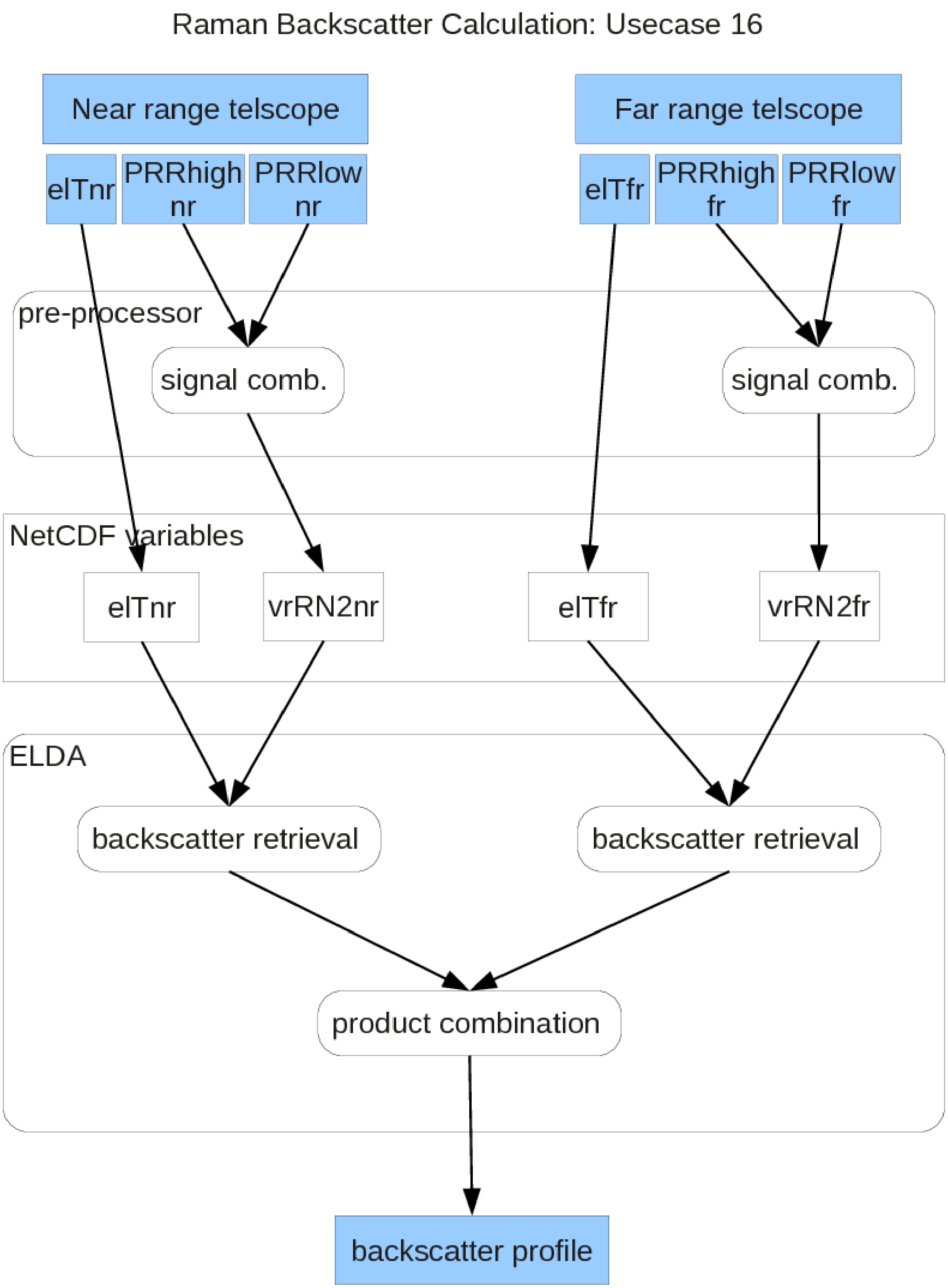

Raman Backscatter Calculation: Usecase 11
elPRnr
elPRfr
elPTnr
elPTfr
vrRN2nr
vrRN2fr
pre-processor
signal combinationl
signal combinationl
signal combinationl
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

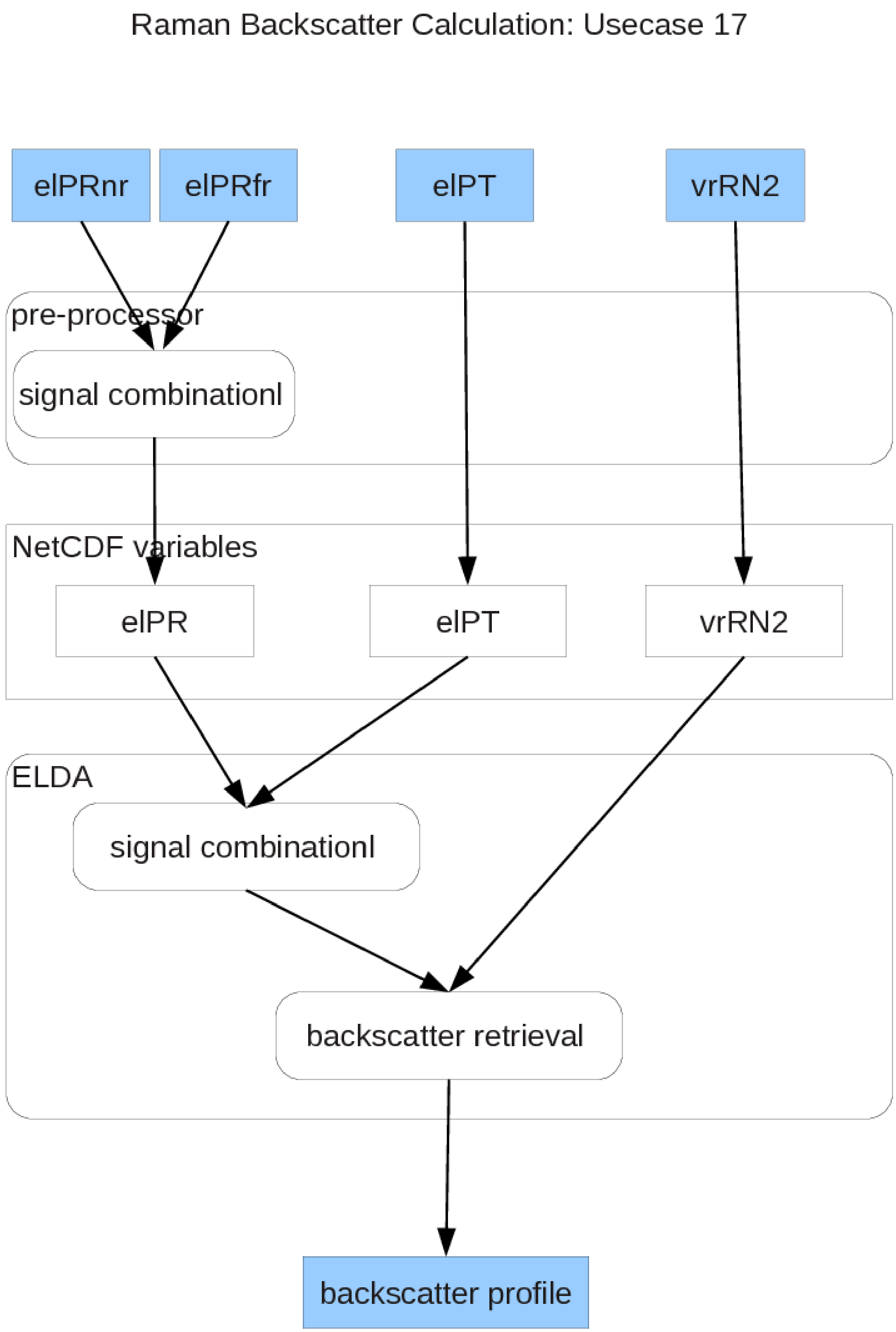

Raman Backscatter Calculation: Usecase 12
Near range telscope
Far range telscope
elPRnr
elPTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
pre-processor
NetCDF variables
elPRnr
elPTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
ELDA
signal combinationl
signal combinationl
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

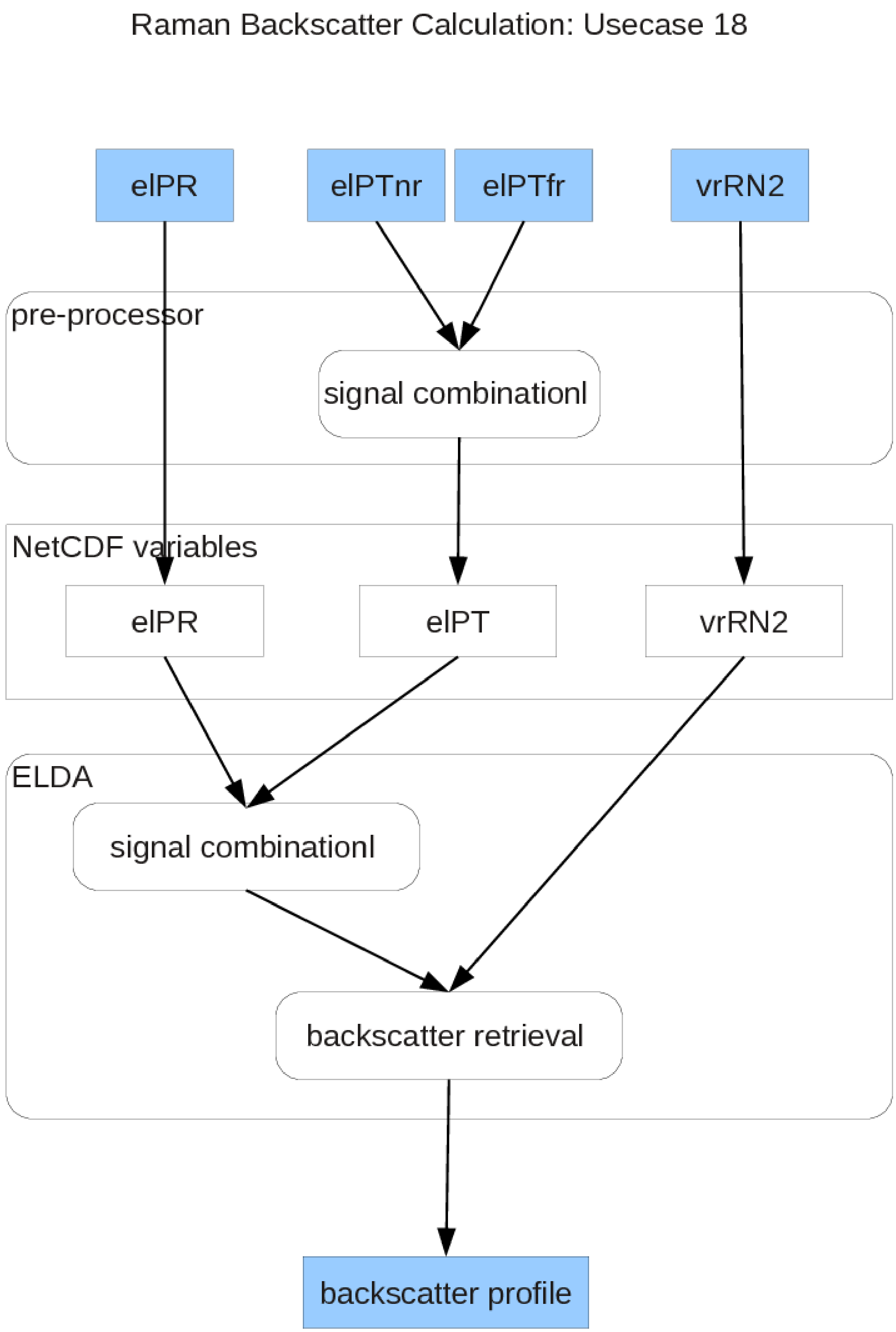

Raman Backscatter Calculation: Usecase 13
Near range telscope
Far range telscope
elTnr (an)
elTnr (pc)
vrRN2nr (an)
vrRN2nr (pc)
elTfr (an)
elTfr (pc)
vrRN2fr (an)
vrRN2fr (pc)
pre-processor
signal comb.
signal comb.
signal comb.
signal comb.
NetCDF variables
elTnr
vrRN2nr
elTfr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

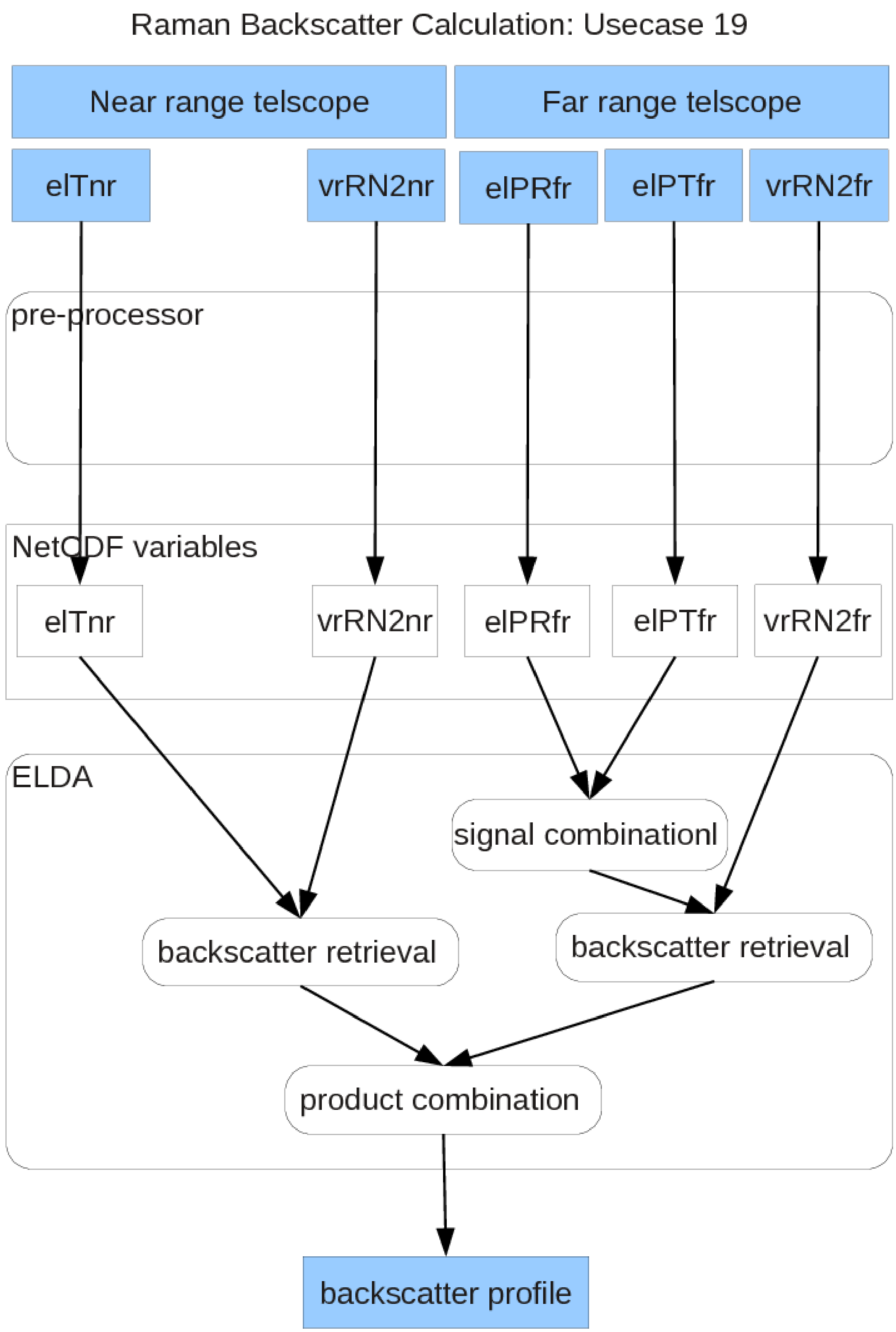

Raman Backscatter Calculation: Usecase 14
Ultra near tel.
Near range telscope
Far range telscope
elTunr
elTnr
PRRhigh nr
PRRlow nr
elTfr
PRRhigh fr
PRRlow fr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elTnr
vrRN2nr
elTfr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile


Raman Backscatter Calculation: Usecase 15
Ultra near tel.
Near range telscope
Far range telscope
elTunr
elTnr
vrRN2nr
elTfr
vrRN2fr
pre-processor
signal comb.
NetCDF variables
elTnr
vrRN2nr
elTfr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

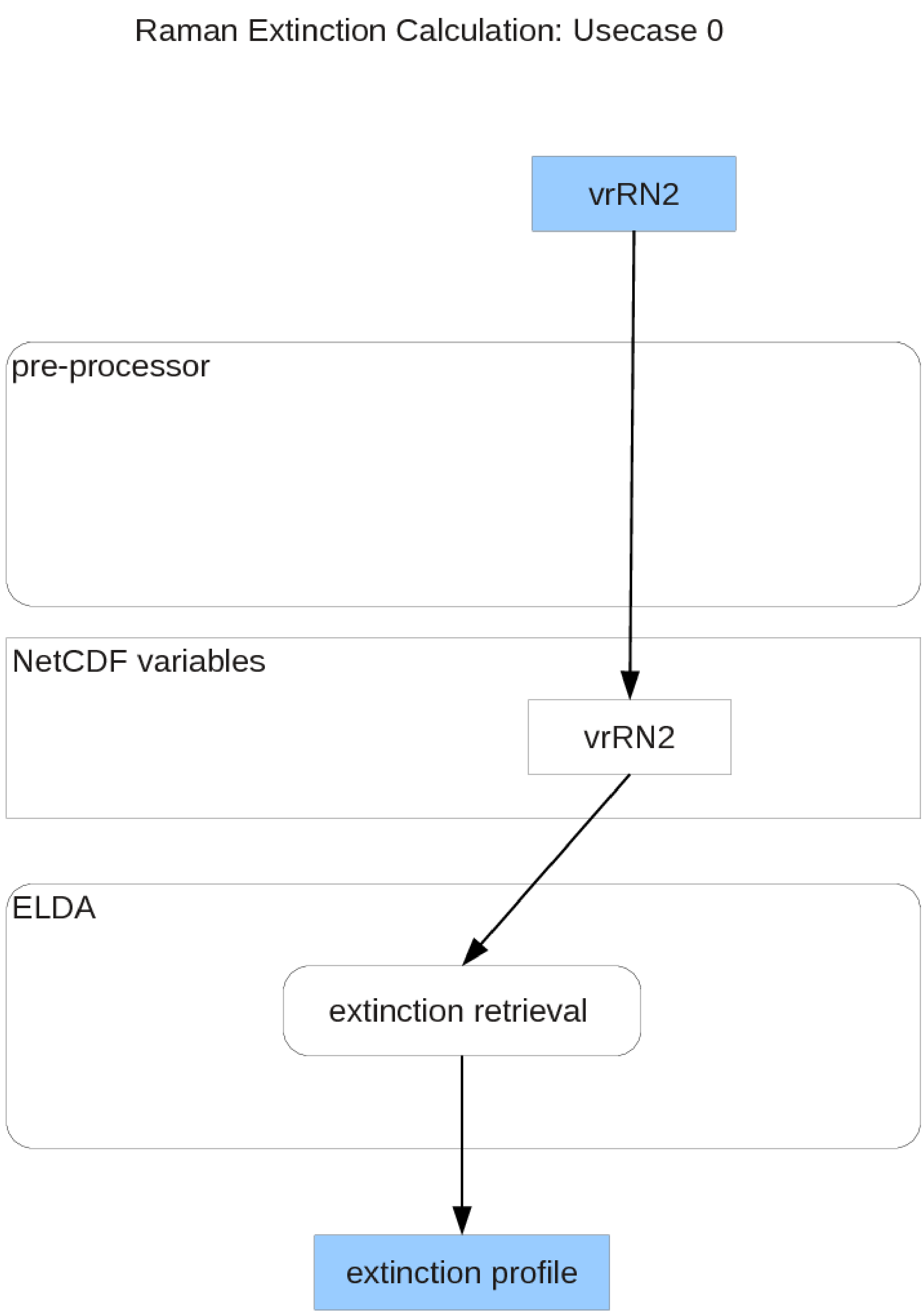

Raman Backscatter Calculation: Usecase 16
Near range telscope
elTnr
PRRhigh nr
PRRlow nr
Far range telscope
elTfr
PRRhigh fr
PRRlow fr
pre-processor
signal comb.
signal comb.
NetCDF variables
elTnr
vrRN2nr
elTfr
vrRN2fr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

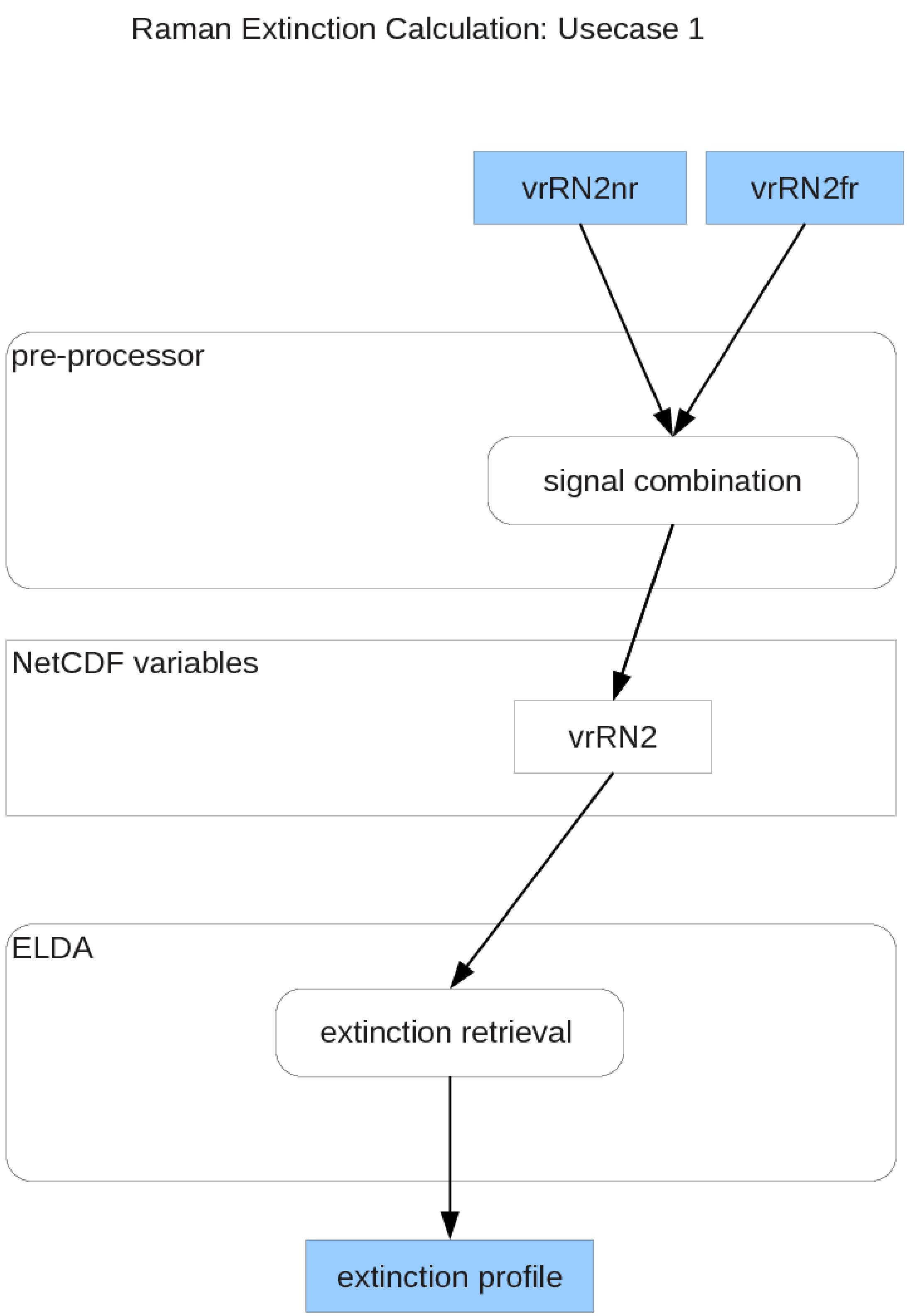

Raman Backscatter Calculation: Usecase 17
elPRnr
elPRfr
elPT
vrRN2
pre-processor
signal combinationl
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

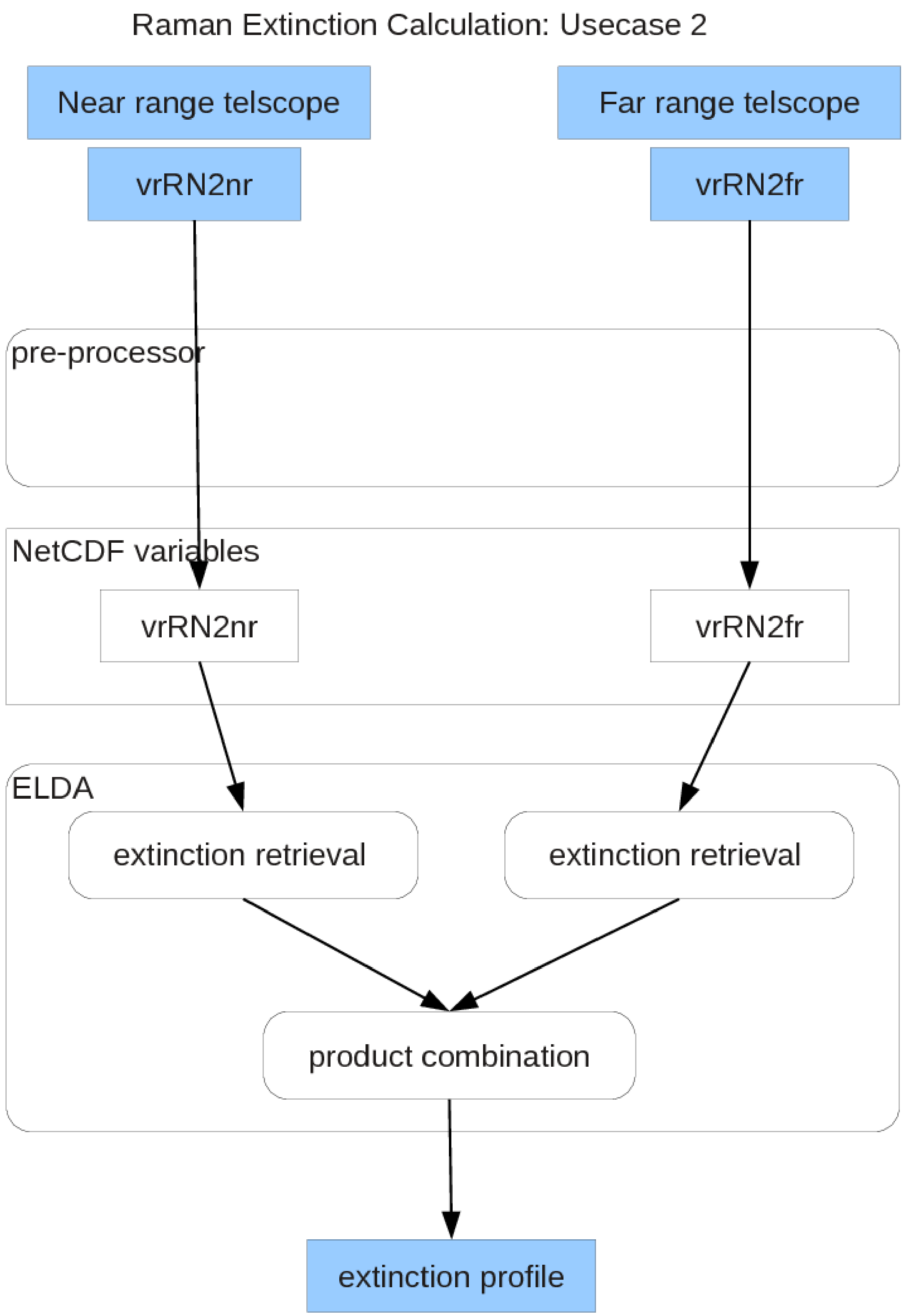

Raman Backscatter Calculation: Usecase 18
elPR
elPTnr
elPTfr
vrRN2
pre-processor
signal combinationl
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal combinationl
backscatter retrieval
backscatter profile

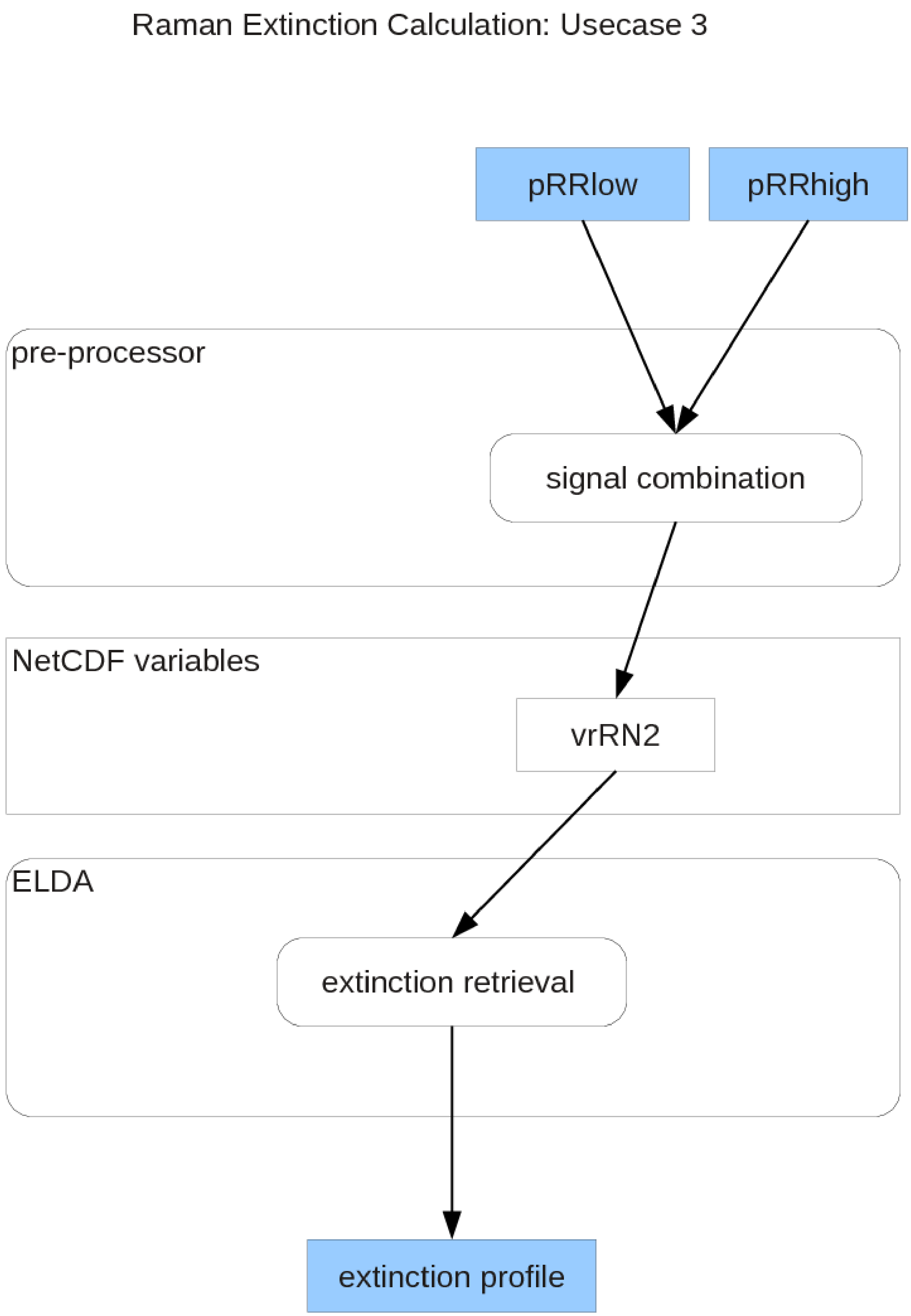

Raman Backscatter Calculation: Usecase 19
Near range telscope
Far range telscope
elTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
pre-processor
NetCDF variables
elTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
ELDA
signal combinationl
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

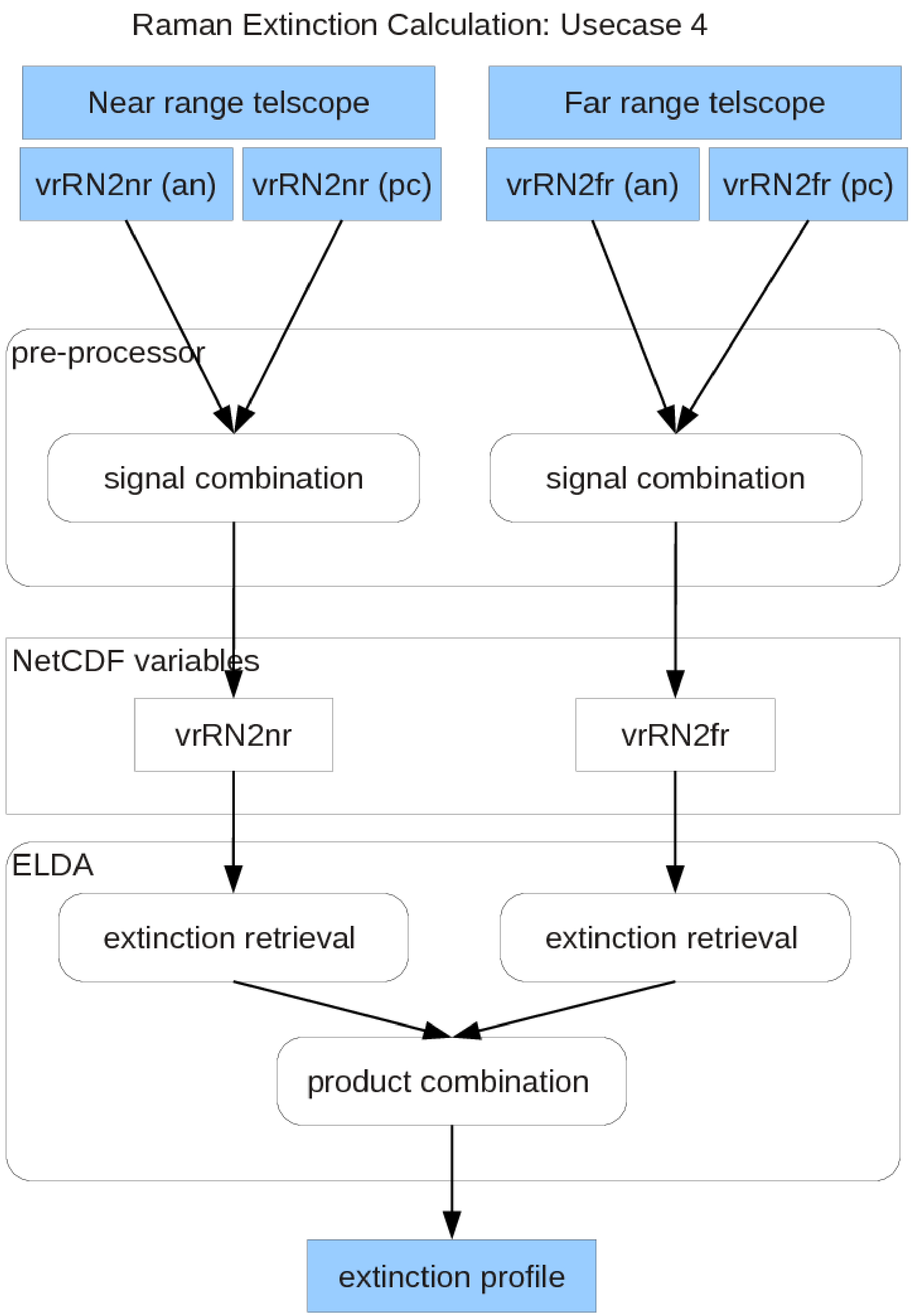
3.6.2 Raman extinction

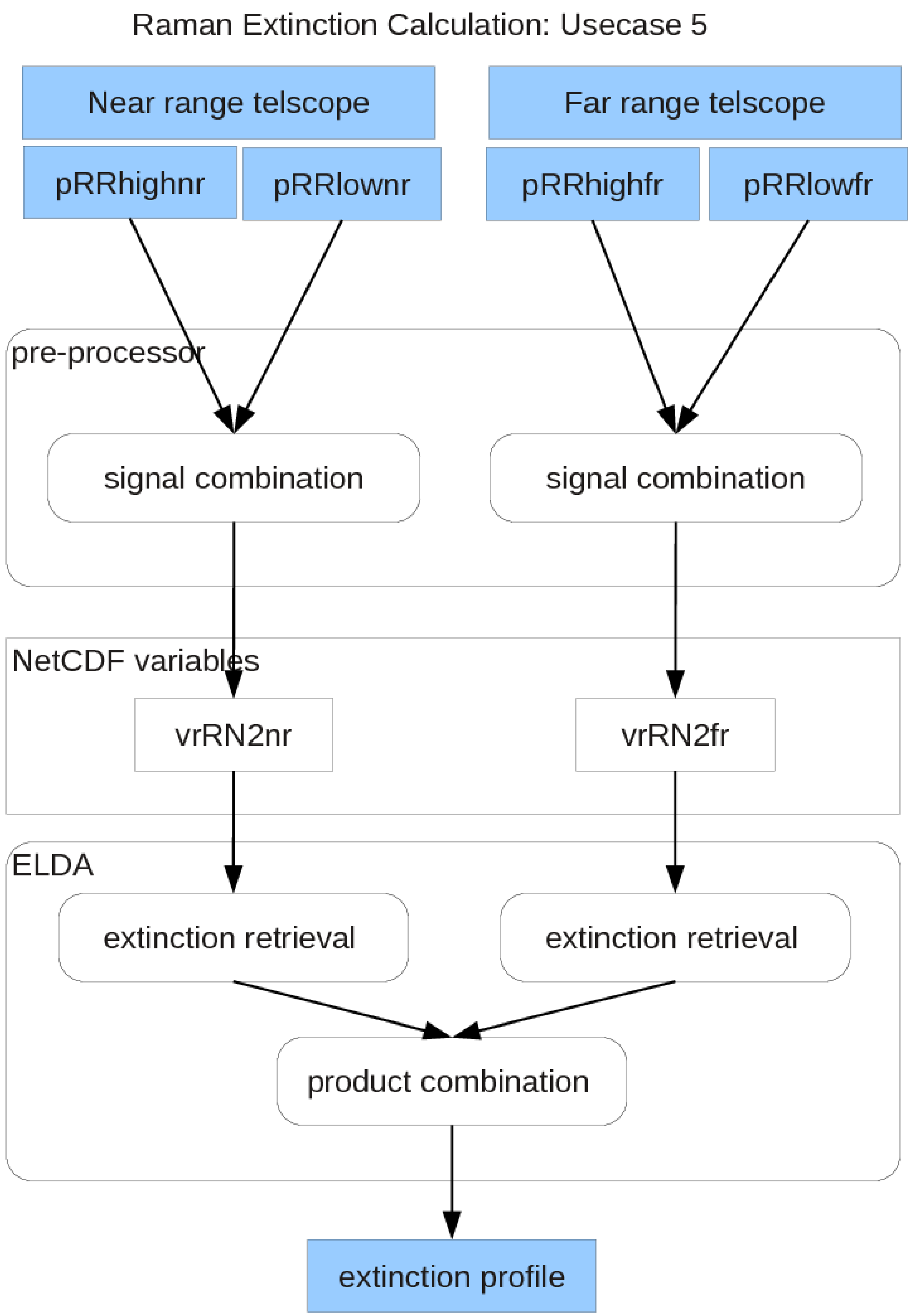



Raman Extinction Calculation: Usecase 1
vrRN2nr
vrRN2fr
pre-processor
signal combination
NetCDF variables
vrRN2
ELDA
extinction retrieval
extinction profile

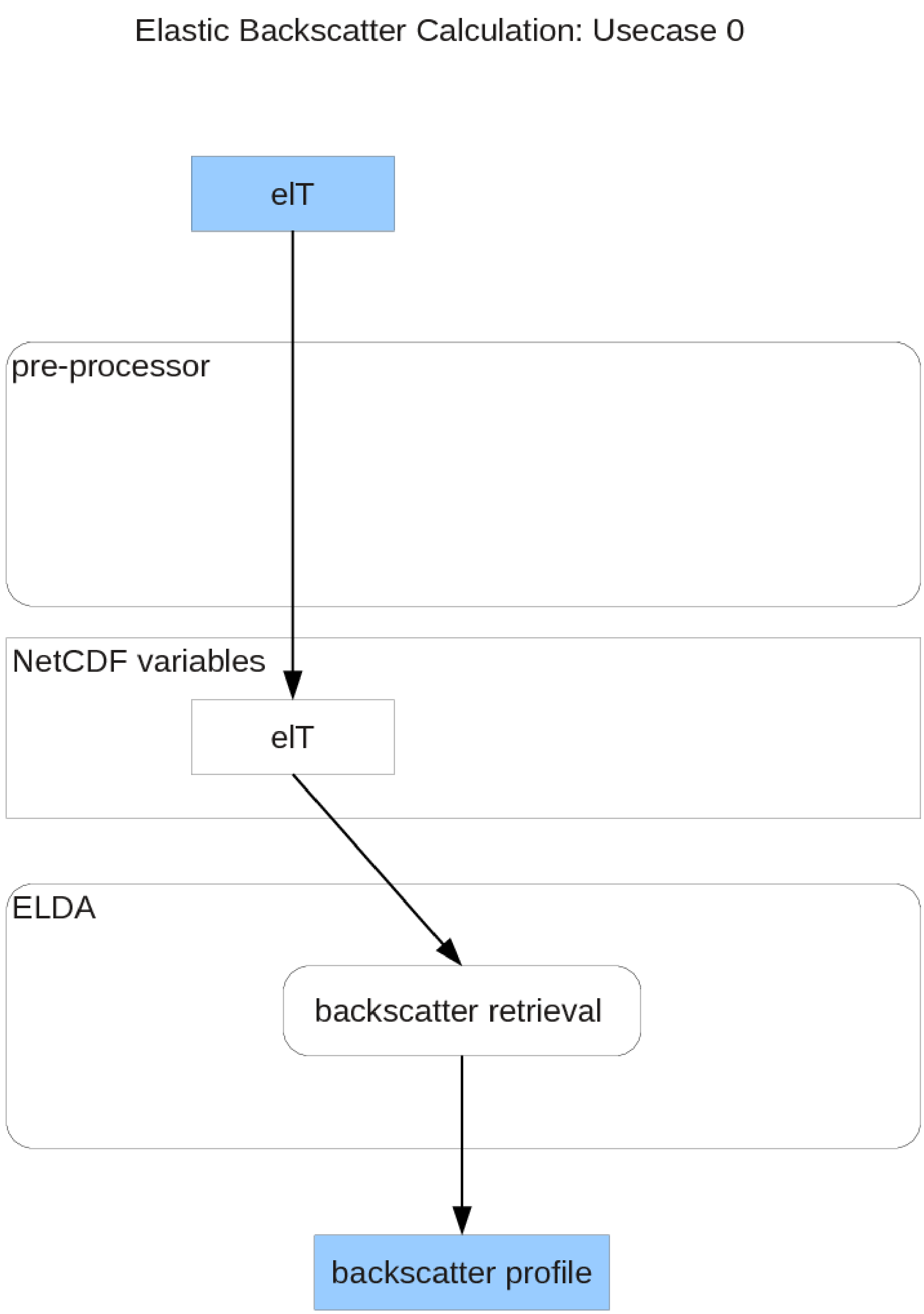

Raman Extinction Calculation: Usecase 2
Near range telscope
Far range telscope
vrRN2nr
vrRN2fr
pre-processor
NetCDF variables
vrRN2nr
vrRN2fr
ELDA
extinction retrieval
extinction retrieval
product combination
extinction profile

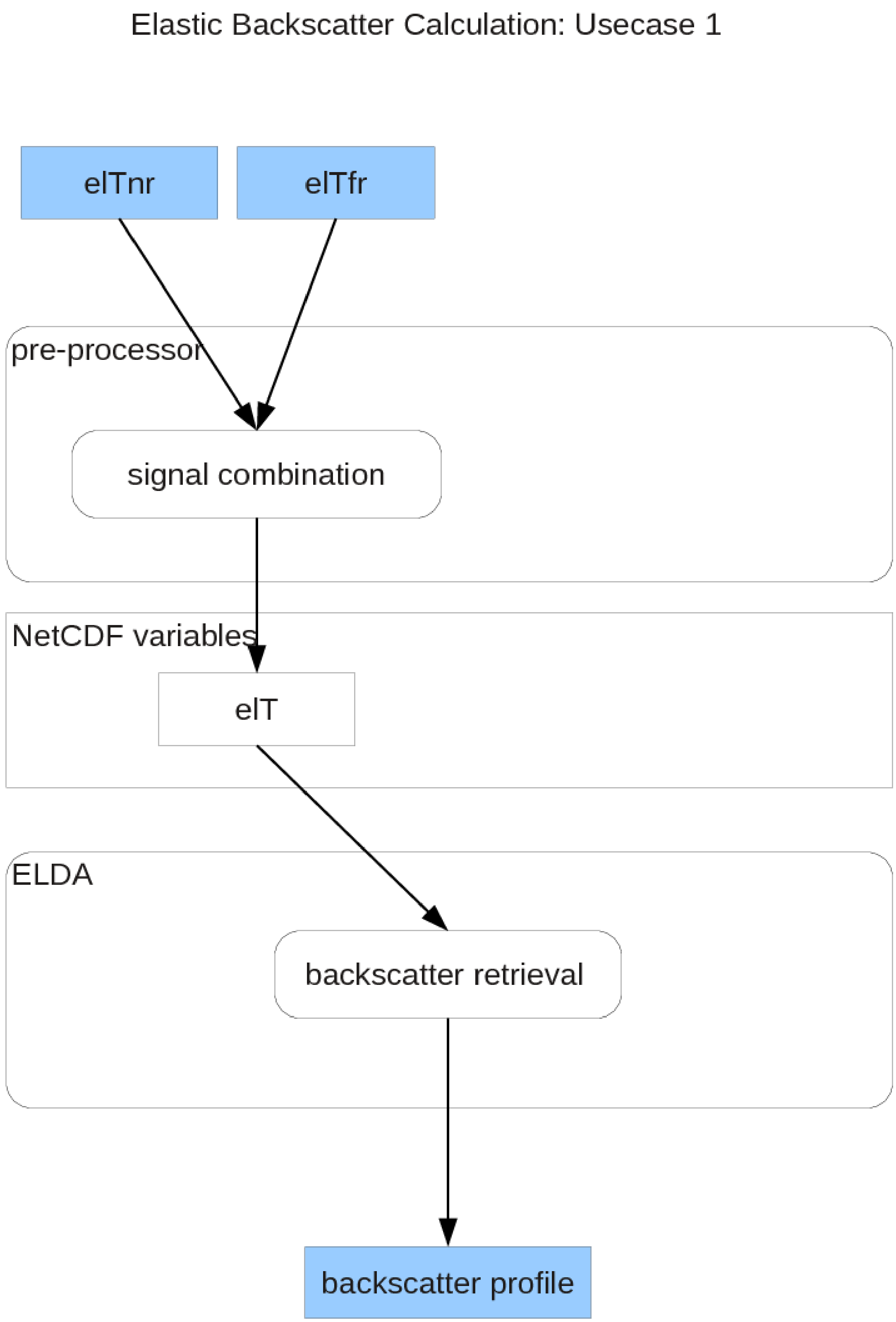

Raman Extinction Calculation: Usecase 3
pRRlow
pRRhigh
pre-processor
signal combination
NetCDF variables
vrRN2
ELDA
extinction retrieval
extinction profile

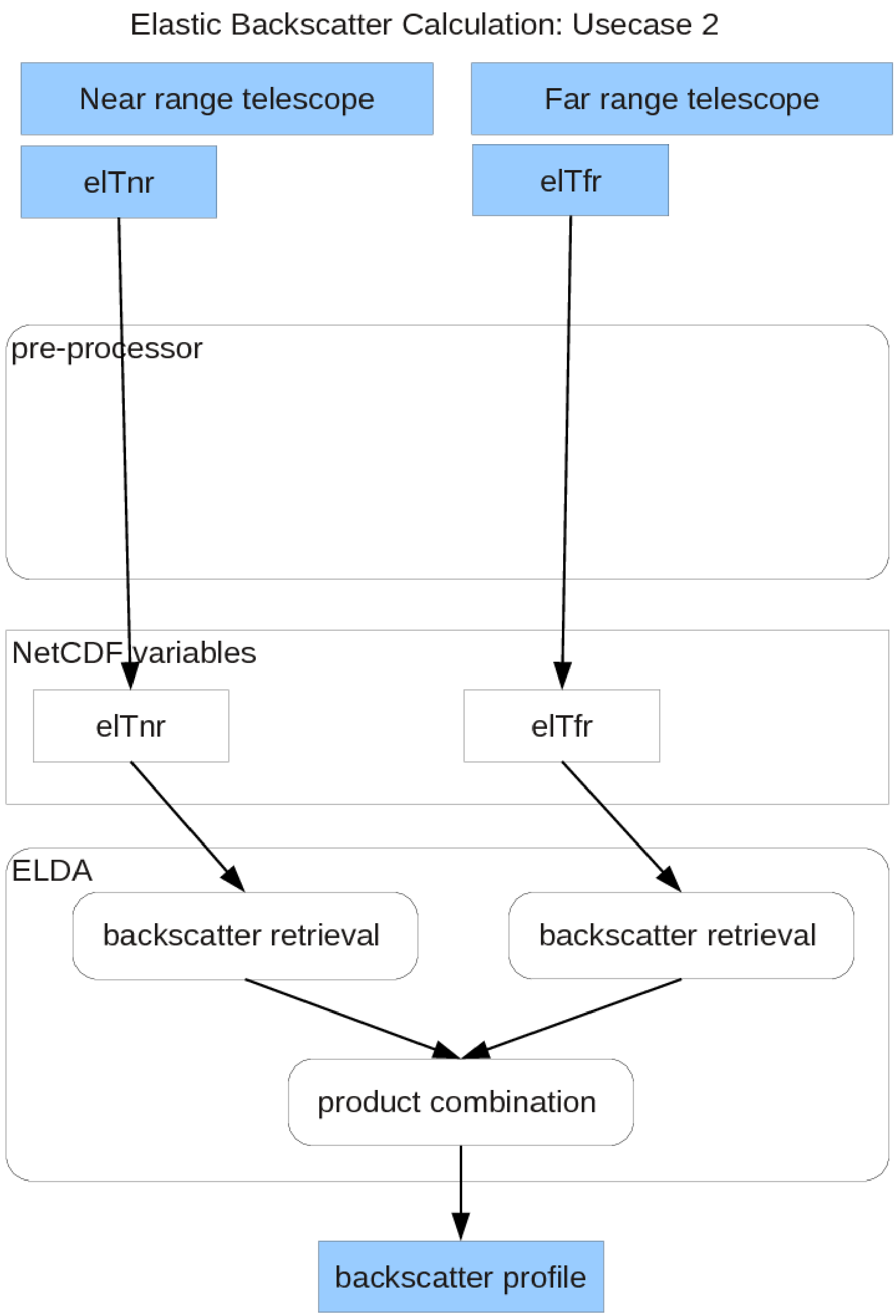

Raman Extinction Calculation: Usecase 4
Near range telscope
Far range telscope
vrRN2nr (an)
vrRN2nr (pc)
vrRN2fr (an)
vrRN2fr (pc)
pre-processor
signal combination
signal combination
NetCDF variables
vrRN2nr
vrRN2fr
ELDA
extinction retrieval
extinction retrieval
product combination
extinction profile

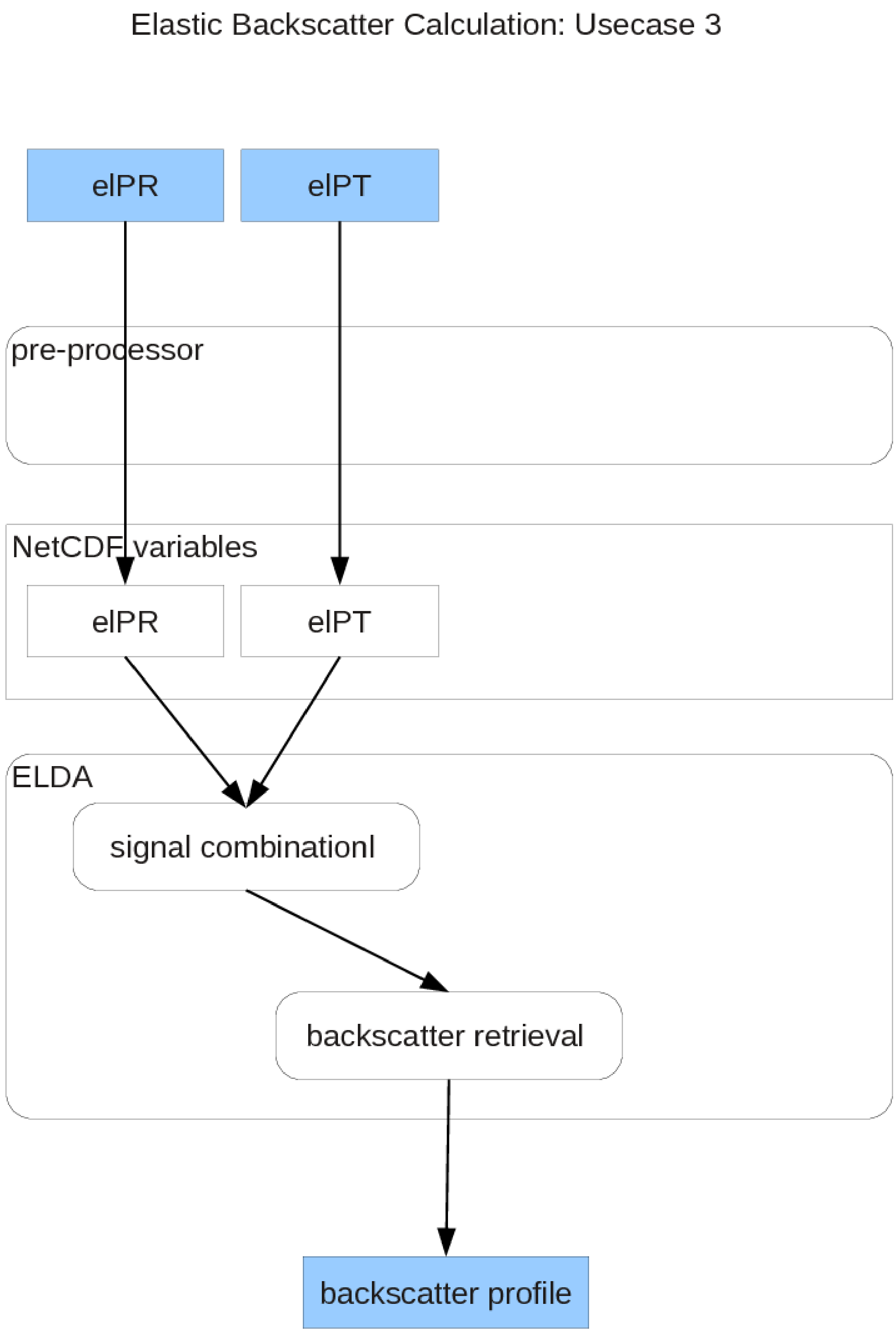

Raman Extinction Calculation: Usecase 5
Near range telscope
Far range telscope
pRRhighnr
pRRlownr
pRRhighfr
pRRlowfr
pre-processor
signal combination
signal combination
NetCDF variables
vrRN2nr
vrRN2fr
ELDA
extinction retrieval
extinction retrieval
product combination
extinction profile

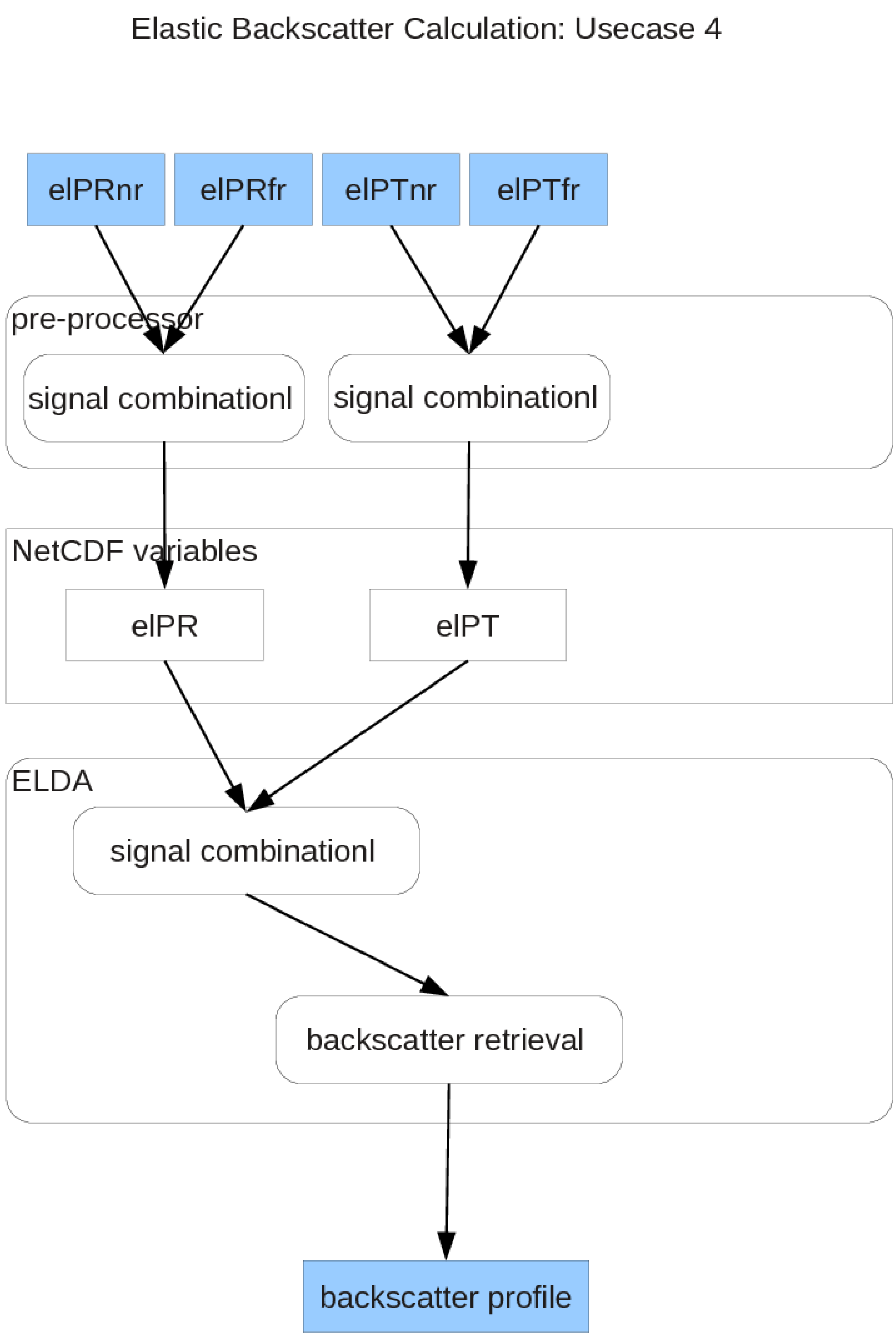
3.6.3 Elastic backscatter

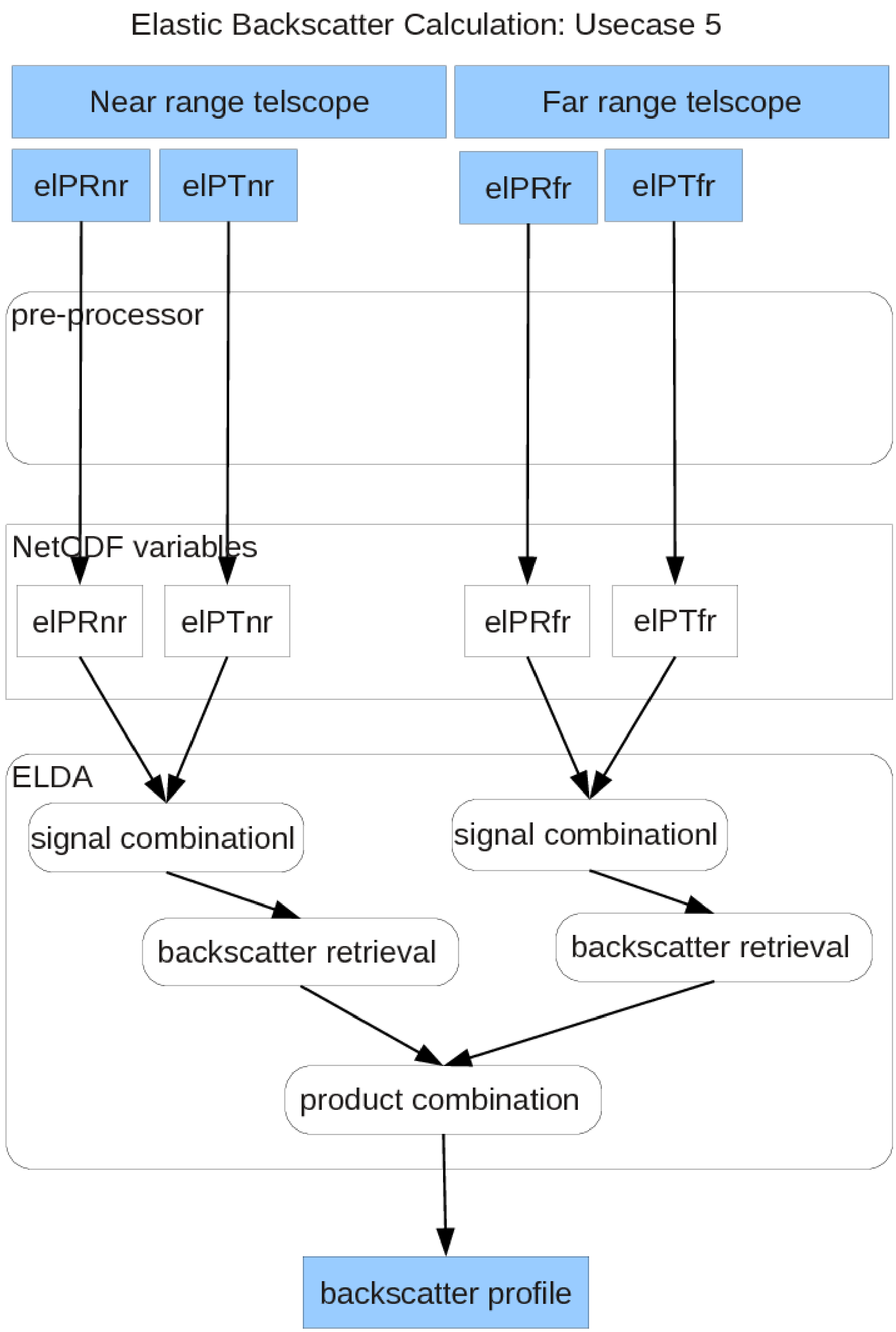


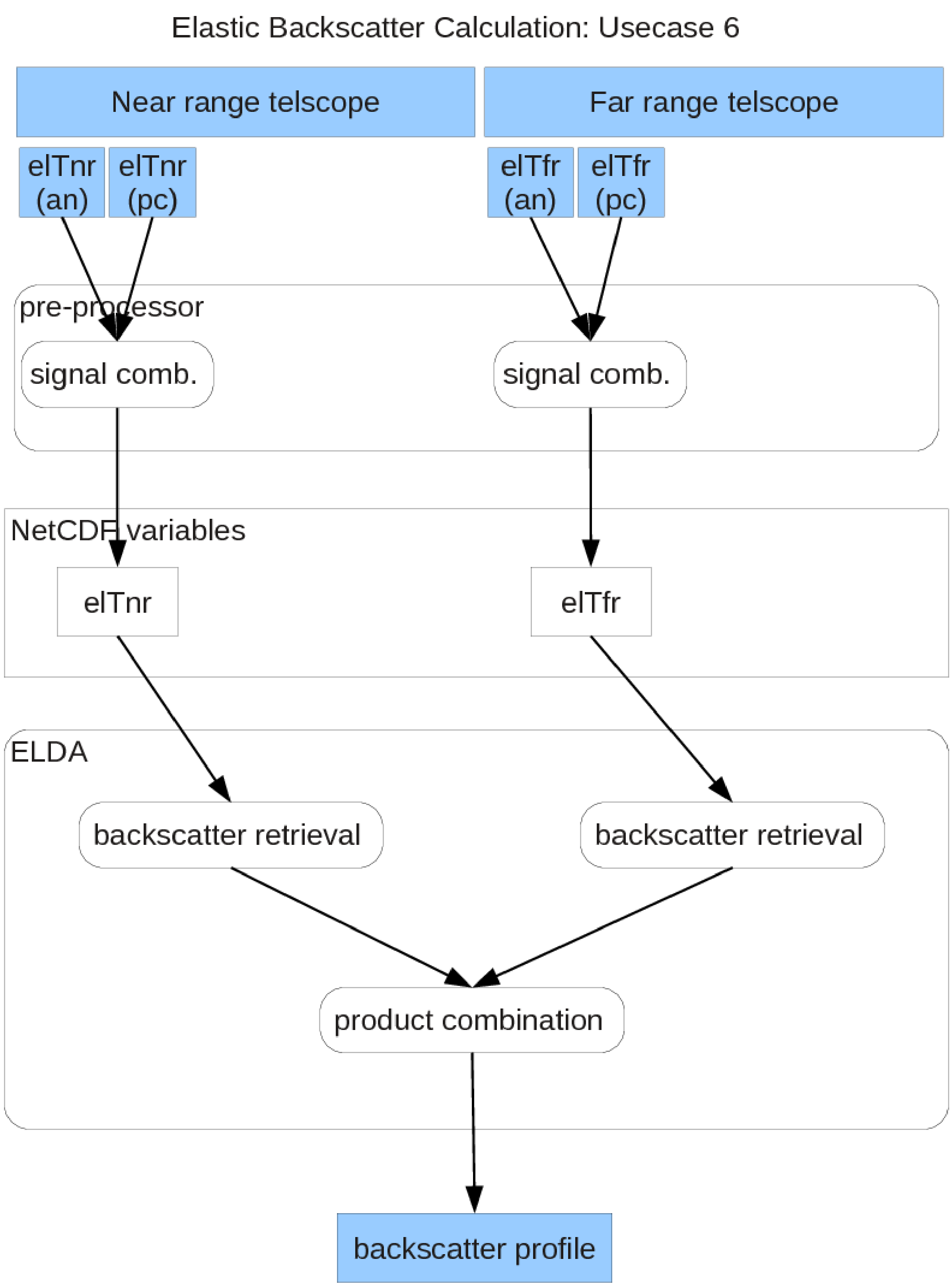

Elastic Backscatter Calculation: Usecase 1
elTnr
elTfr
pre-processor
signal combination
NetCDF variables
elT
ELDA
backscatter retrieval
backscatter profile

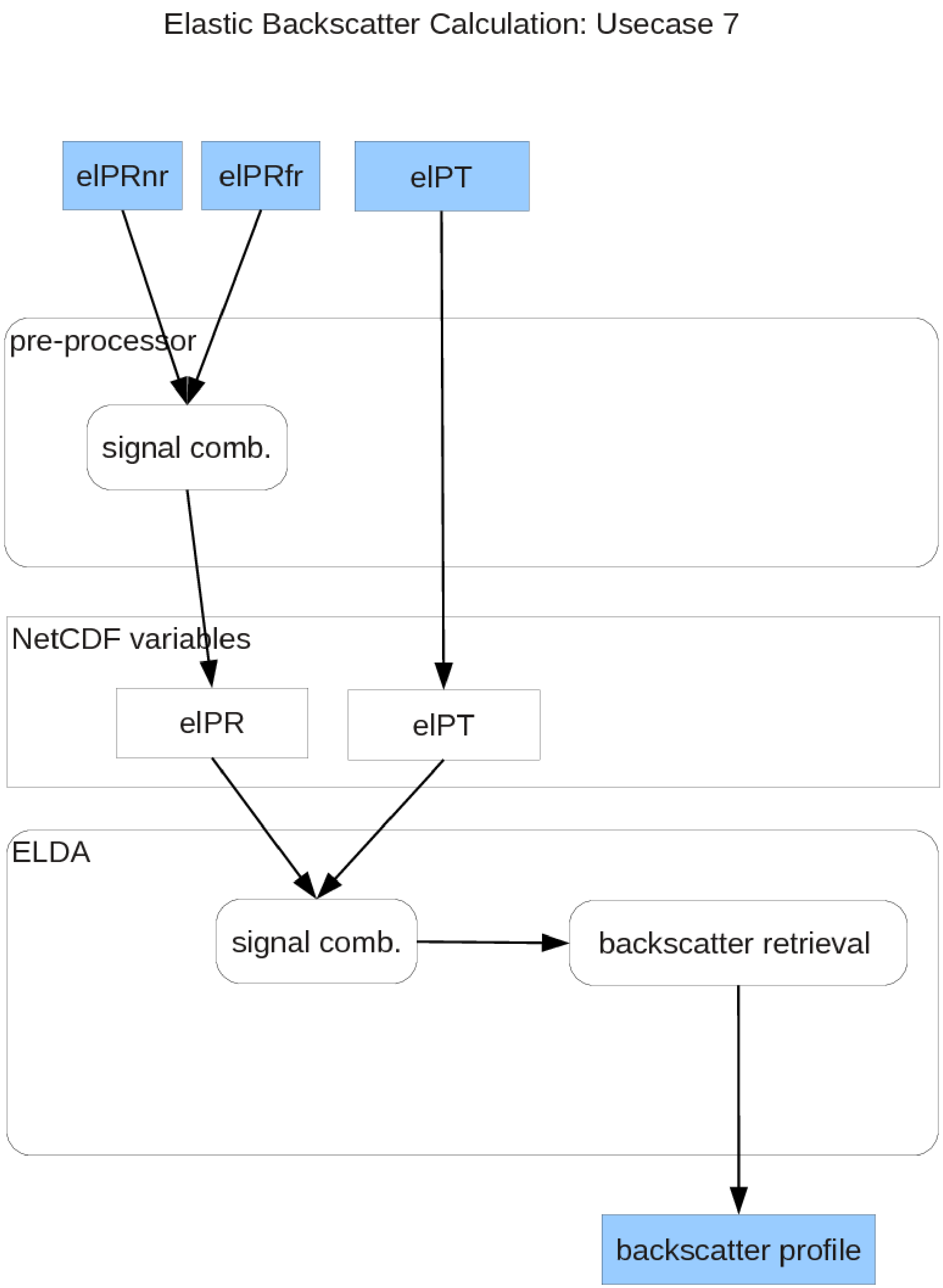

Elastic Backscatter Calculation: Usecase 2
Near range telescope
Far range telescope
elTnr
elTfr
pre-processor
NetCDF variables
elTnr
elTfr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

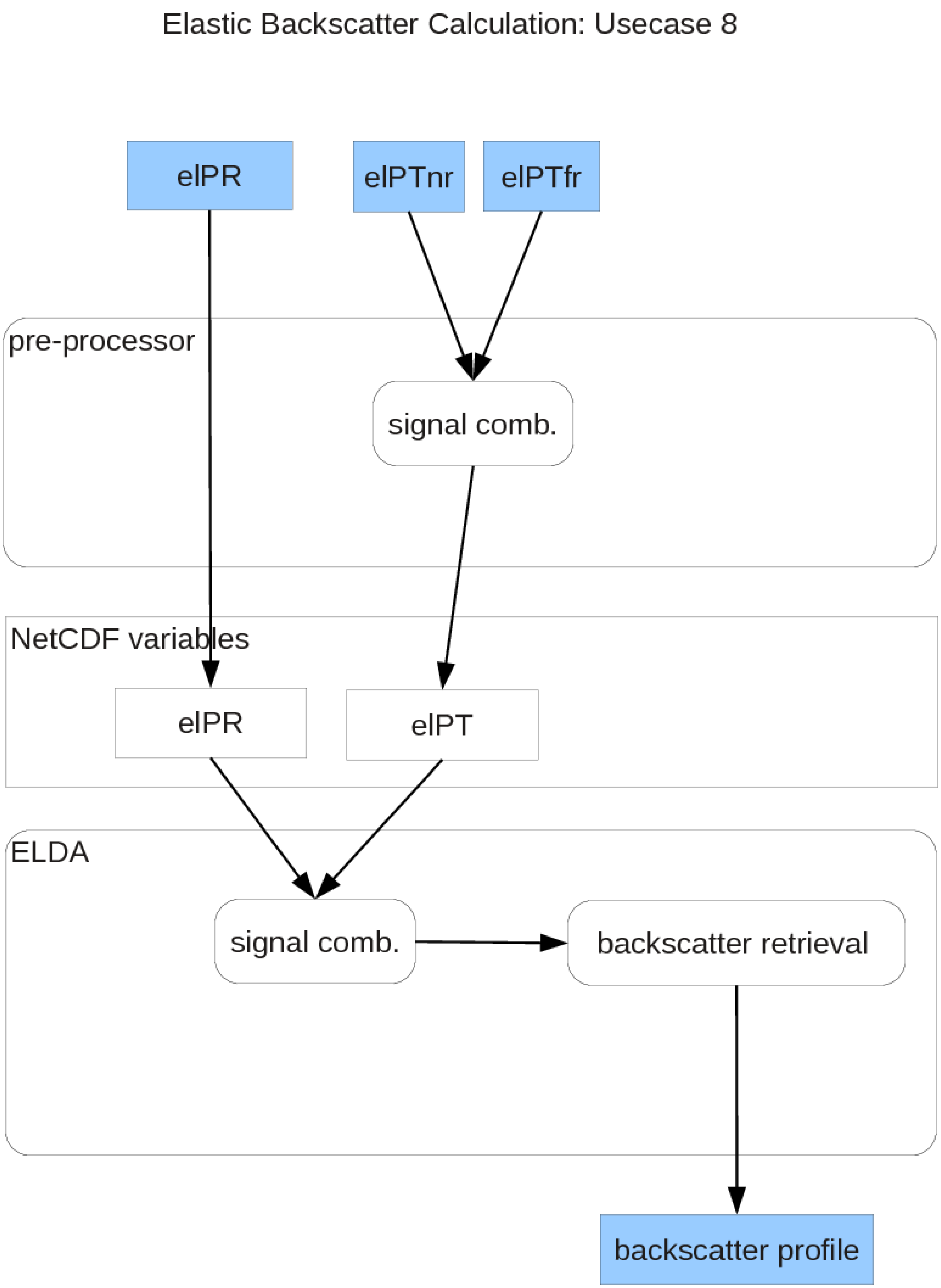

Elastic Backscatter Calculation: Usecase 3
elPR
elPT
pre-processor
NetCDF variables
elPR
elPT
ELDA
signal combinationl
backscatter retrieval
backscatter profile

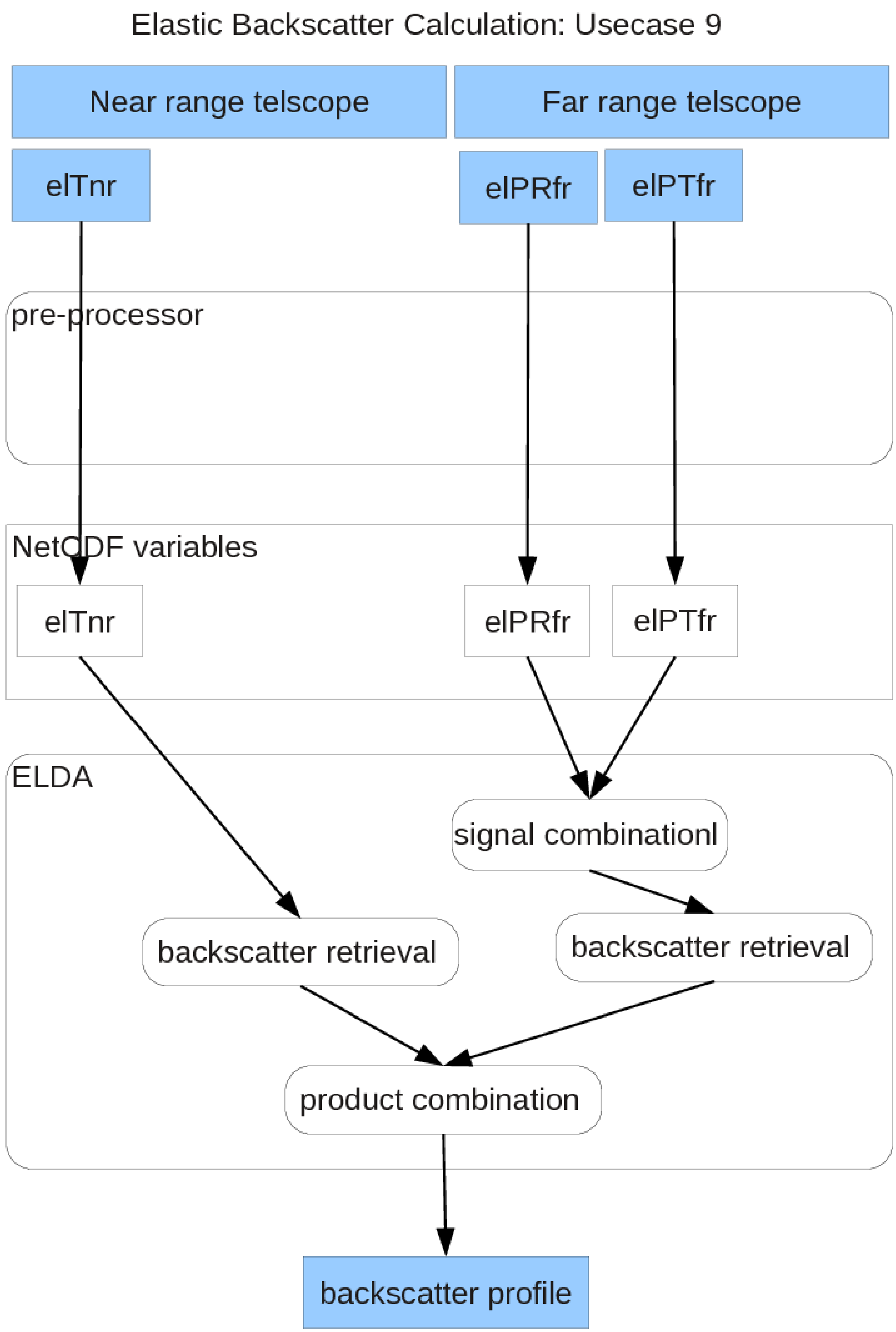

Elastic Backscatter Calculation: Usecase 4
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal combinationl
signal combinationl
NetCDF variables
elPR
elPT
ELDA
signal combinationl
backscatter retrieval
backscatter profile


Elastic Backscatter Calculation: Usecase 5
Near range telscope
Far range telscope
elPRnr
elPTnr
elPRfr
elPTfr
pre-processor
NetCDF variables
elPRnr
elPTnr
elPRfr
elPTfr
ELDA
signal combinationl
signal combinationl
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

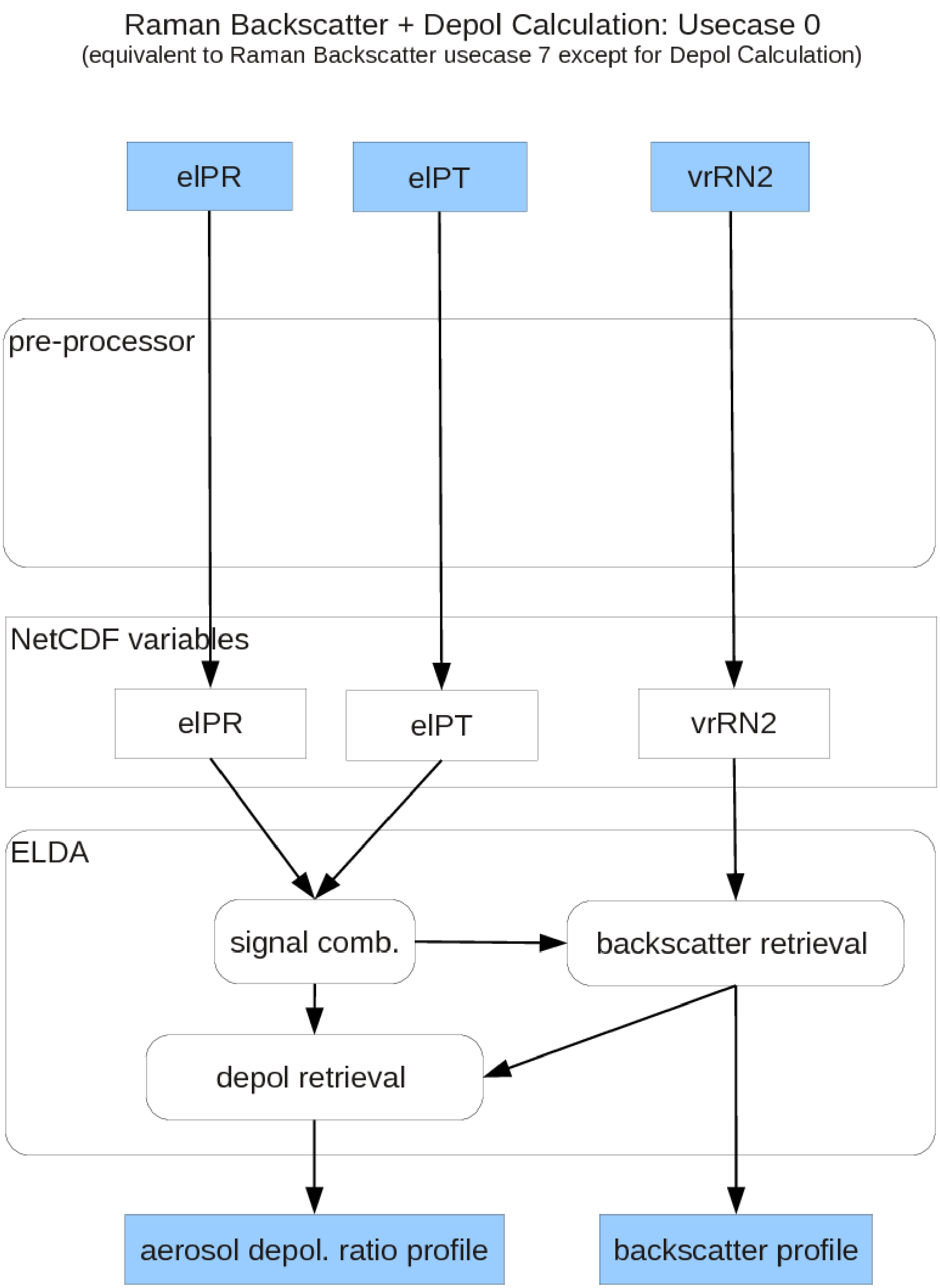

Elastic Backscatter Calculation: Usecase 6
Near range telscope
Far range telscope
elTnr (an)
elTnr (pc)
elTfr (an)
elTfr (pc)
pre-processor
signal comb.
signal comb.
NetCDF variables
elTnr
elTfr
ELDA
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

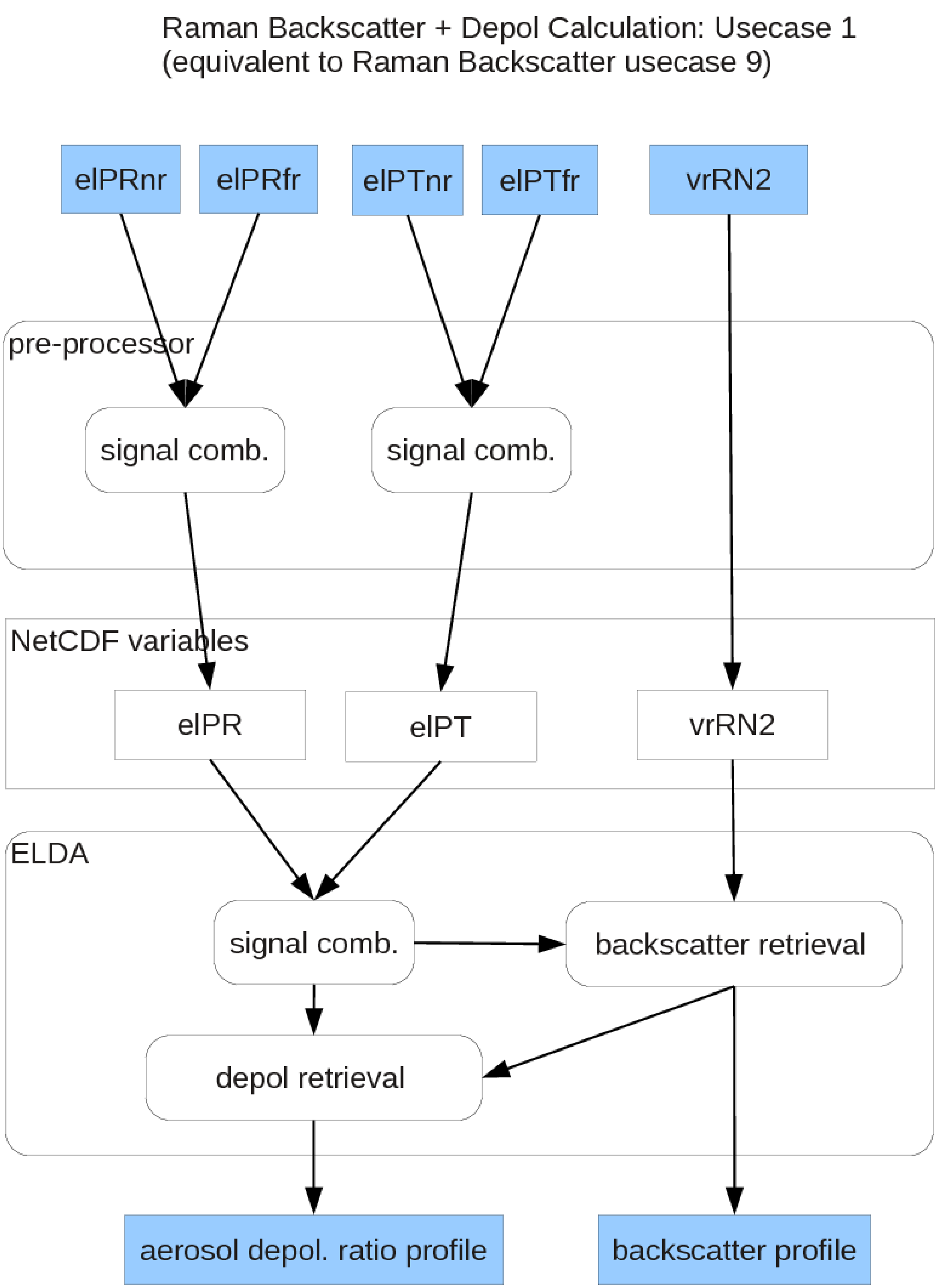

Elastic Backscatter Calculation: Usecase 7
elPRnr
elPRfr
elPT
pre-processor
signal comb.
NetCDF variables
elPR
elPT
ELDA
signal comb.
backscatter retrieval
backscatter profile

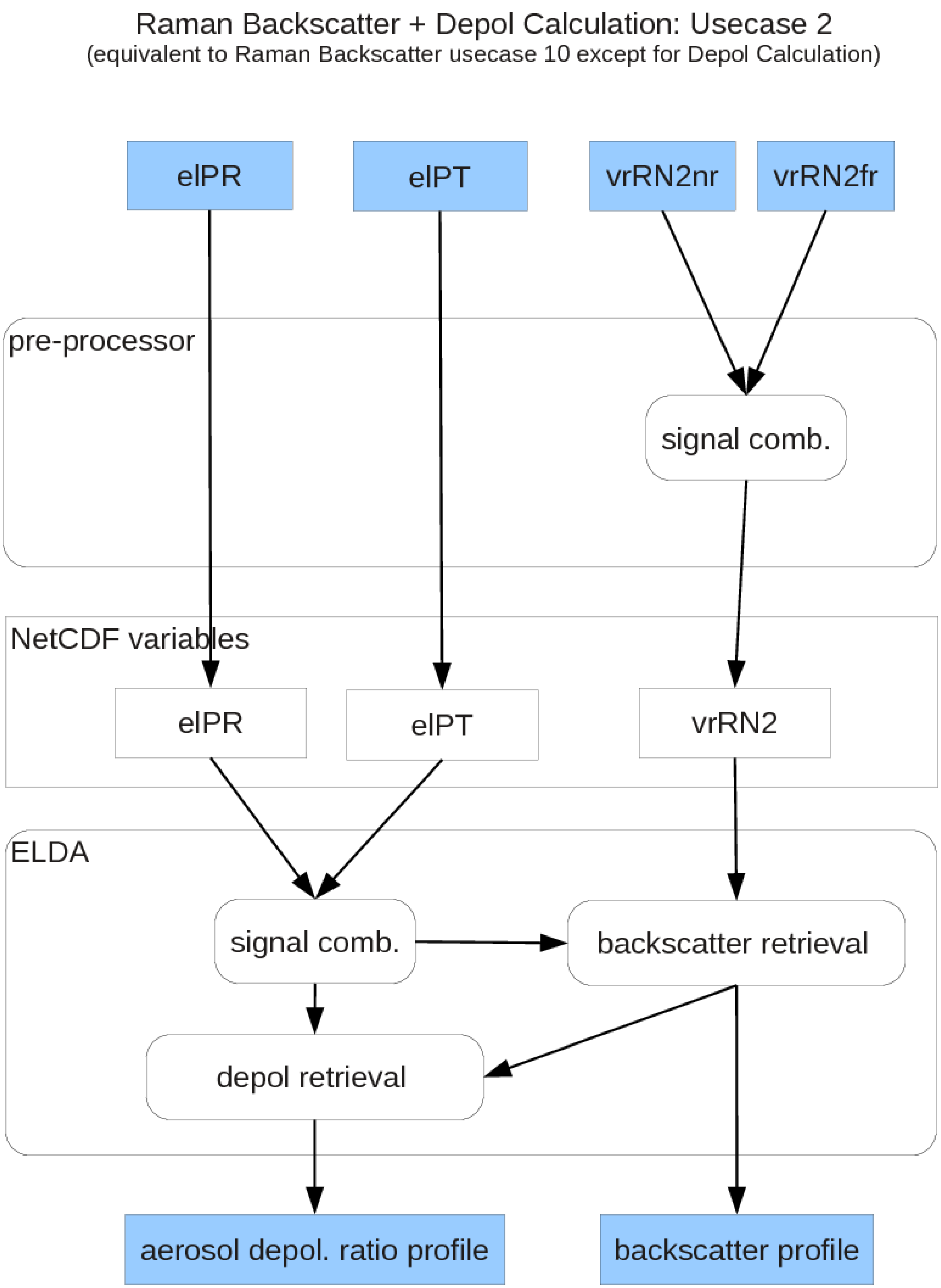

Elastic Backscatter Calculation: Usecase 8
elPR
elPTnr
elPTfr
pre-processor
signal comb.
NetCDF variables
elPR
elPT
ELDA
signal comb.
backscatter retrieval
backscatter profile

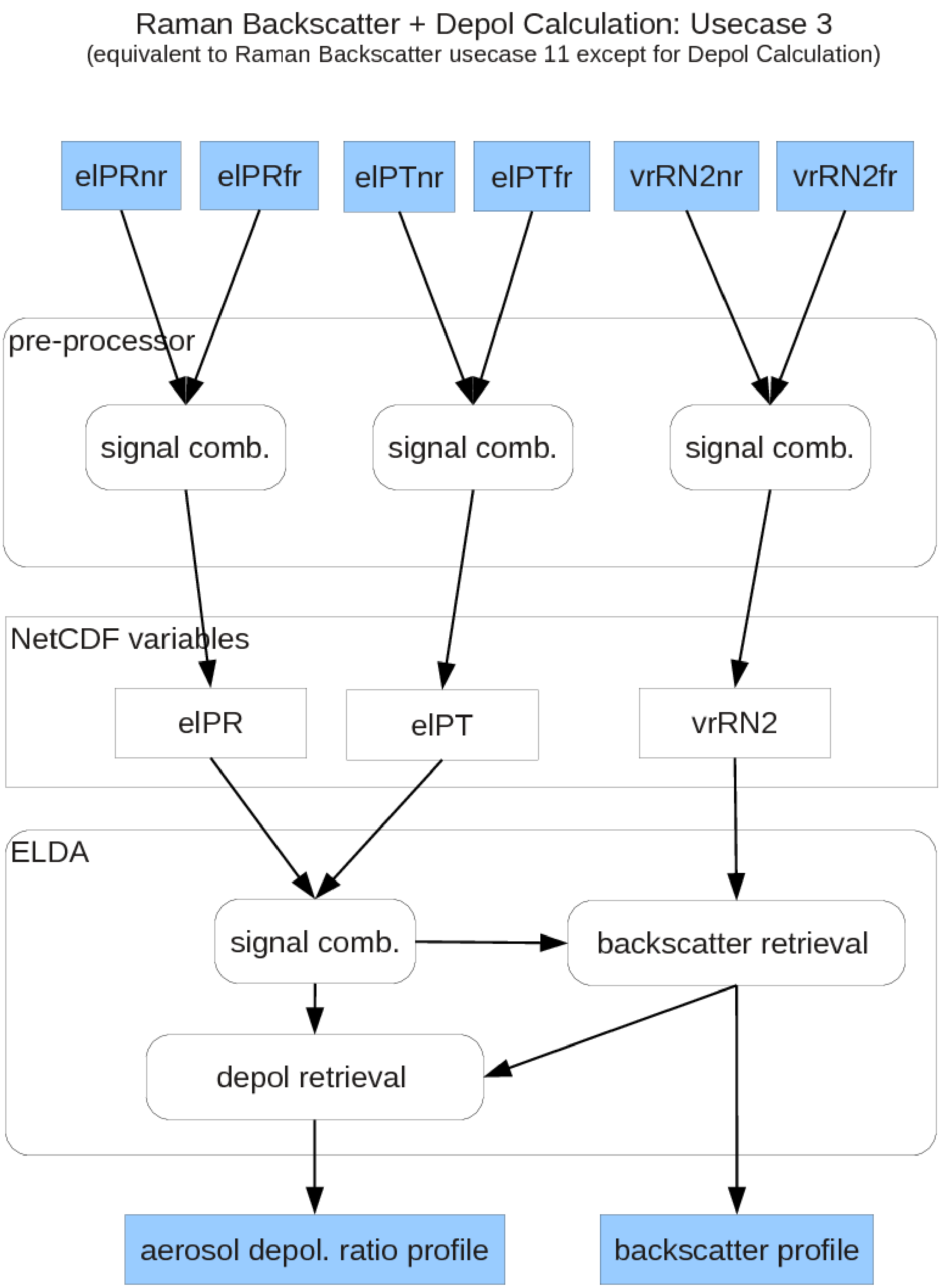

Elastic Backscatter Calculation: Usecase 9
Near range telscope
Far range telscope
elTnr
elPRfr
elPTfr
pre-processor
NetCDF variables
elTnr
elPRfr
elPTfr
ELDA
signal combinationl
backscatter retrieval
backscatter retrieval
product combination
backscatter profile

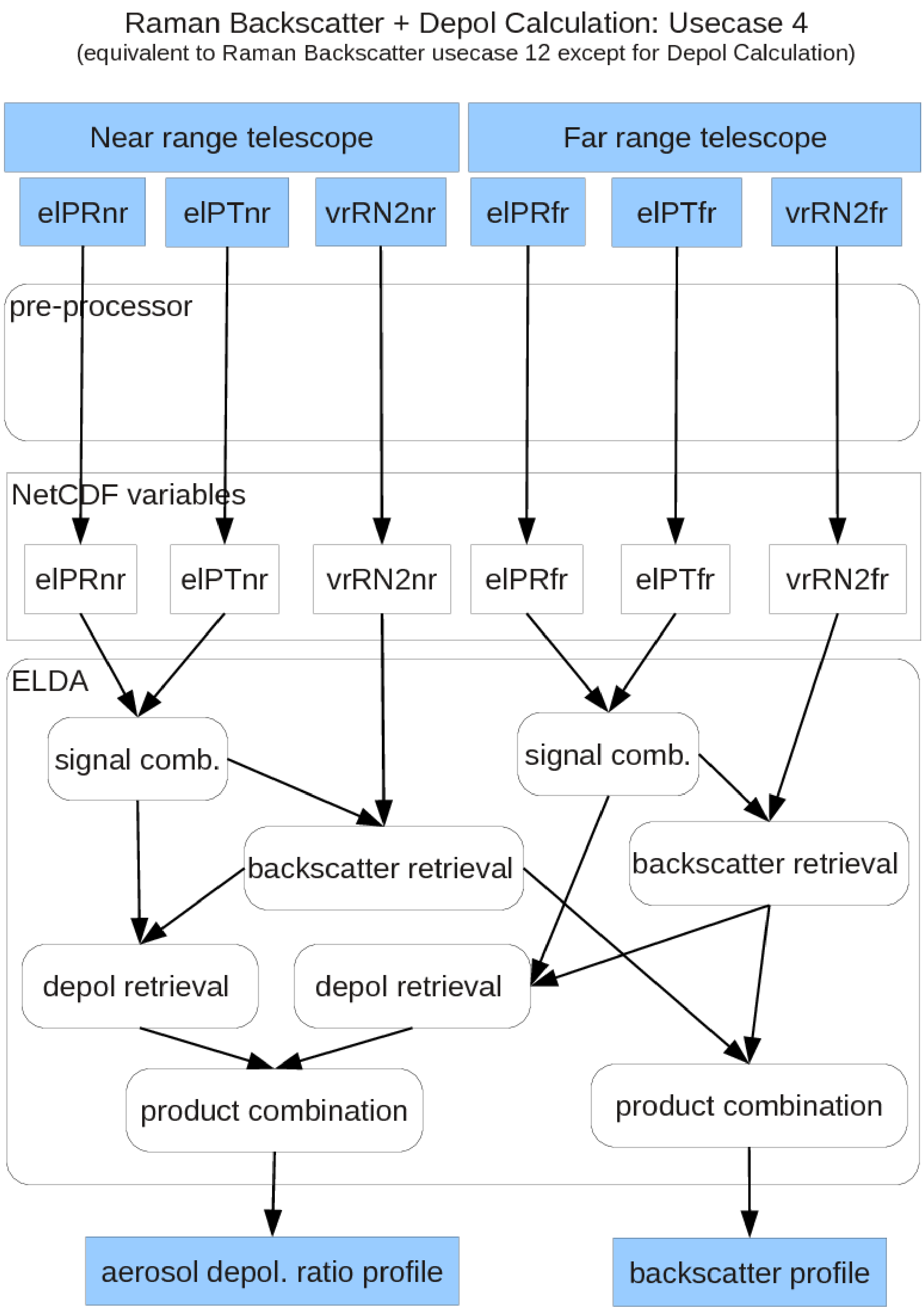
3.6.4 Raman backscatter and depolarization

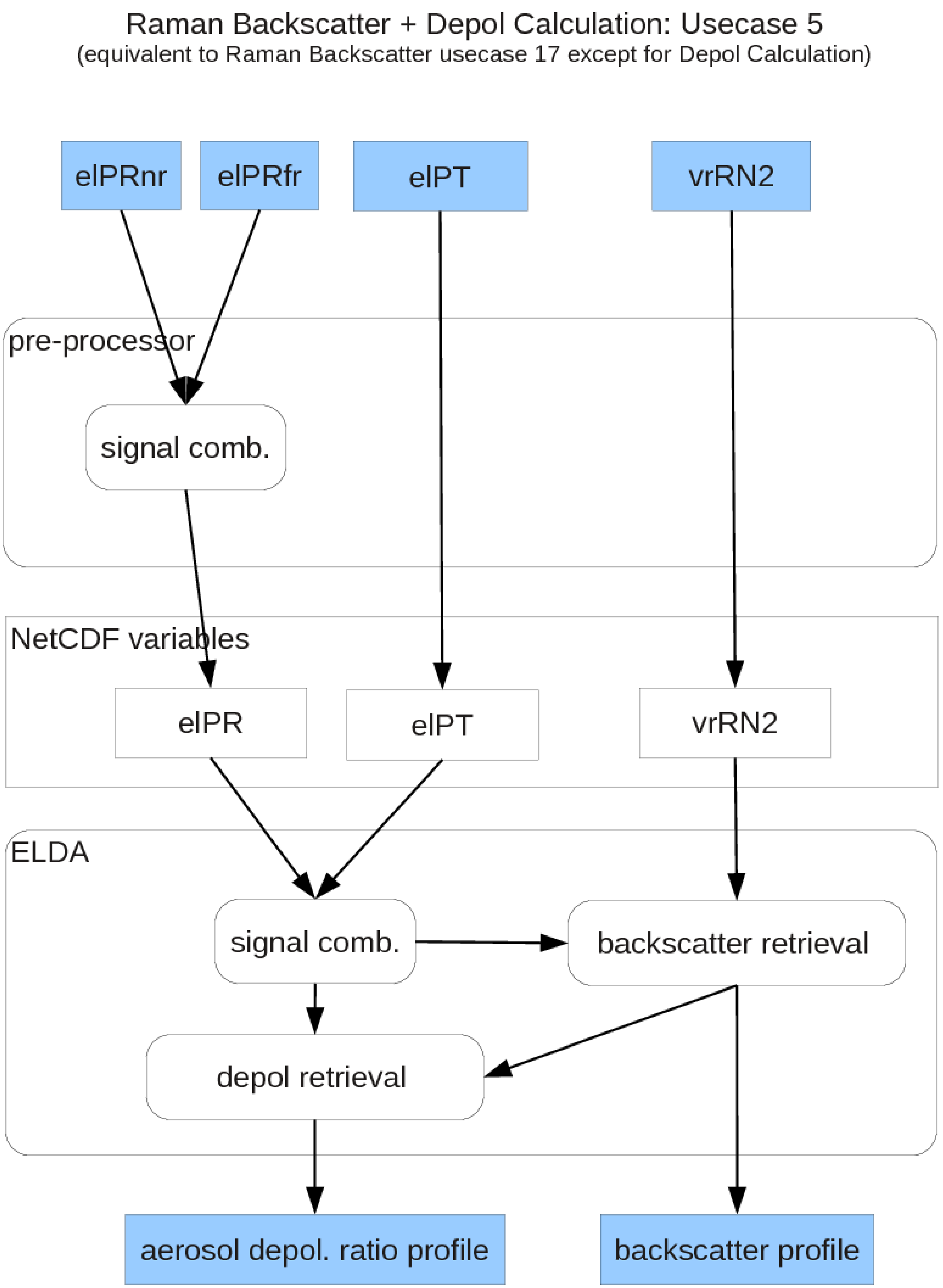


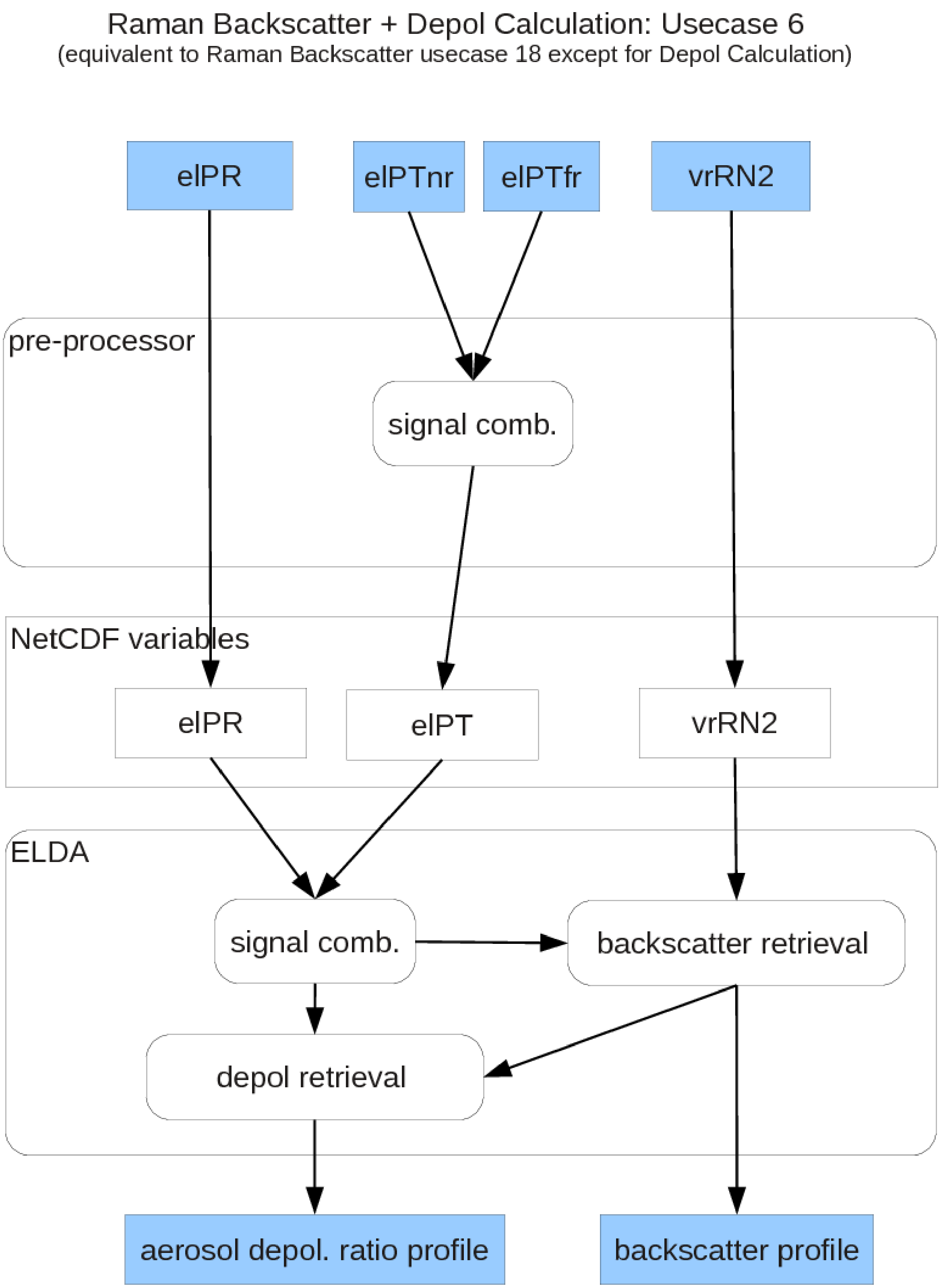

Raman Backscatter + Depol Calculation: Usecase 1
(equivalent to Raman Backscatter usecase 9)
elPRnr
elPRfr
elPTnr
elPTfr
vrRN2
pre-processor
signal comb.
signal comb.
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

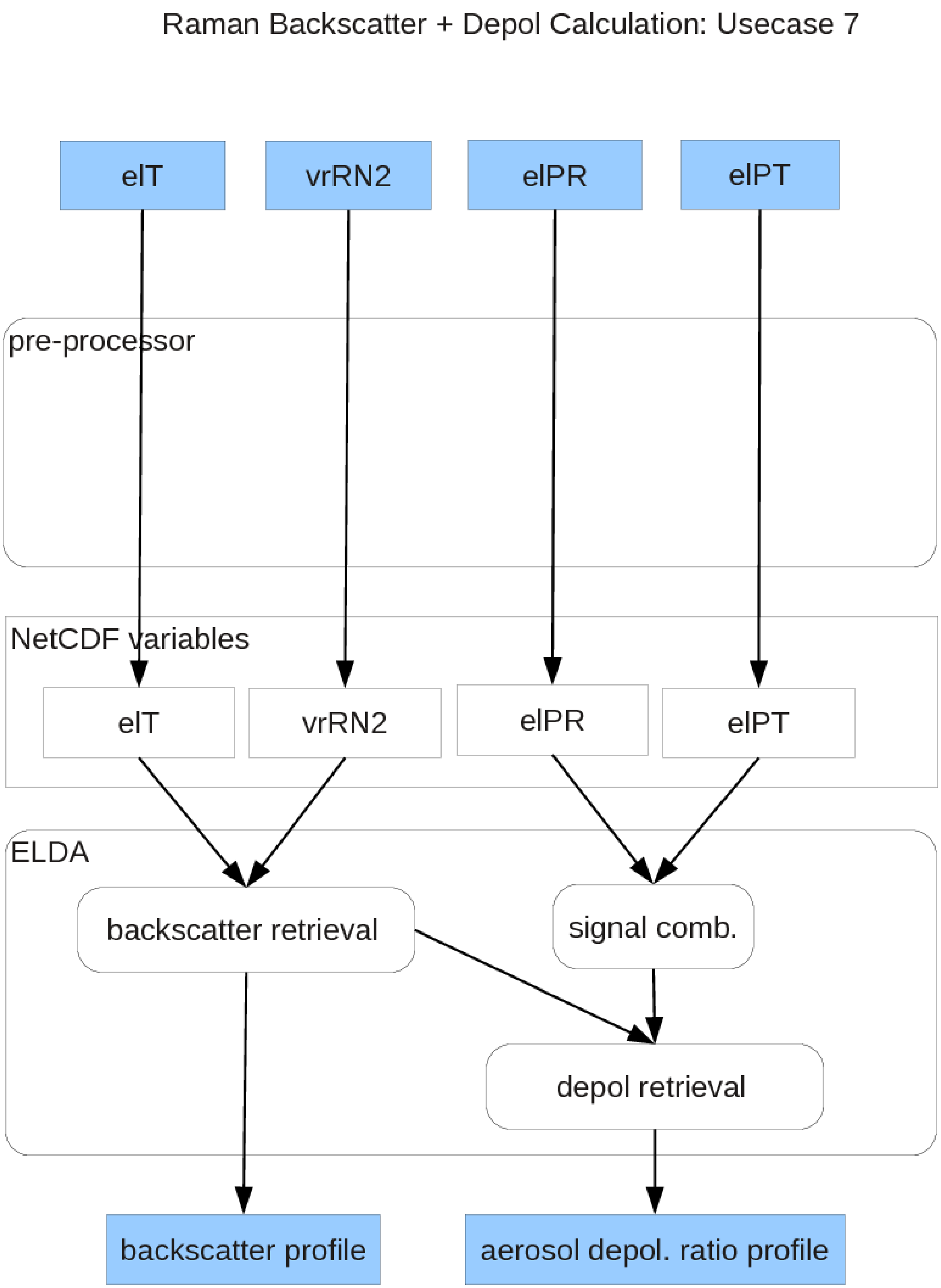

Raman Backscatter + Depol Calculation: Usecase 2
(equivalent to Raman Backscatter usecase 10 except for Depol Calculation)
elPR
elPT
vrRN2nr
vrRN2fr
pre-processor
signal comb.
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

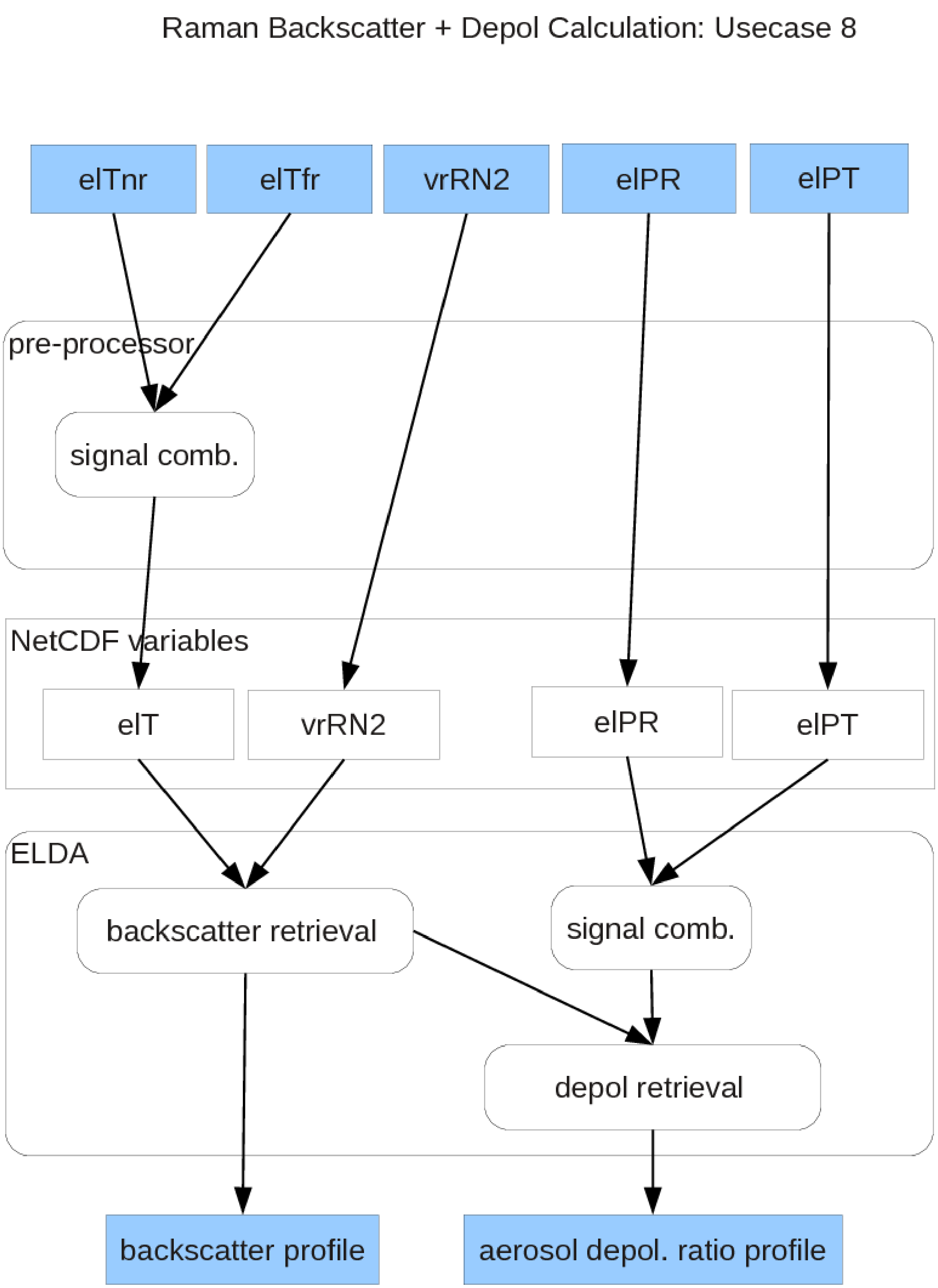

Raman Backscatter + Depol Calculation: Usecase 3
(equivalent to Raman Backscatter usecase 11 except for Depol Calculation)
elPRnr
elPRfr
elPTnr
elPTfr
vrRN2nr
vrRN2fr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

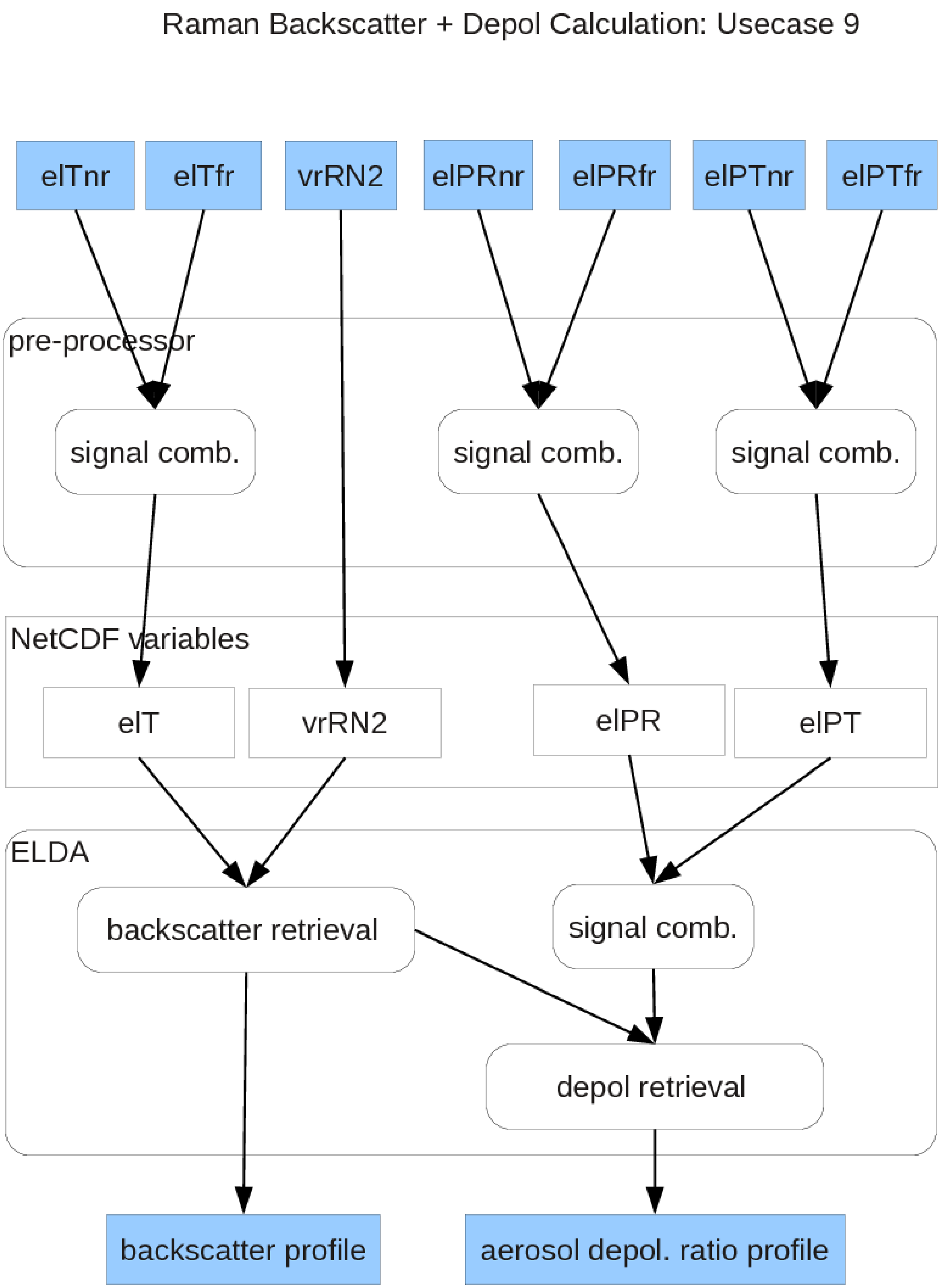

Raman Backscatter + Depol Calculation: Usecase 4
(equivalent to Raman Backscatter usecase 12 except for Depol Calculation)
Near range telescope
Far range telescope
elPRnr
elPTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
pre-processor
NetCDF variables
elPRnr
elPTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
ELDA
signal comb.
signal comb.
backscatter retrieval
backscatter retrieval
depol retrieval
depol retrieval
product combination
product combination
aerosol depol. ratio profile
backscatter profile

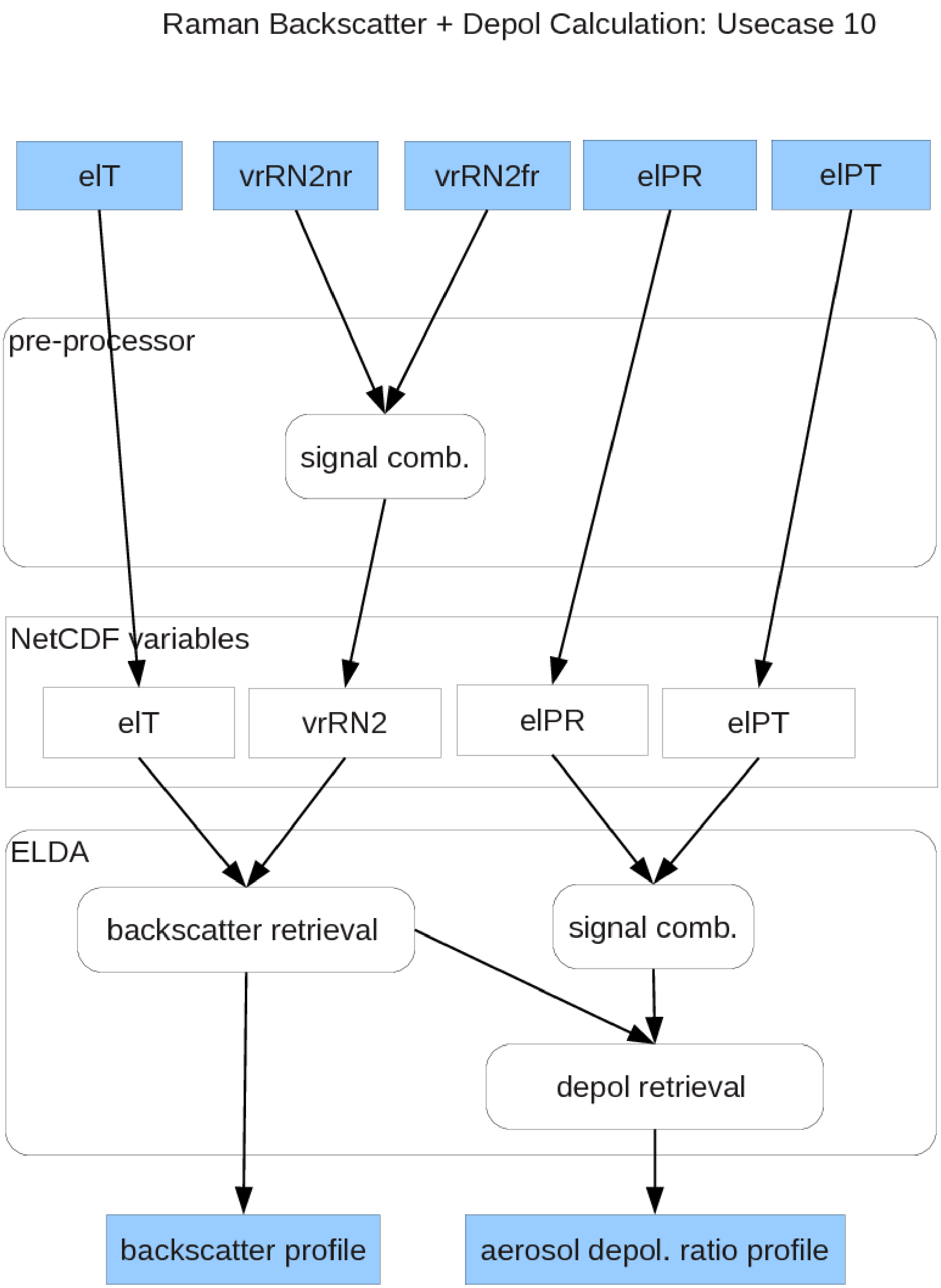

Raman Backscatter + Depol Calculation: Usecase 5
(equivalent to Raman Backscatter usecase 17 except for Depol Calculation)
elPRnr
elPRfr
elPT
vrRN2
pre-processor
signal comb.
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

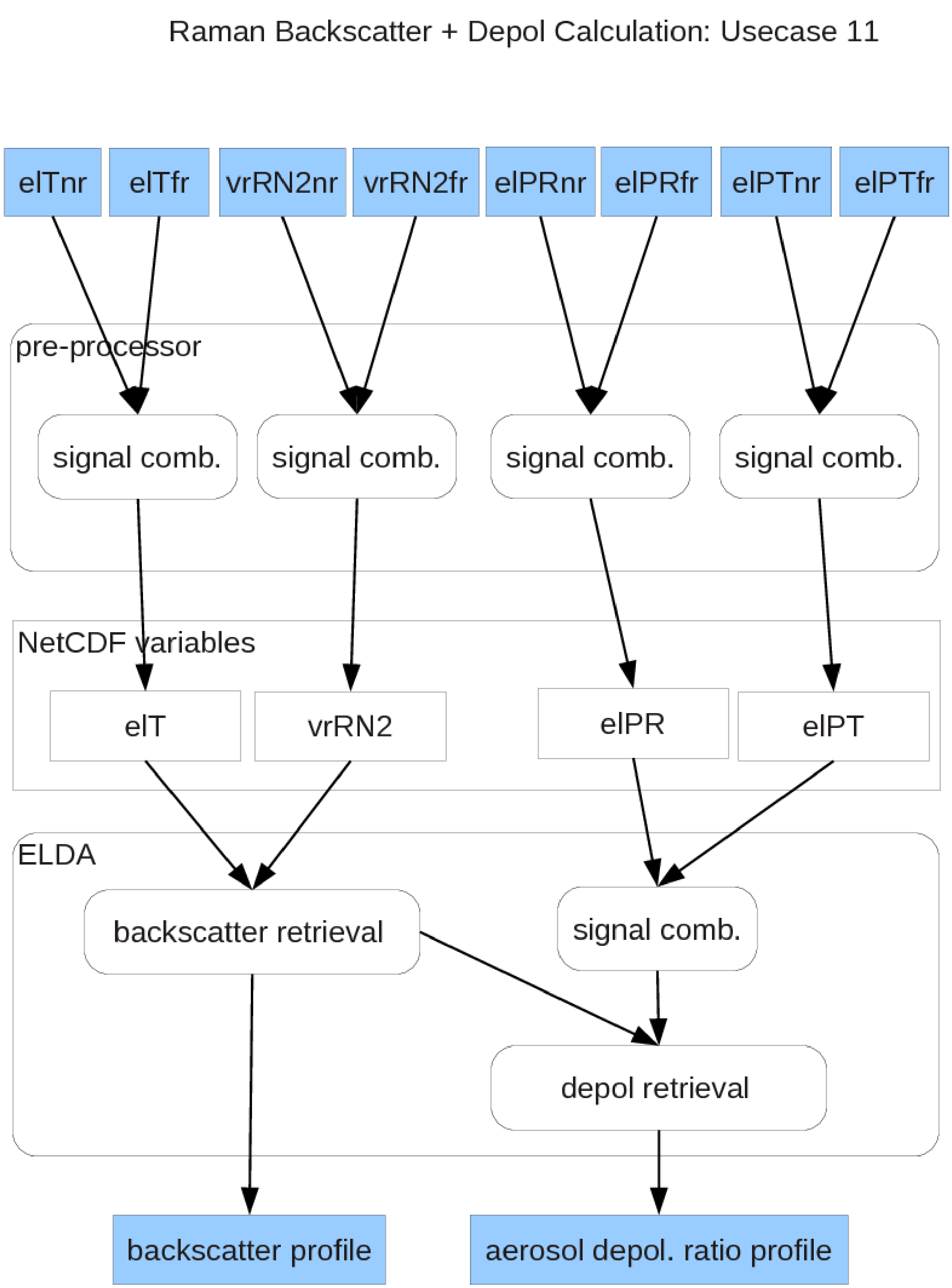

Raman Backscatter + Depol Calculation: Usecase 6
(equivalent to Raman Backscatter usecase 18 except for Depol Calculation)
elPR
elPTnr
elPTfr
vrRN2
pre-processor
signal comb.
NetCDF variables
elPR
elPT
vrRN2
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

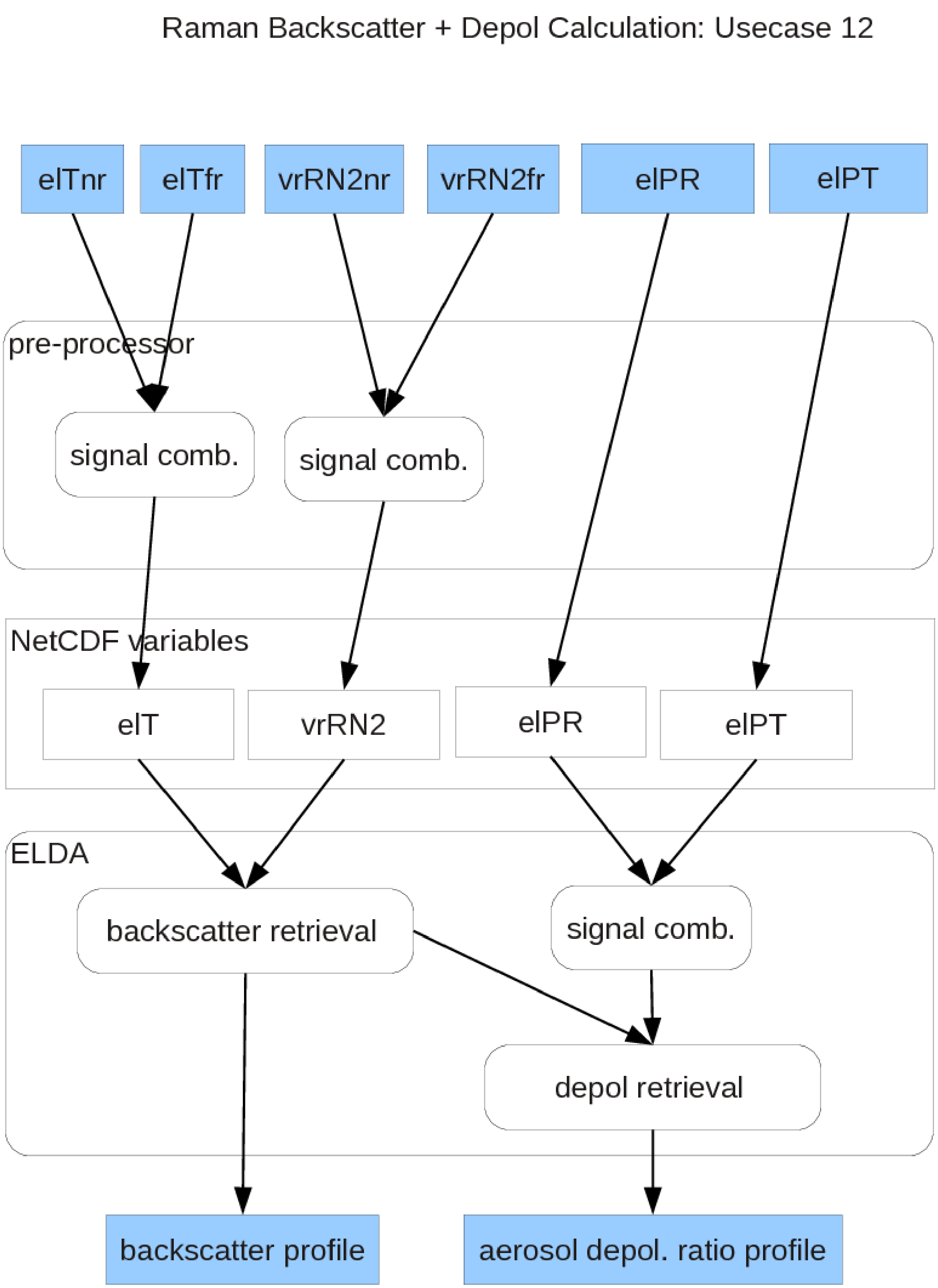

Raman Backscatter + Depol Calculation: Usecase 7
elT
vrRN2
elPR
elPT
pre-processor
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

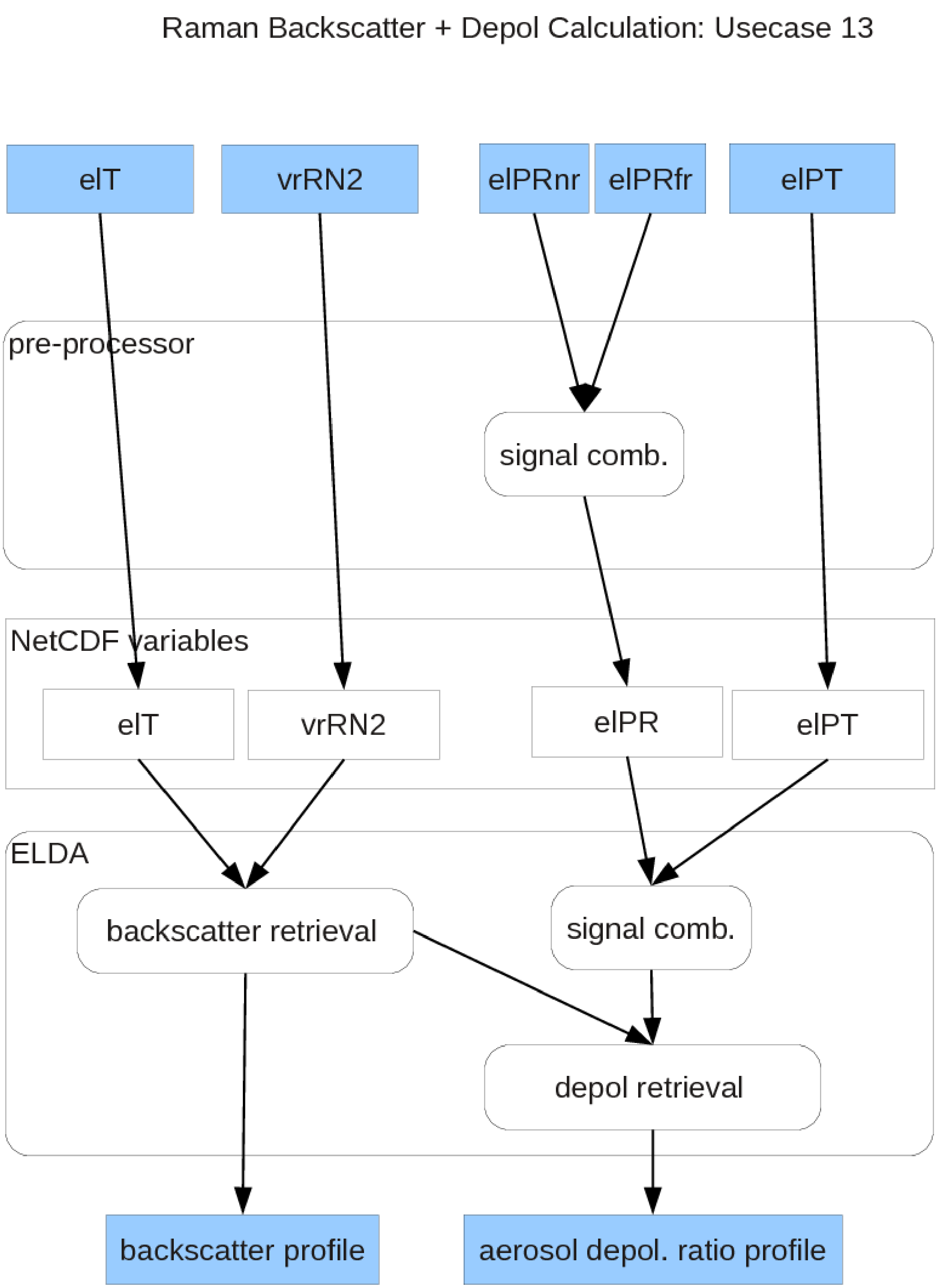

Raman Backscatter + Depol Calculation: Usecase 8
elTnr
elTfr
vrRN2
elPR
elPT
pre-processor
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

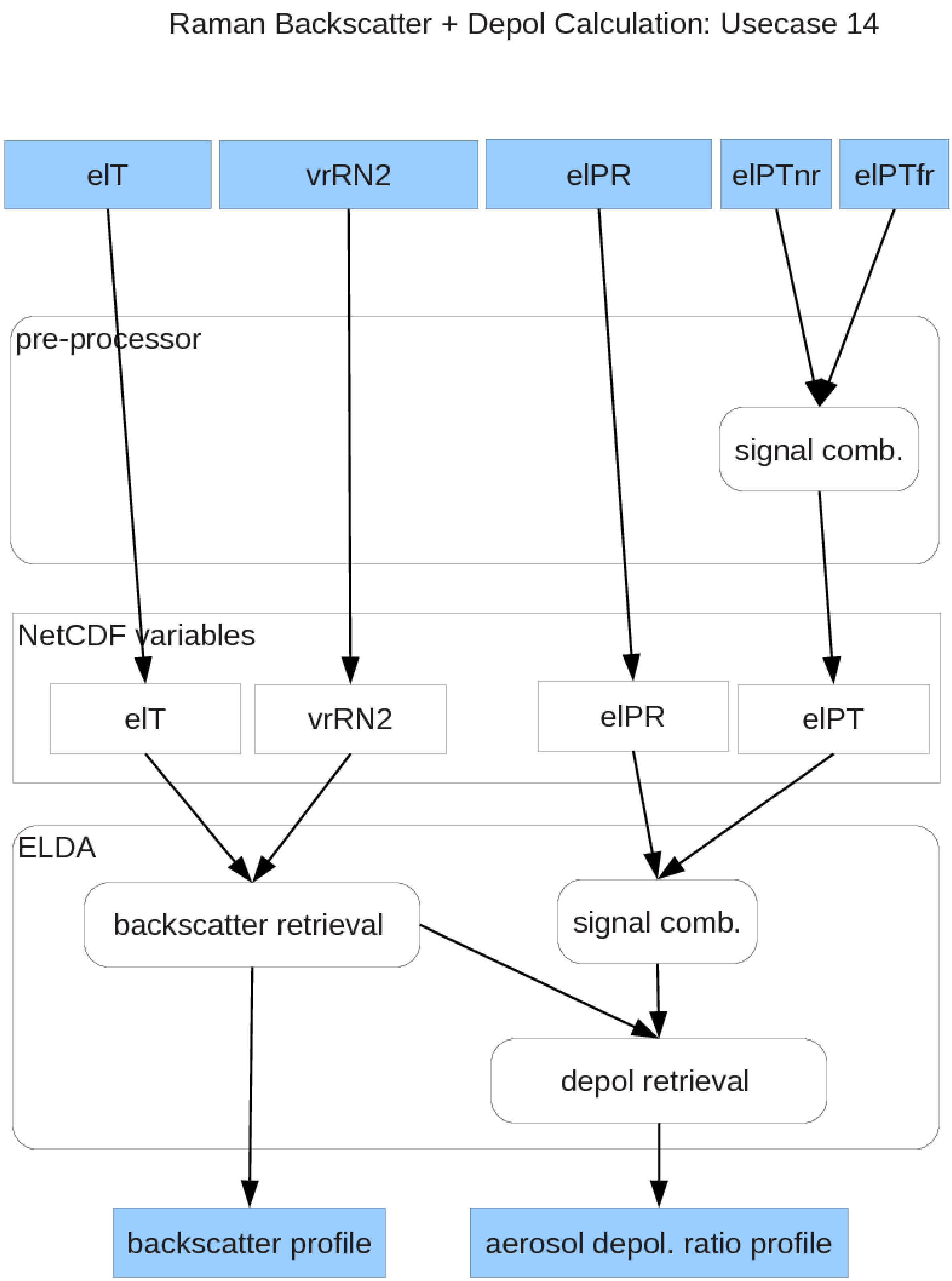

Raman Backscatter + Depol Calculation: Usecase 9
elTnr
elTfr
vrRN2
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

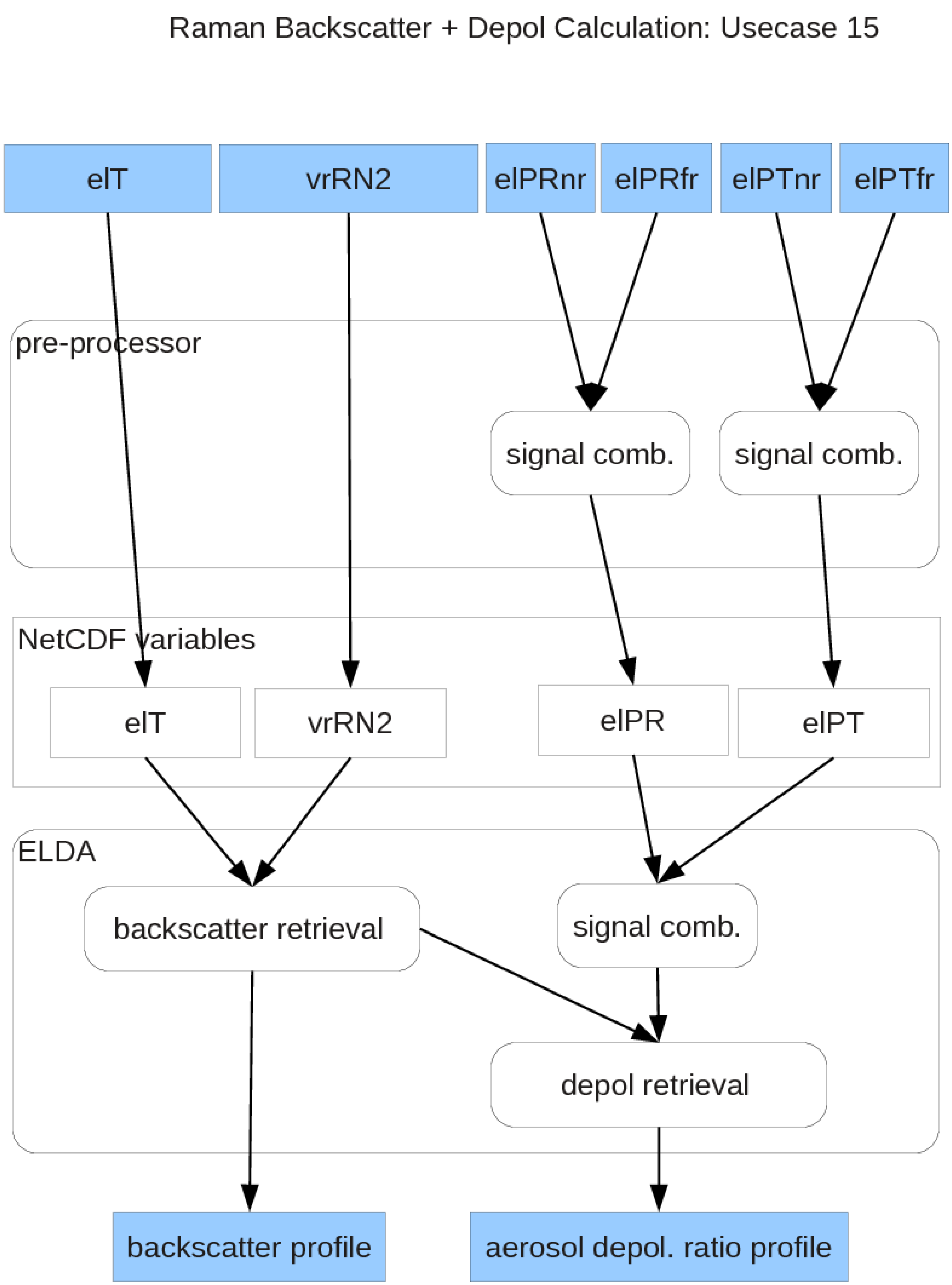

Raman Backscatter + Depol Calculation: Usecase 10
elT
vrRN2nr
vrRN2fr
elPR
elPT
pre-processor
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

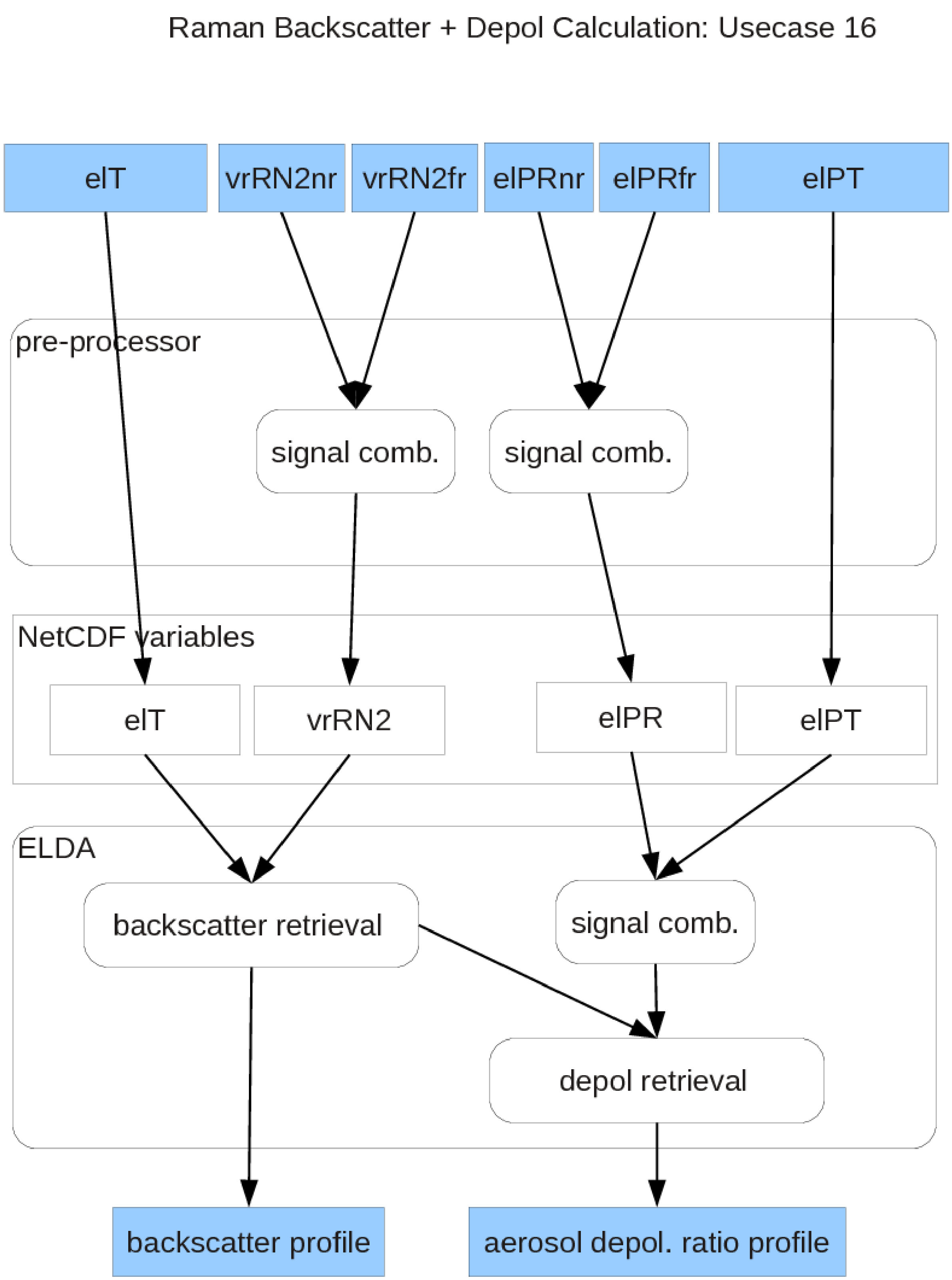

Raman Backscatter + Depol Calculation: Usecase 11
elTnr
elTfr
vrRN2nr
vrRN2fr
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

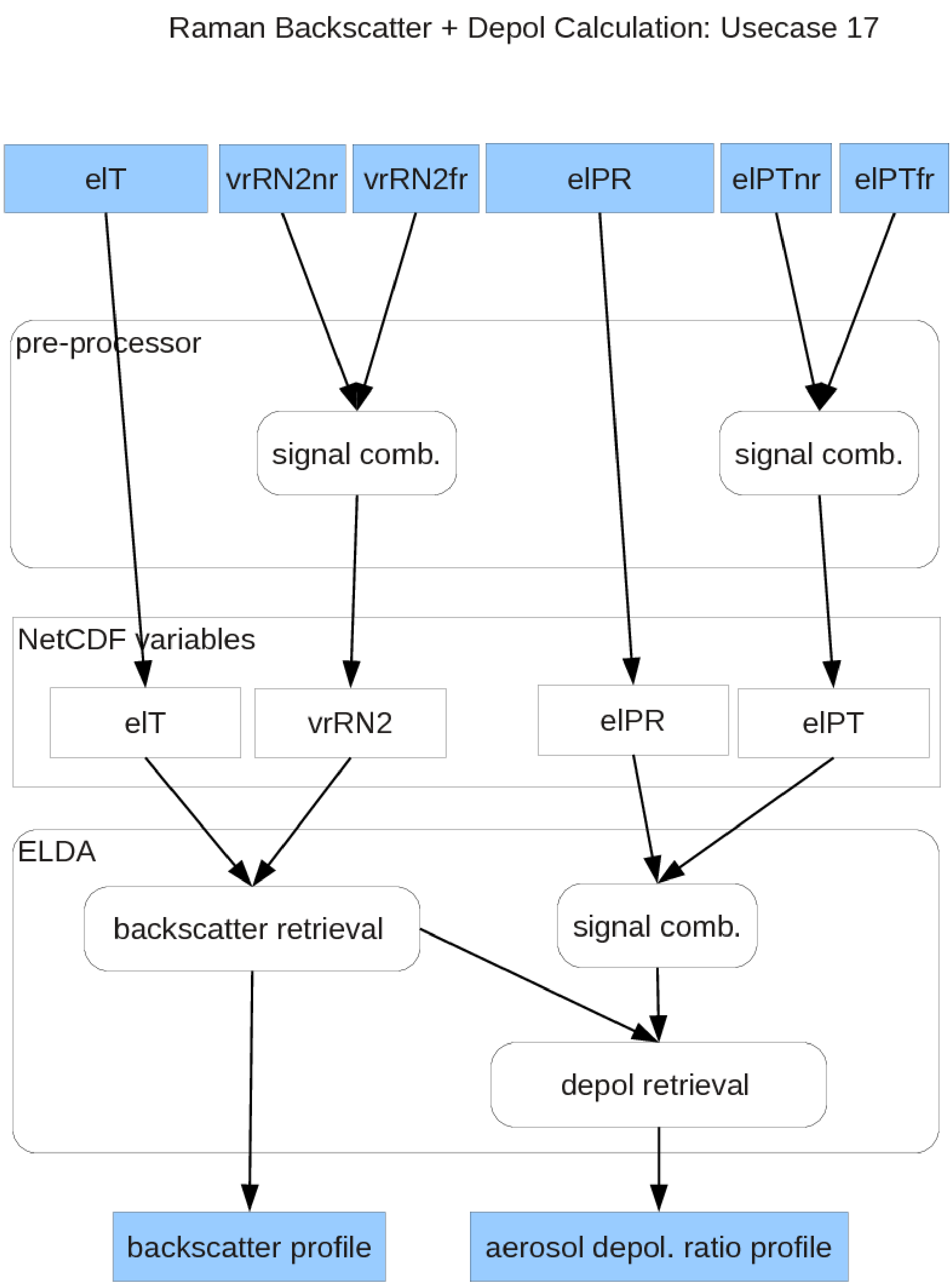

Raman Backscatter + Depol Calculation: Usecase 12
elTnr
elTfr
vrRN2nr
vrRN2fr
elPR
elPT
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

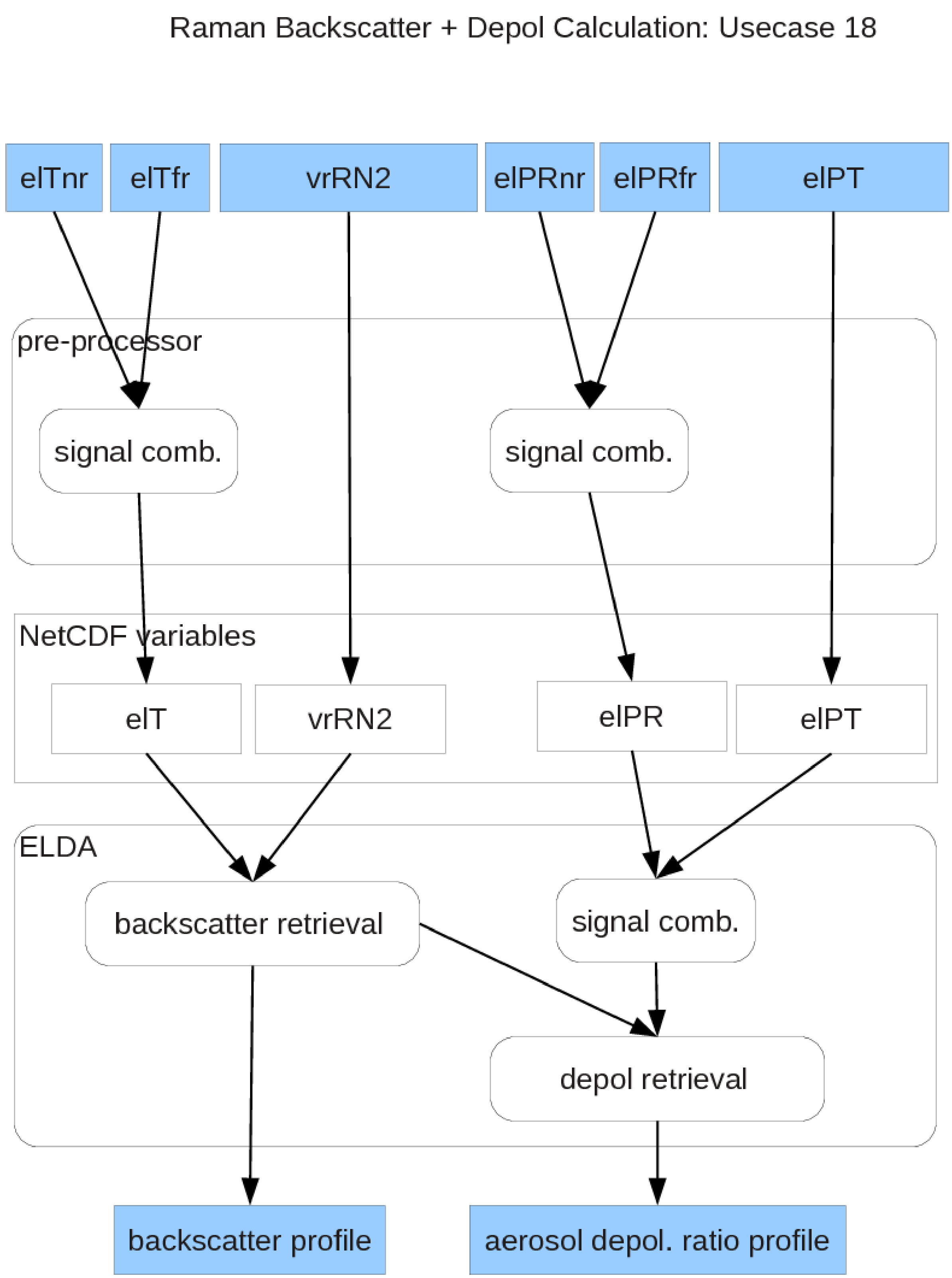

Raman Backscatter + Depol Calculation: Usecase 13
elT
vrRN2
elPRnr
elPRfr
elPT
pre-processor
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

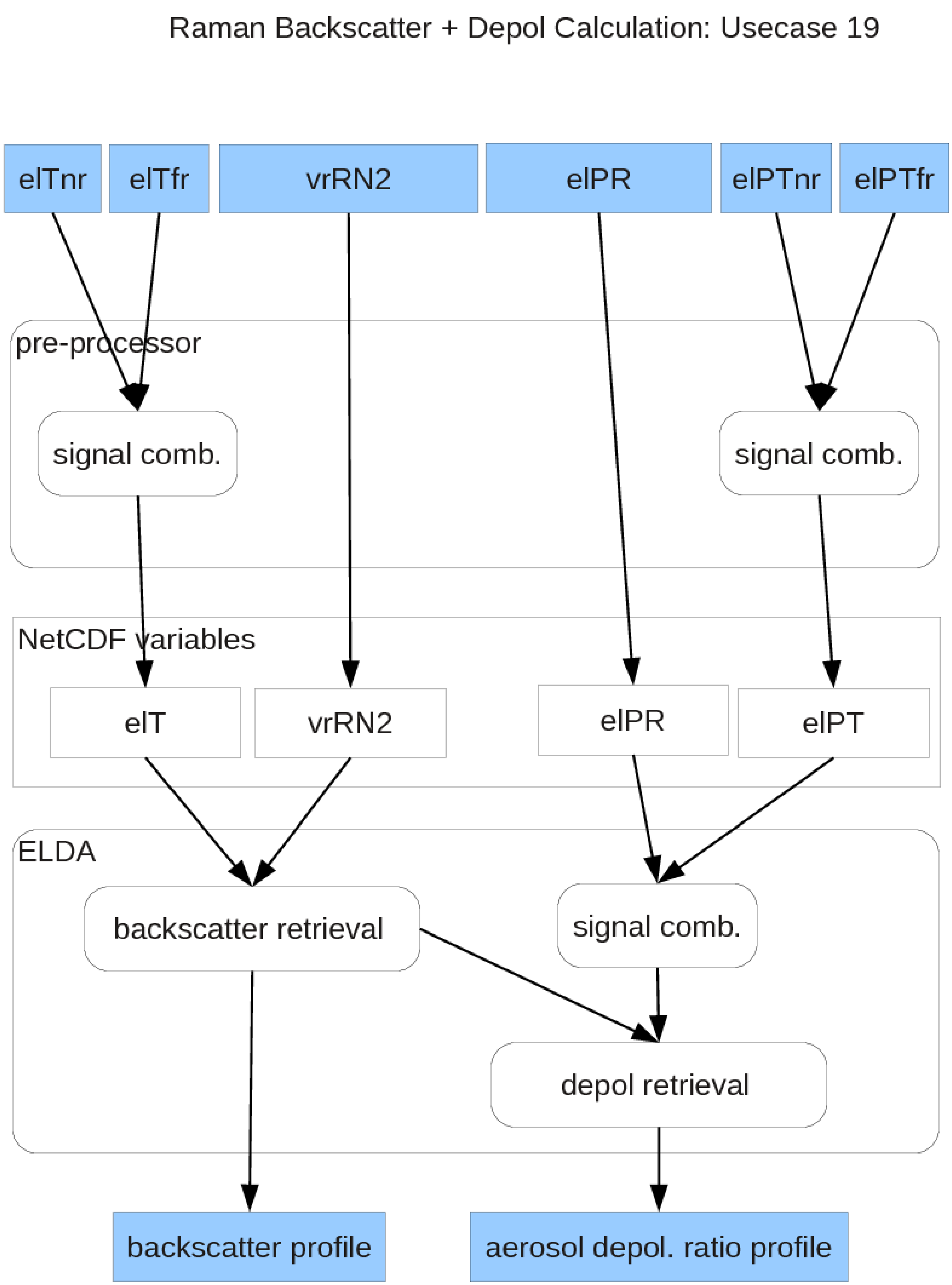

Raman Backscatter + Depol Calculation: Usecase 14
elT
vrRN2
elPR
elPTnr
elPTfr
pre-processor
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

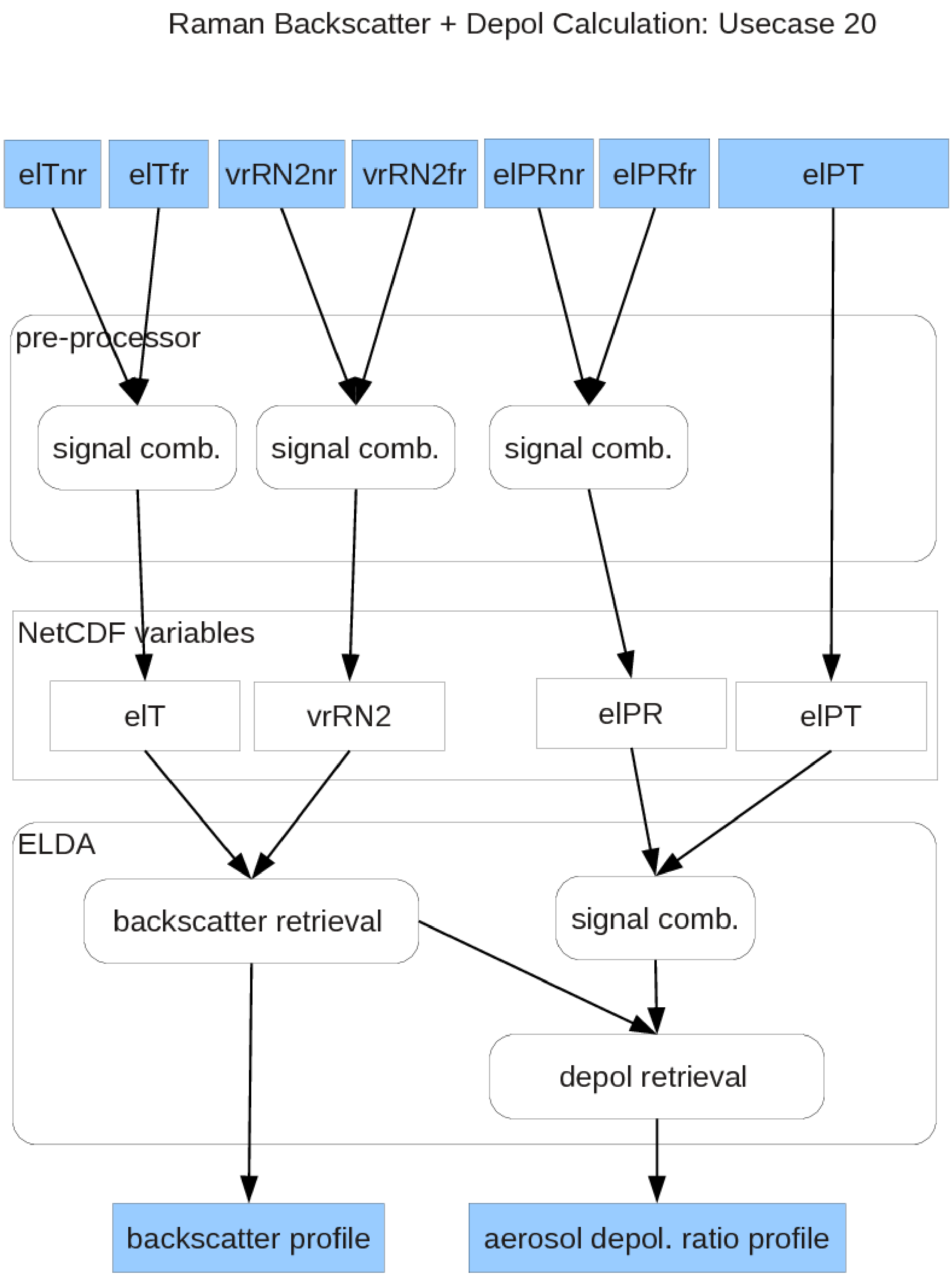

Raman Backscatter + Depol Calculation: Usecase 15
elT
vrRN2
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

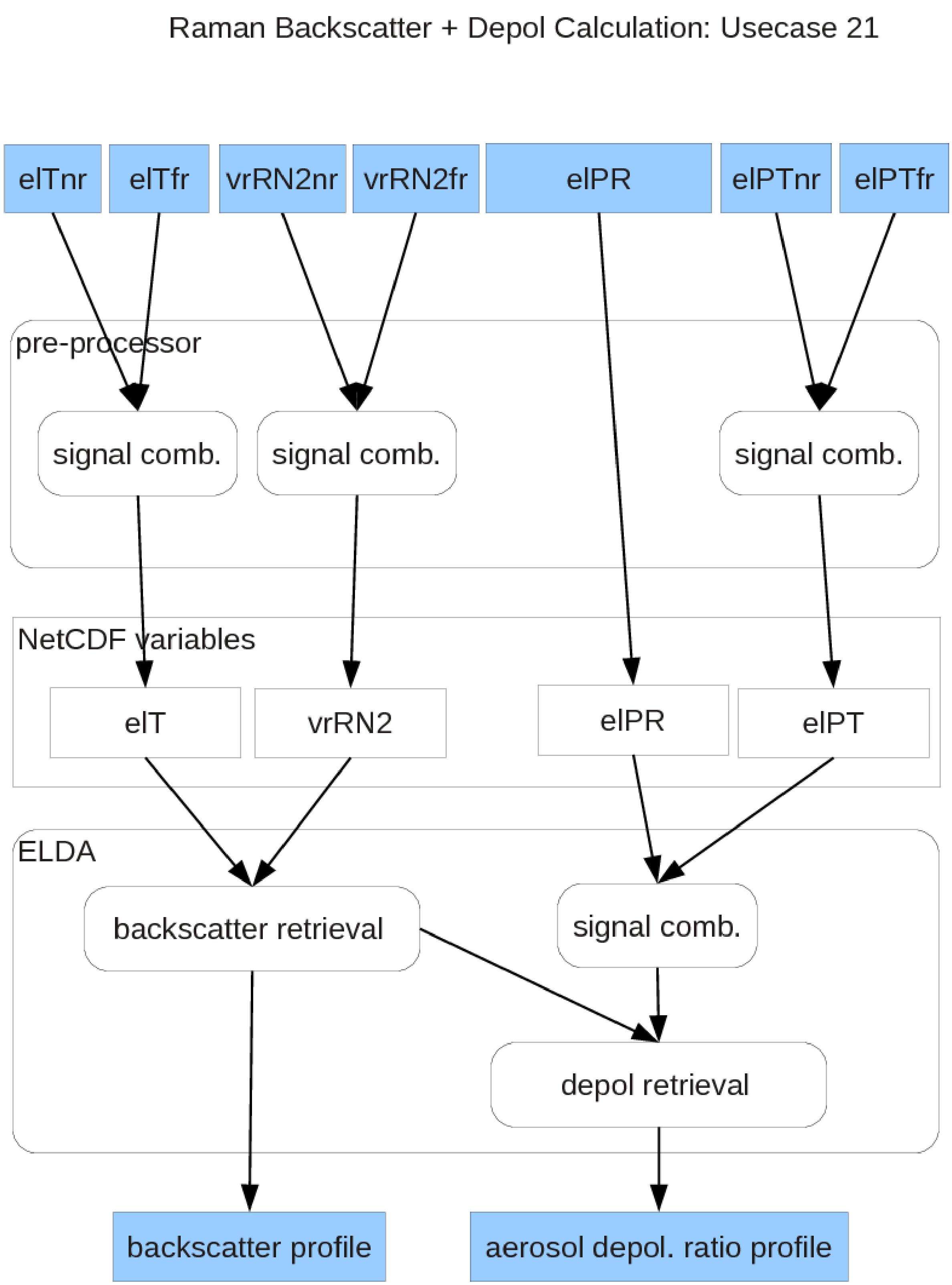

Raman Backscatter + Depol Calculation: Usecase 16
elT
vrRN2nr
vrRN2fr
elPRnr
elPRfr
elPT
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

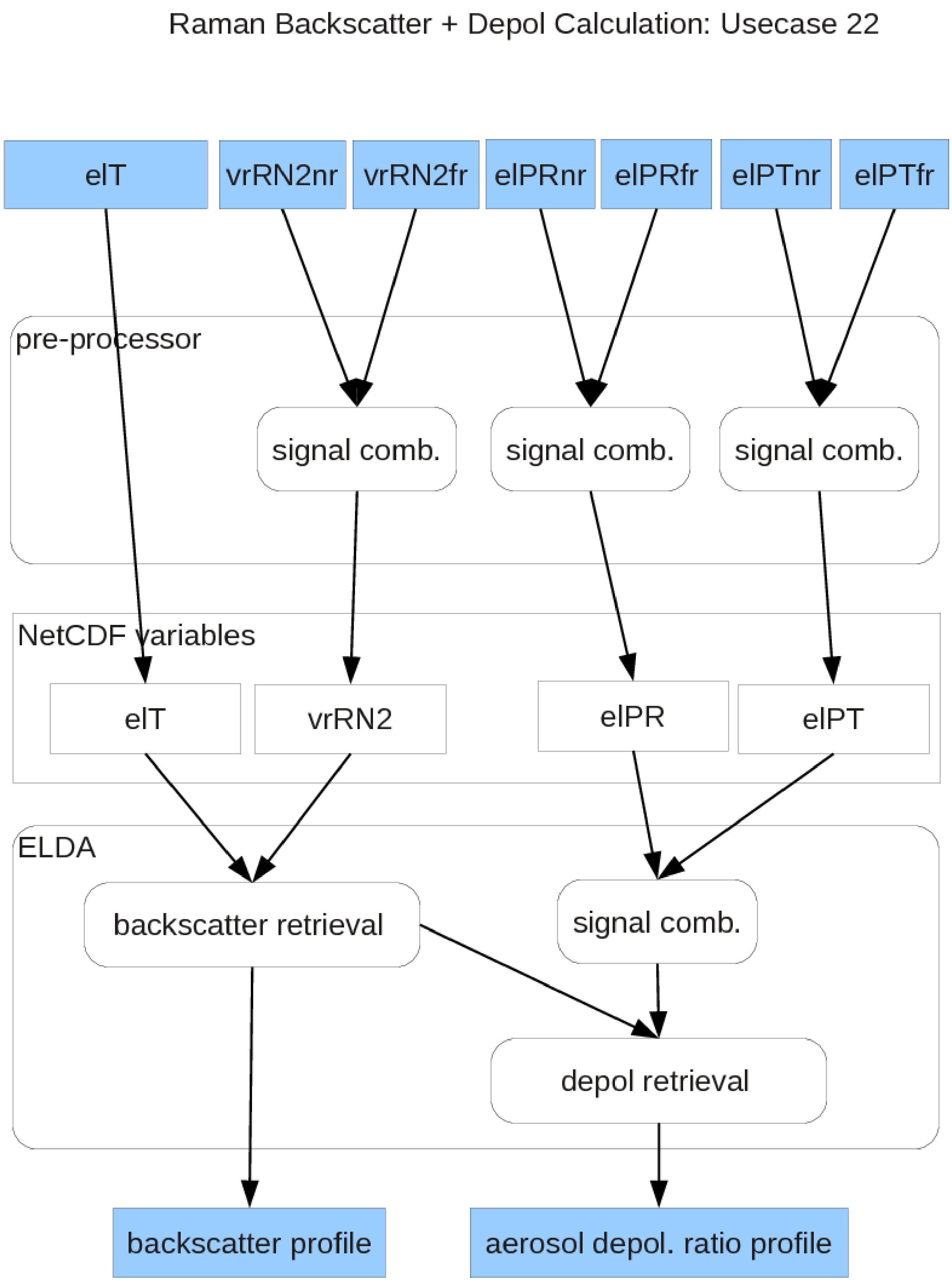

Raman Backscatter + Depol Calculation: Usecase 17
elT
vrRN2nr
vrRN2fr
elPR
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

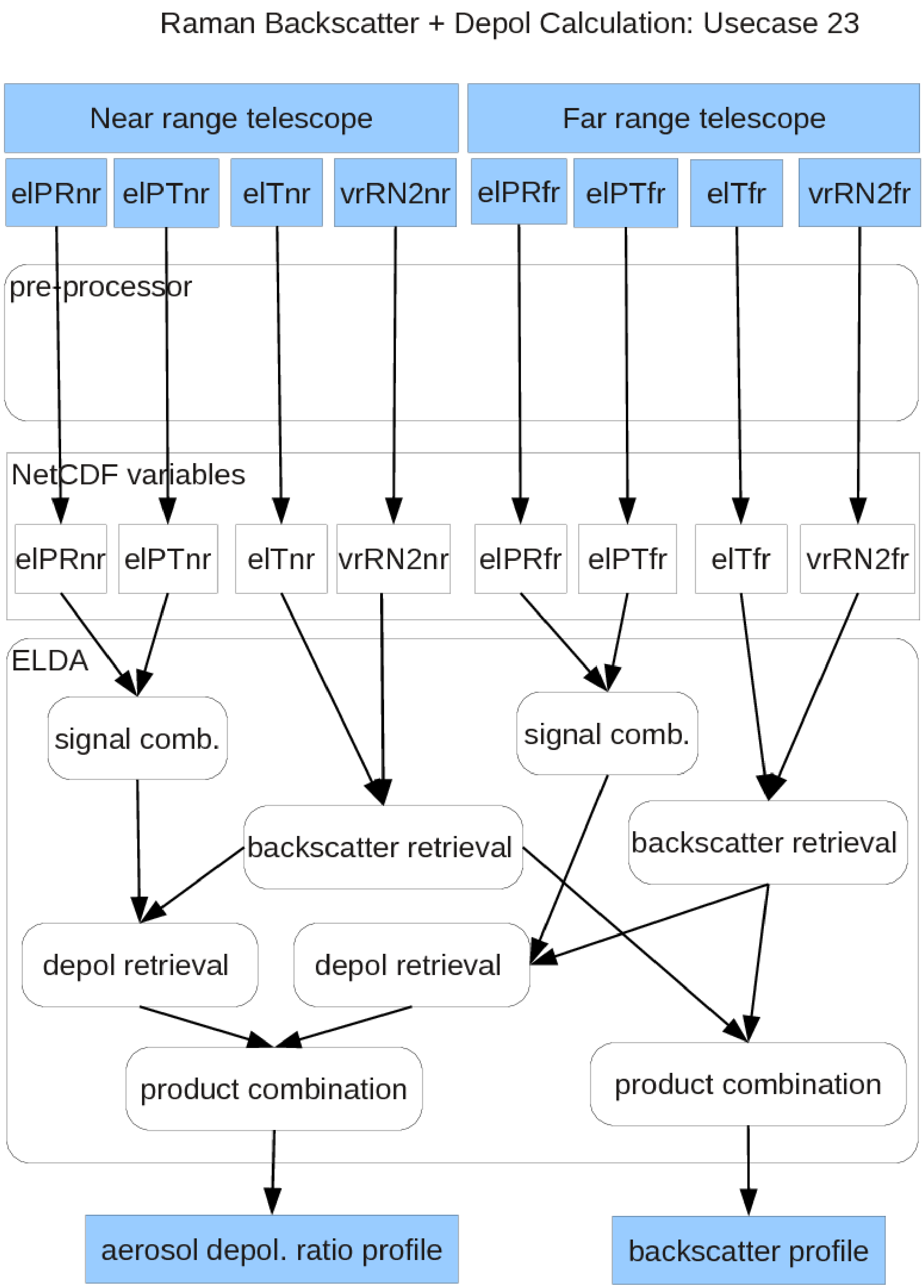

Raman Backscatter + Depol Calculation: Usecase 18
elTnr
elTfr
vrRN2
elPRnr
elPRfr
elPT
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

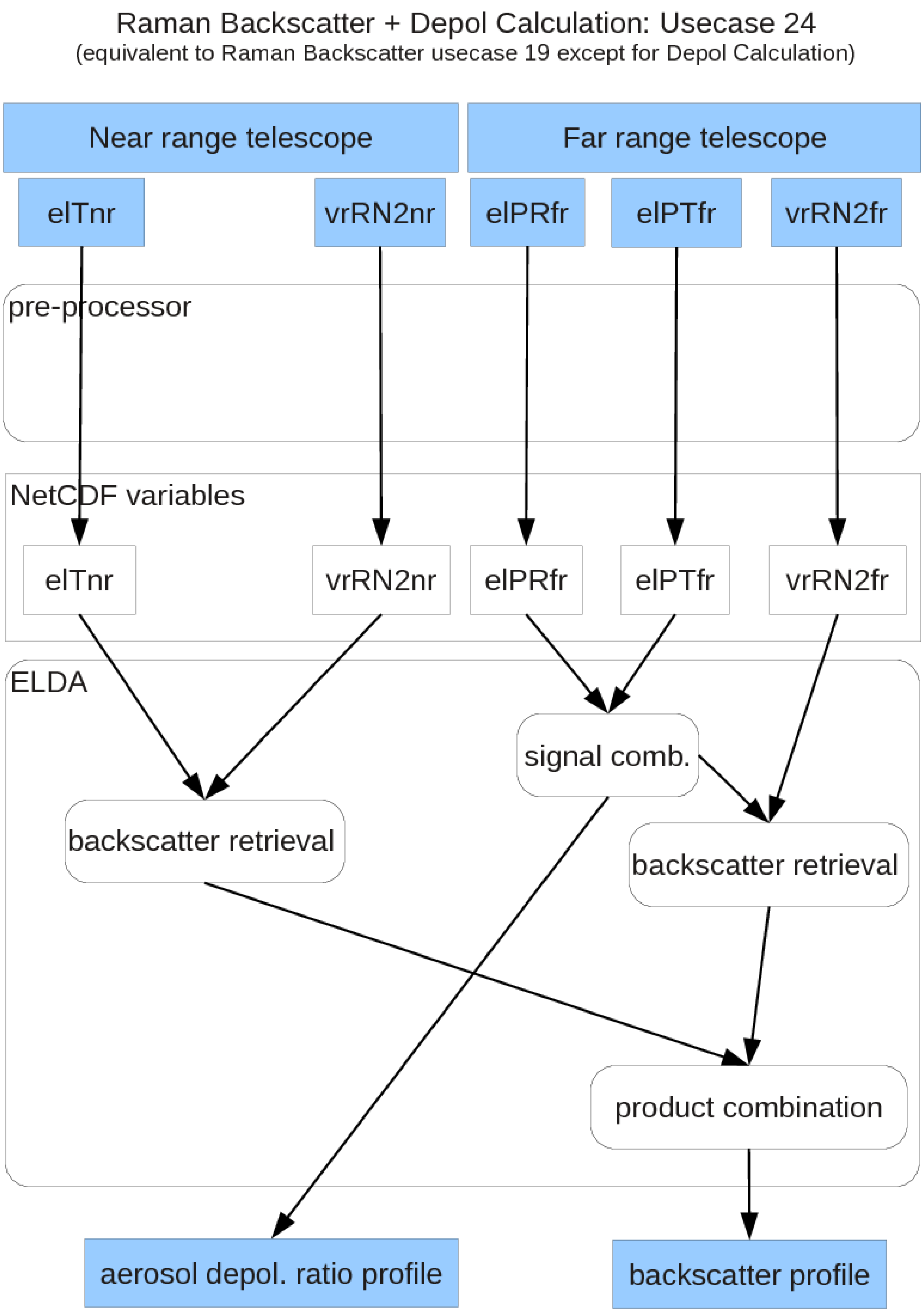

Raman Backscatter + Depol Calculation: Usecase 19
elTnr
elTfr
vrRN2
elPR
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile


Raman Backscatter + Depol Calculation: Usecase 20
elTnr
elTfr
vrRN2nr
vrRN2fr
elPRnr
elPRfr
elPT
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

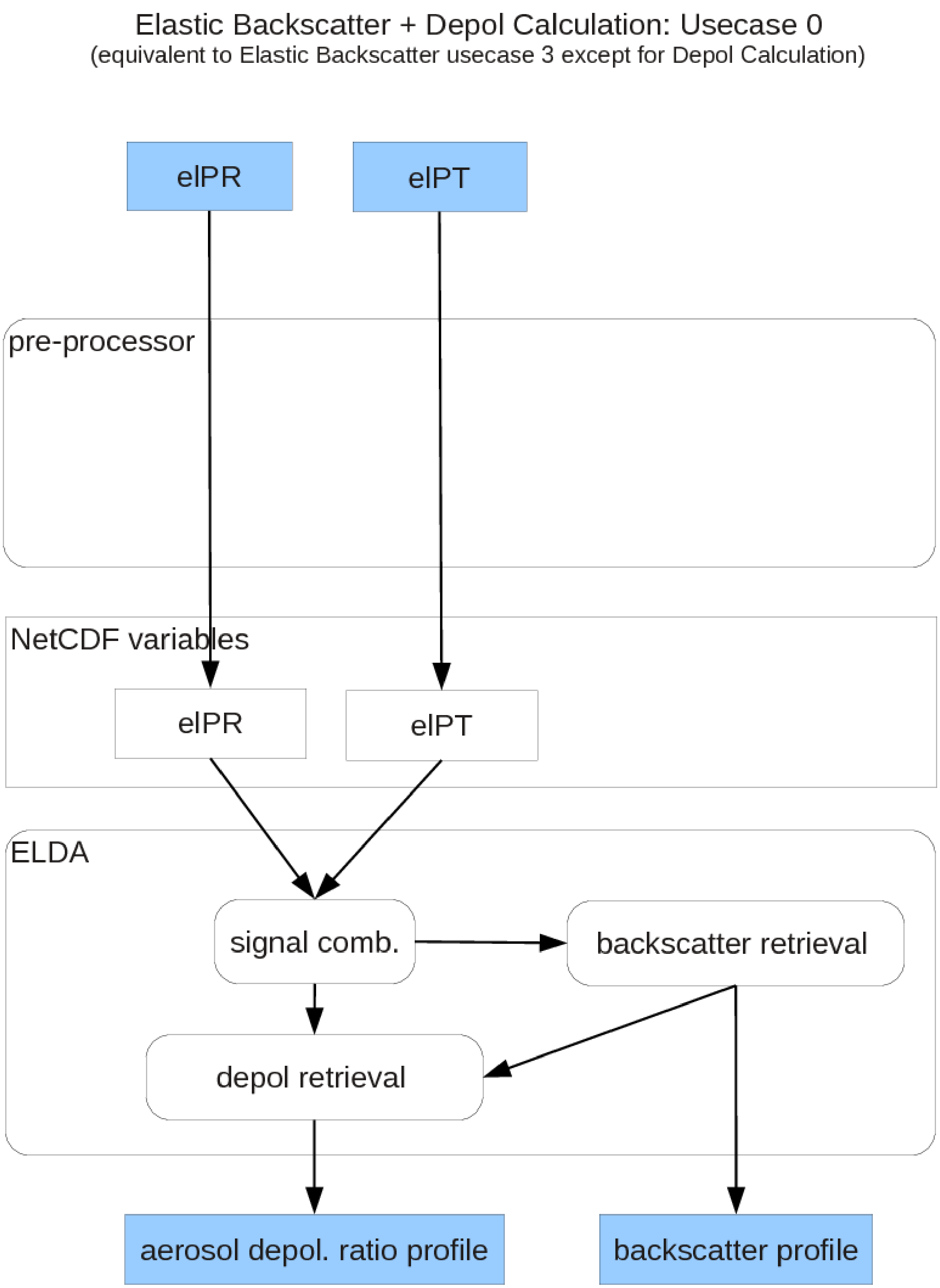

Raman Backscatter + Depol Calculation: Usecase 21
elTnr
elTfr
vrRN2nr
vrRN2fr
elPR
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

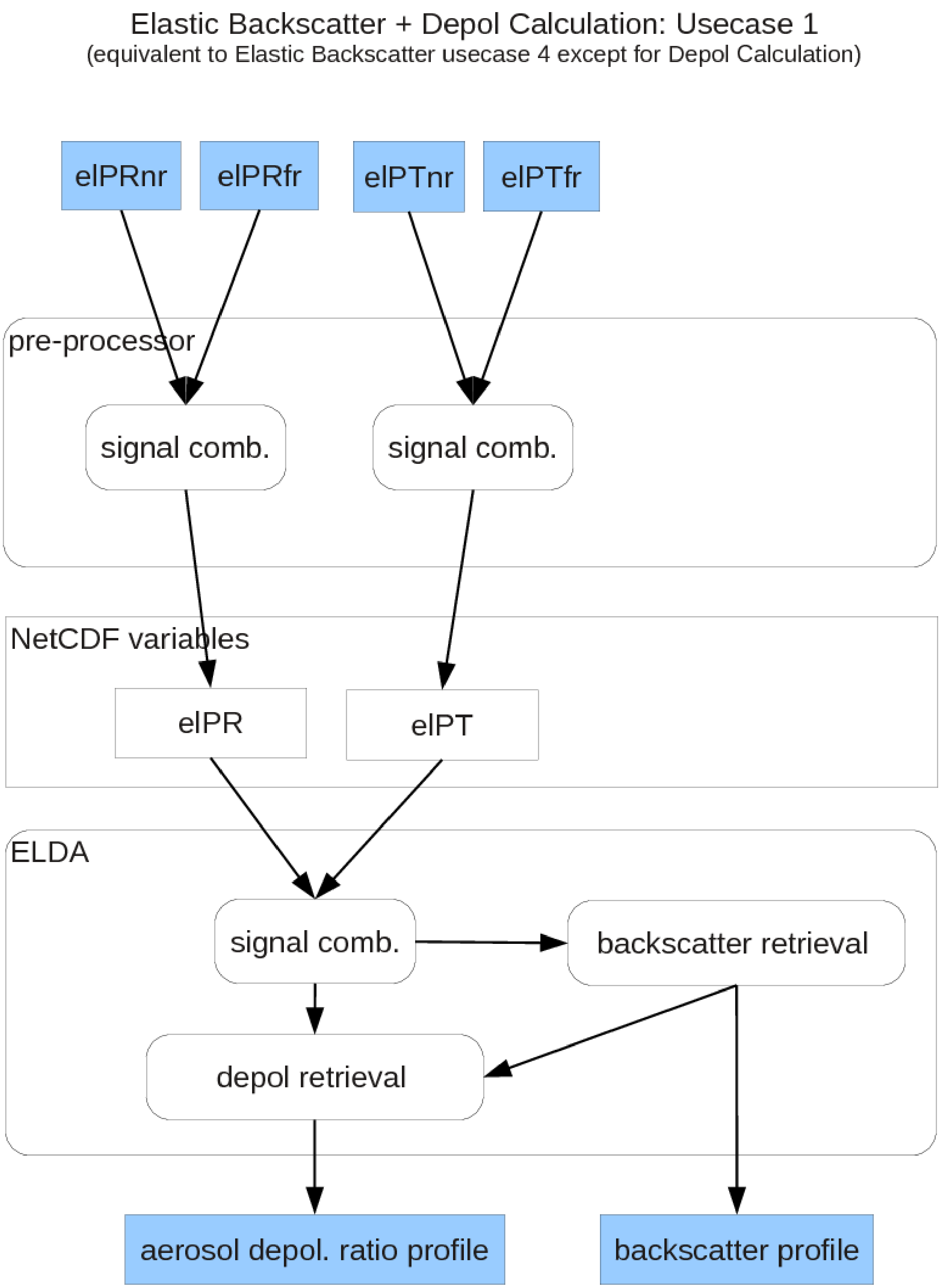

Raman Backscatter + Depol Calculation: Usecase 22
elT
vrRN2nr
vrRN2fr
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
vrRN2
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

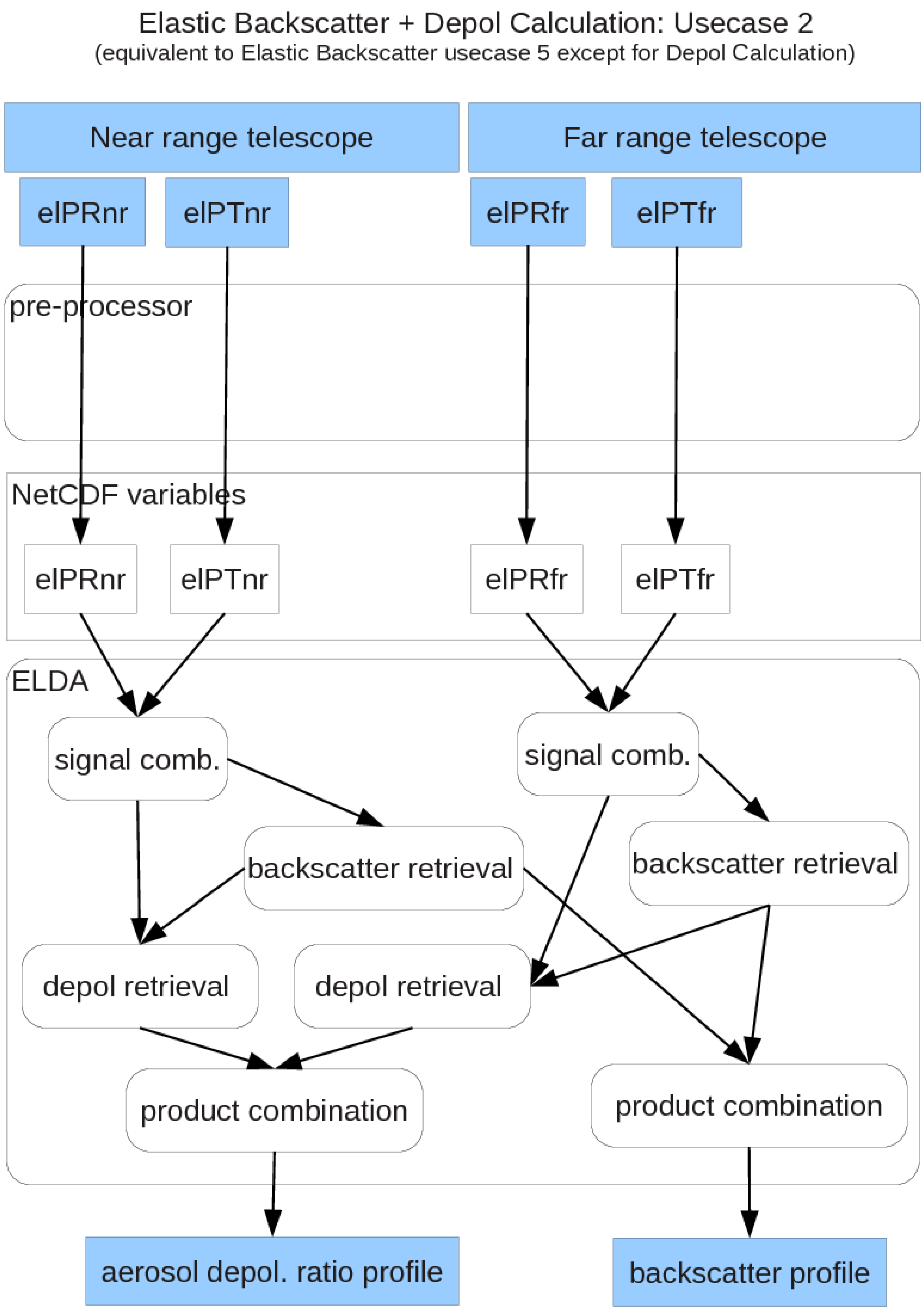

Raman Backscatter + Depol Calculation: Usecase 23
Near range telescope
Far range telescope
elPRnr
elPTnr
elTnr
vrRN2nr
elPRfr
elPTfr
elTfr
vrRN2fr
pre-processor
NetCDF variables
elPRnr
elPTnr
elTnr
vrRN2nr
elPRfr
elPTfr
elTfr
vrRN2fr
ELDA
signal comb.
signal comb.
backscatter retrieval
backscatter retrieval
depol retrieval
depol retrieval
product combination
product combination
aerosol depol. ratio profile
backscatter profile


Raman Backscatter + Depol Calculation: Usecase 24
(equivalent to Raman Backscatter usecase 19 except for Depol Calculation)
Near range telescope
Far range telescope
elTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
pre-processor
NetCDF variables
elTnr
vrRN2nr
elPRfr
elPTfr
vrRN2fr
ELDA
signal comb.
backscatter retrieval
backscatter retrieval
product combination
aerosol depol. ratio profile
backscatter profile

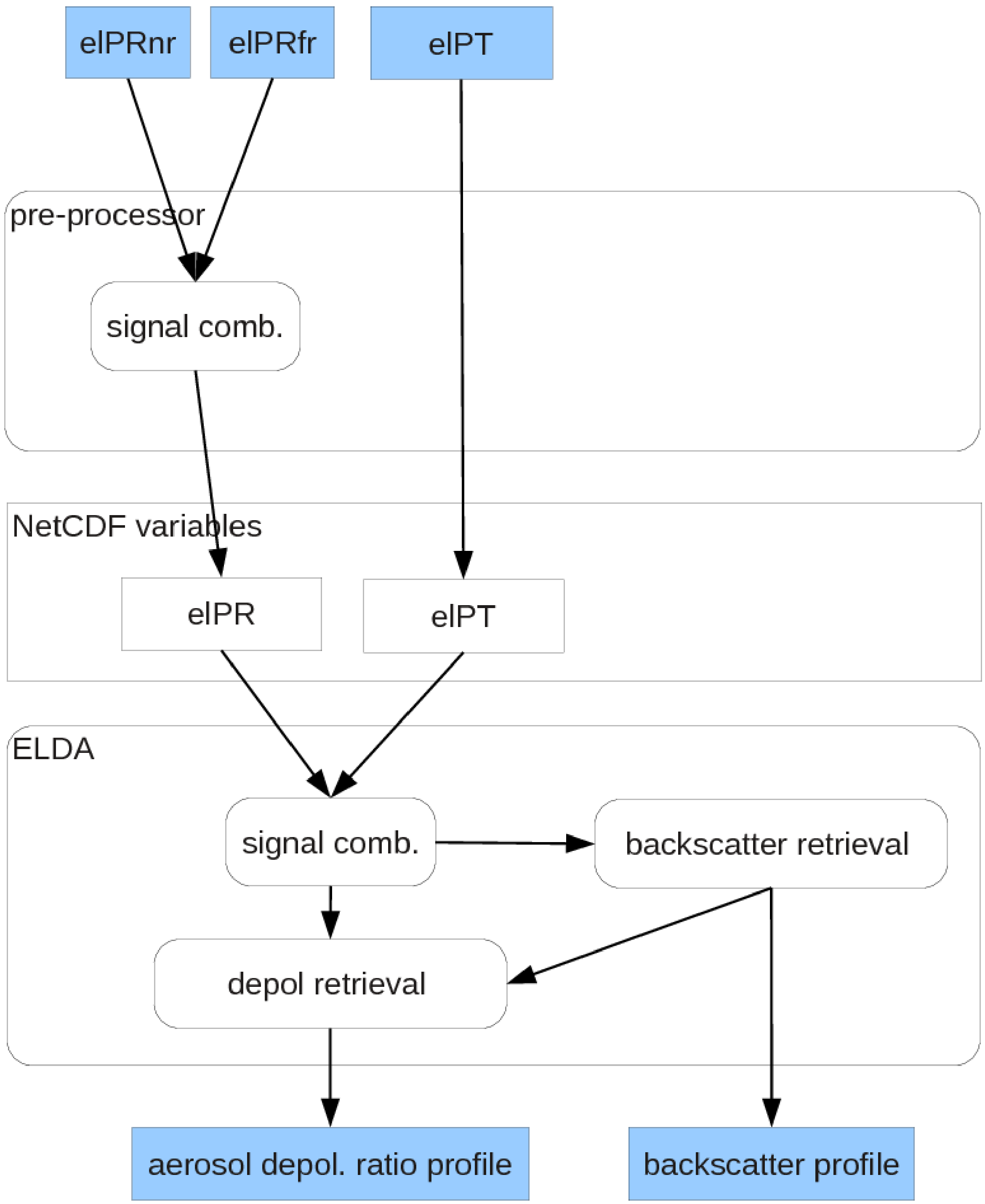
3.6.5 Elastic backscatter and depolarization

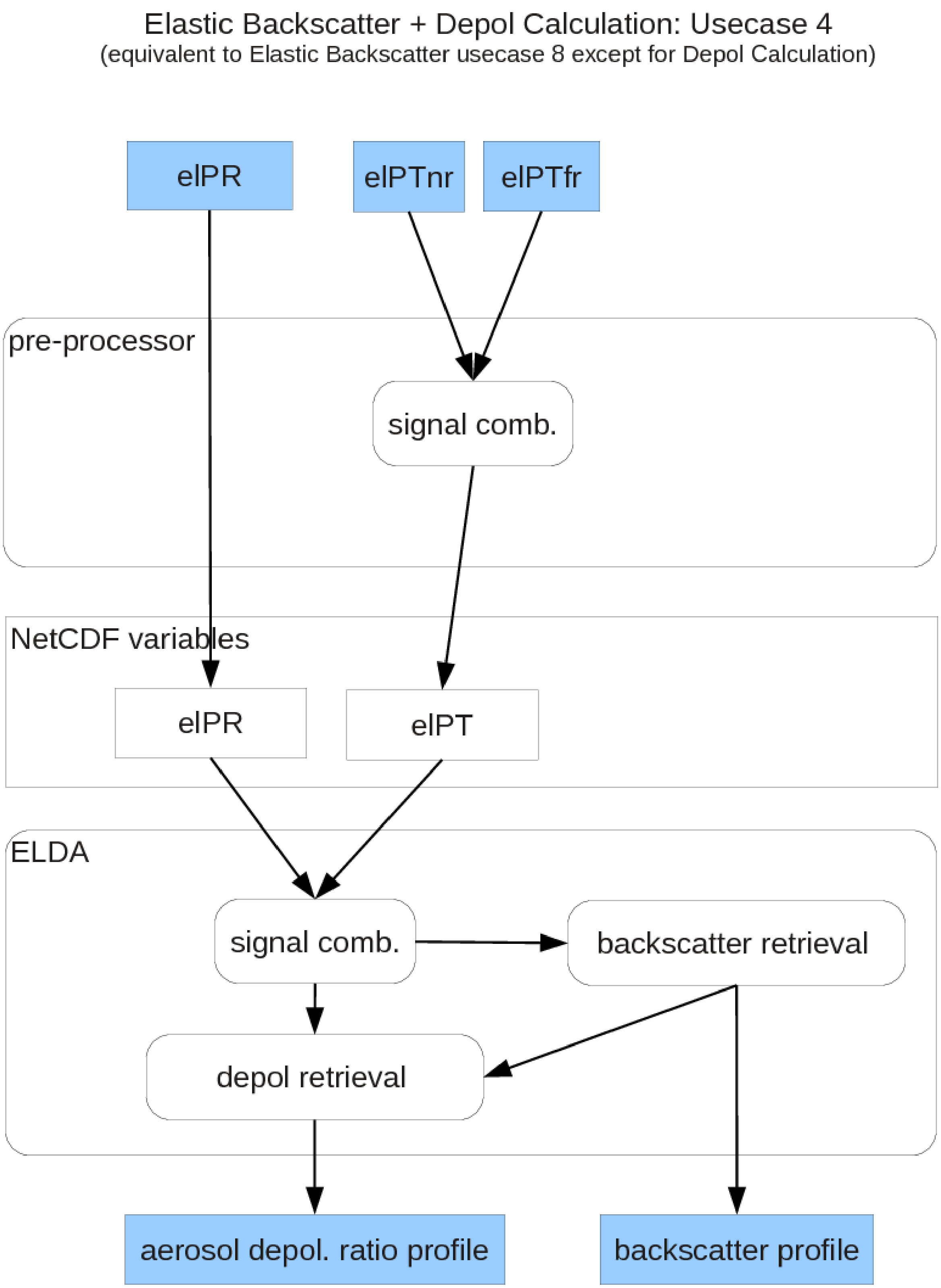


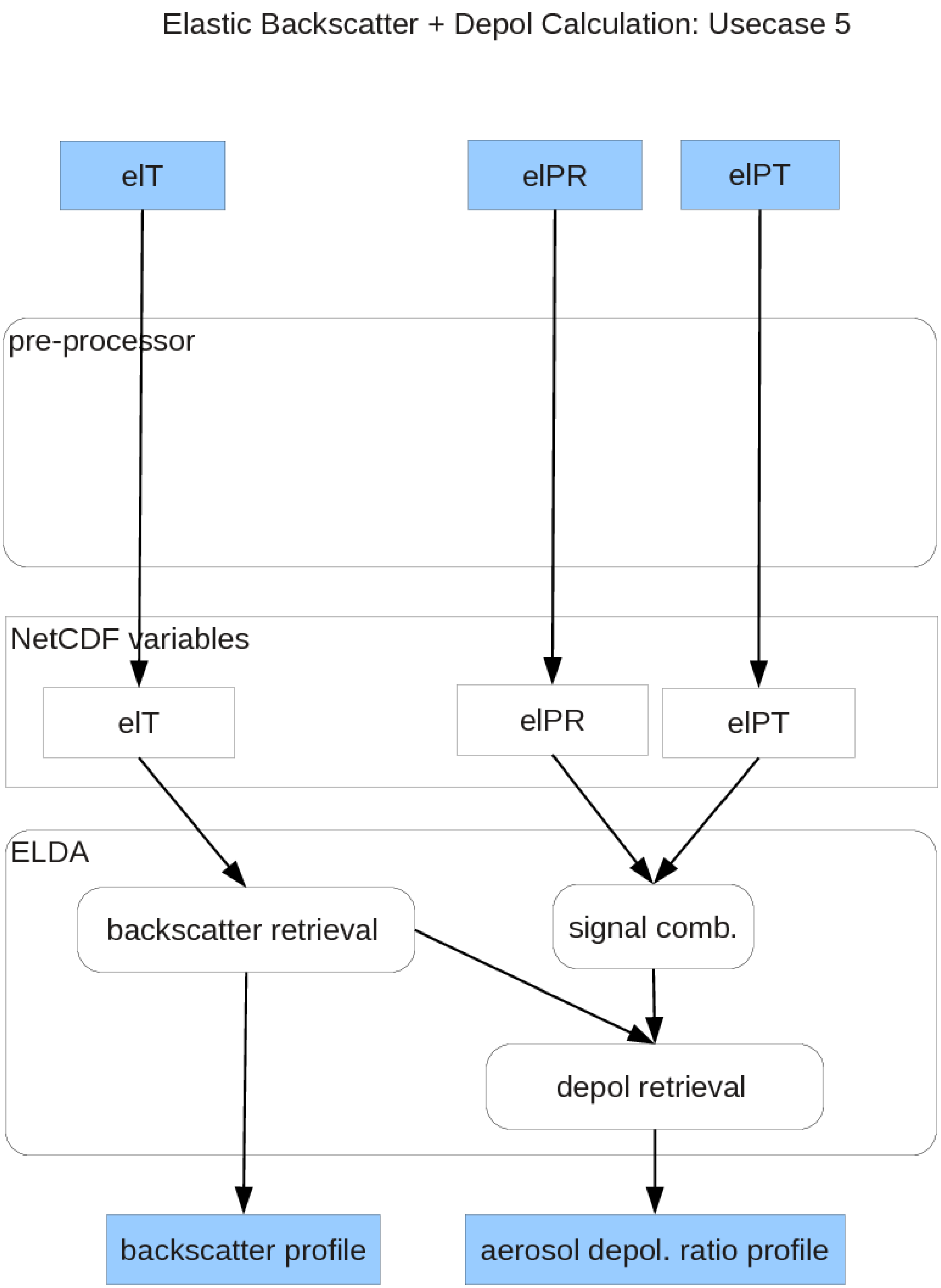

Elastic Backscatter + Depol Calculation: Usecase 1
(equivalent to Elastic Backscatter usecase 4 except for Depol Calculation)
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elPR
elPT
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

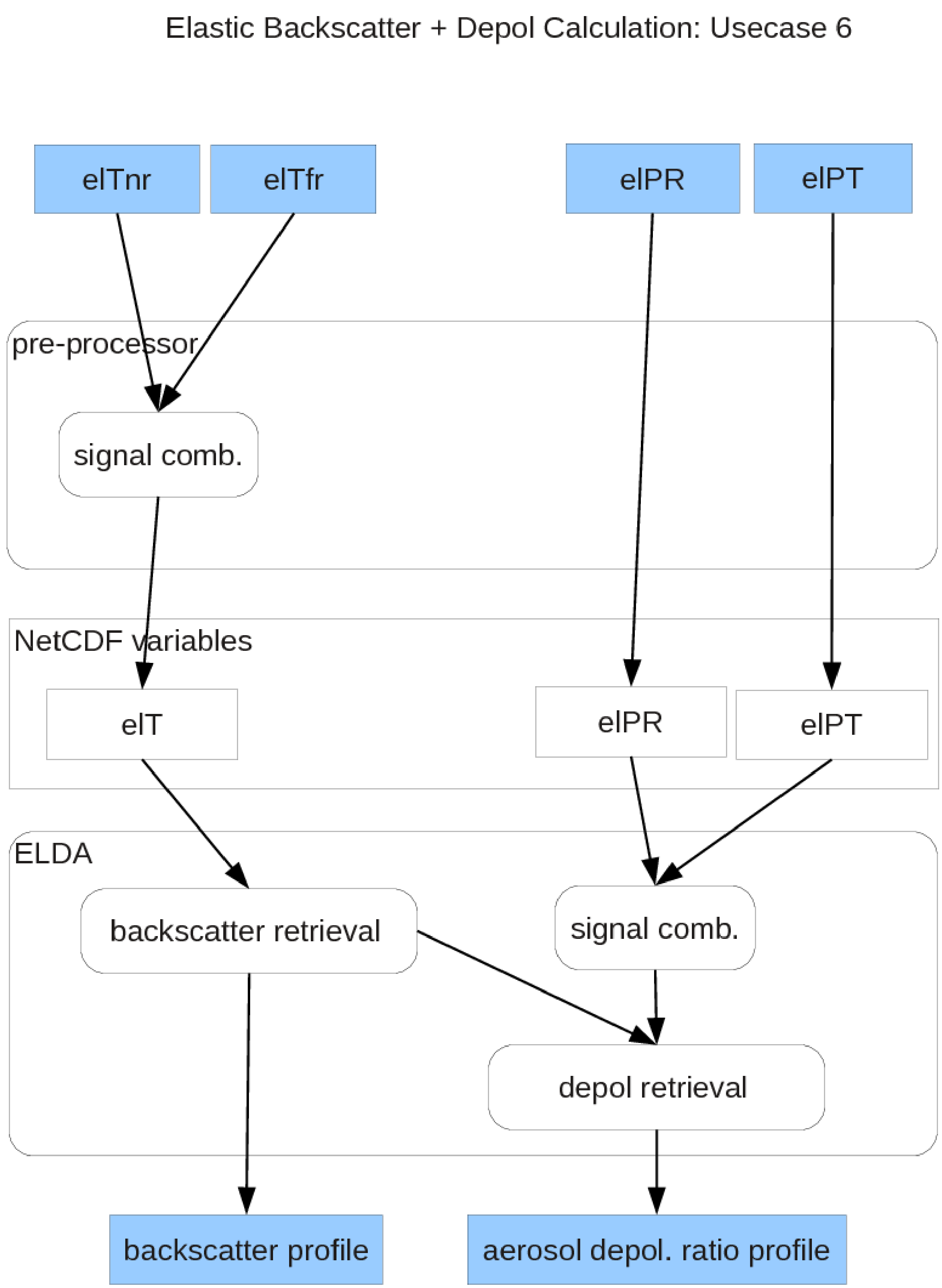

Elastic Backscatter + Depol Calculation: Usecase 2
(equivalent to Elastic Backscatter usecase 5 except for Depol Calculation)
Near range telescope
Far range telescope
elPRnr
elPTnr
elPRfr
elPTfr
pre-processor
NetCDF variables
elPRnr
elPTnr
elPRfr
elPTfr
ELDA
signal comb.
signal comb.
backscatter retrieval
backscatter retrieval
depol retrieval
depol retrieval
product combination
product combination
aerosol depol. ratio profile
backscatter profile

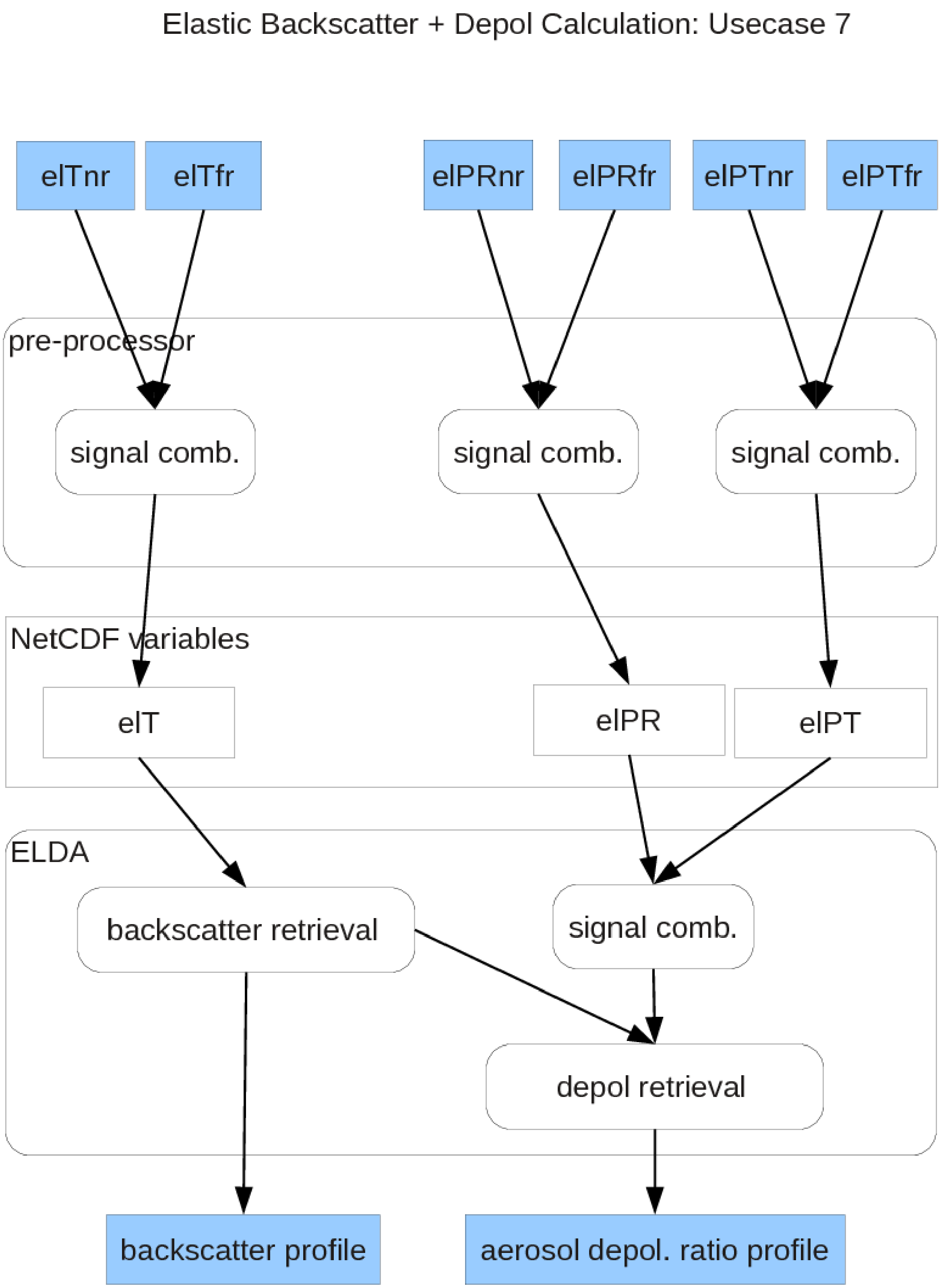
Elastic Backscatter + Depol Calculation: Usecase 3
(equivalent to Elastic Backscatter usecase 7 except for Depol Calculation)

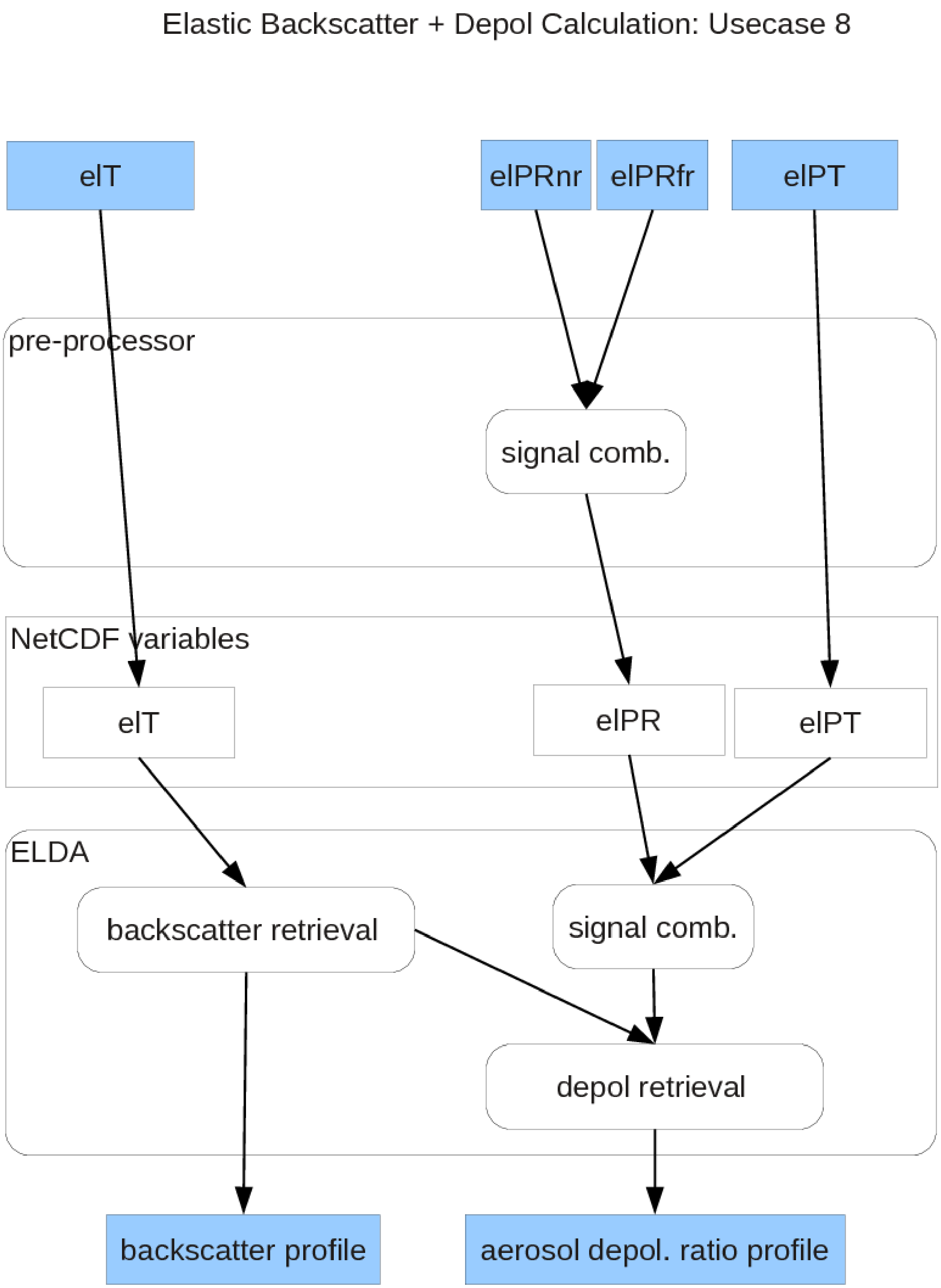


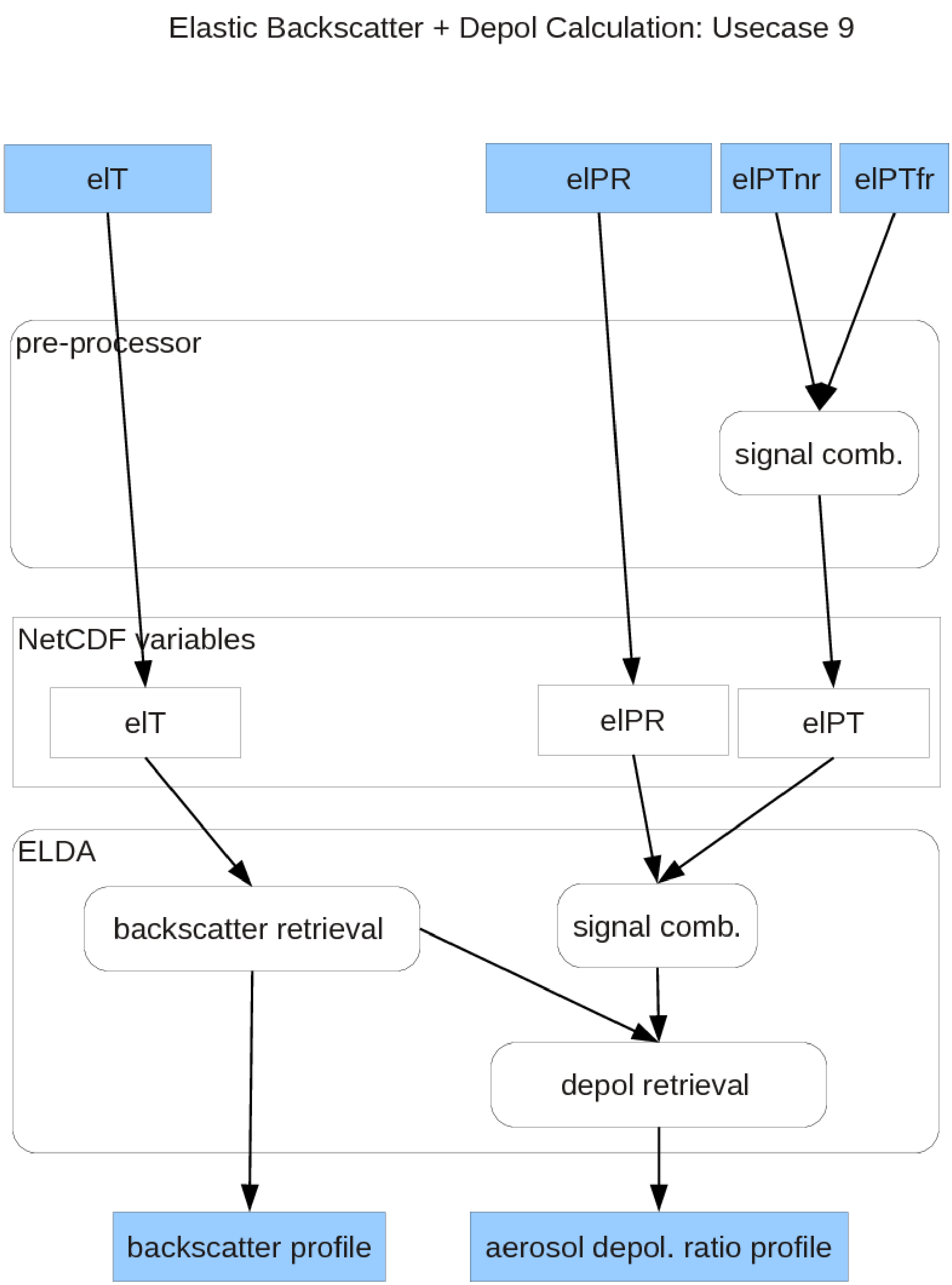

Elastic Backscatter + Depol Calculation: Usecase 4
(equivalent to Elastic Backscatter usecase 8 except for Depol Calculation)
elPR
elPTnr
elPTfr
pre-processor
signal comb.
NetCDF variables
elPR
elPT
ELDA
signal comb.
backscatter retrieval
depol retrieval
aerosol depol. ratio profile
backscatter profile

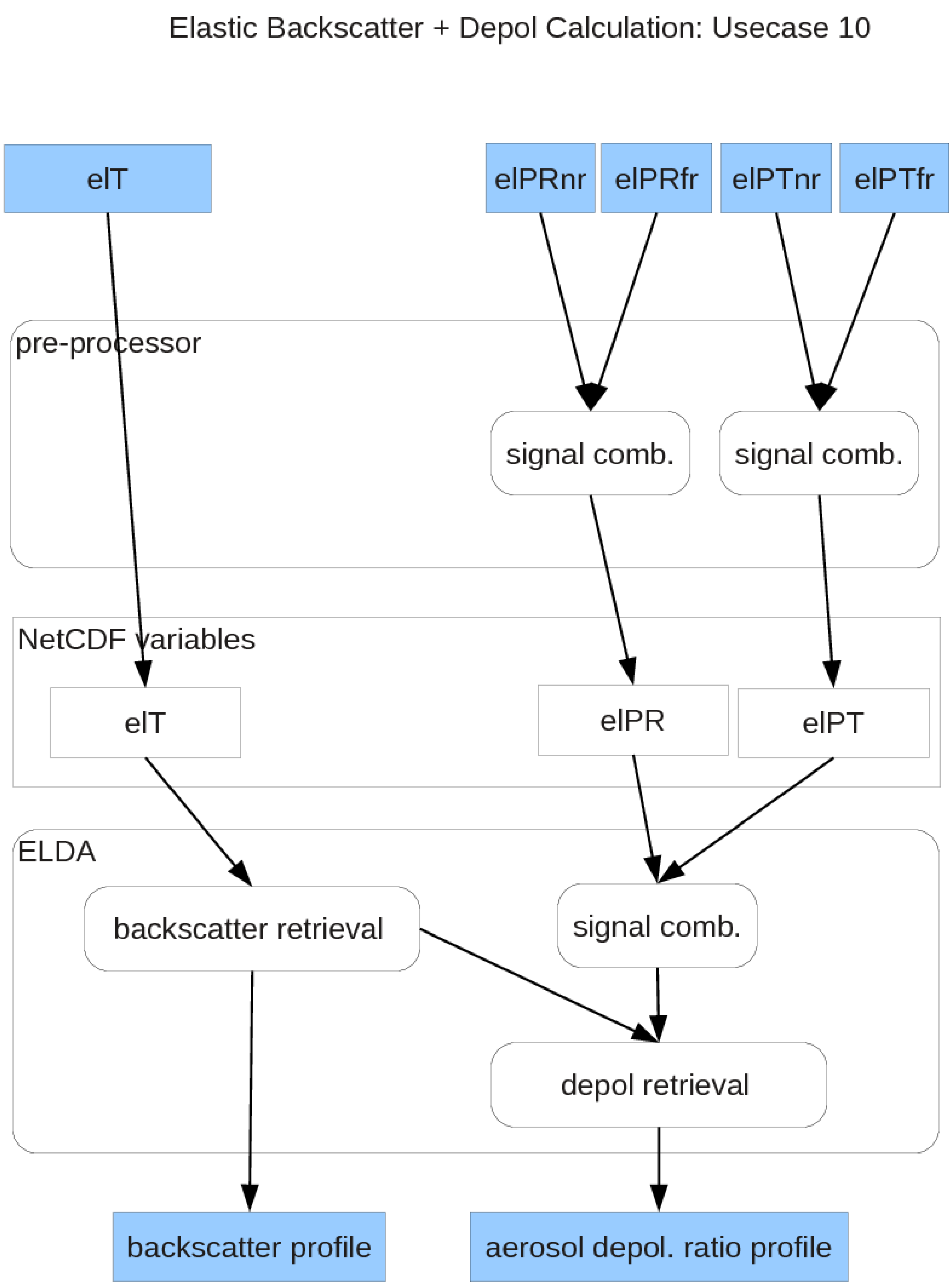

Elastic Backscatter + Depol Calculation: Usecase 5
elT
elPR
elPT
pre-processor
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

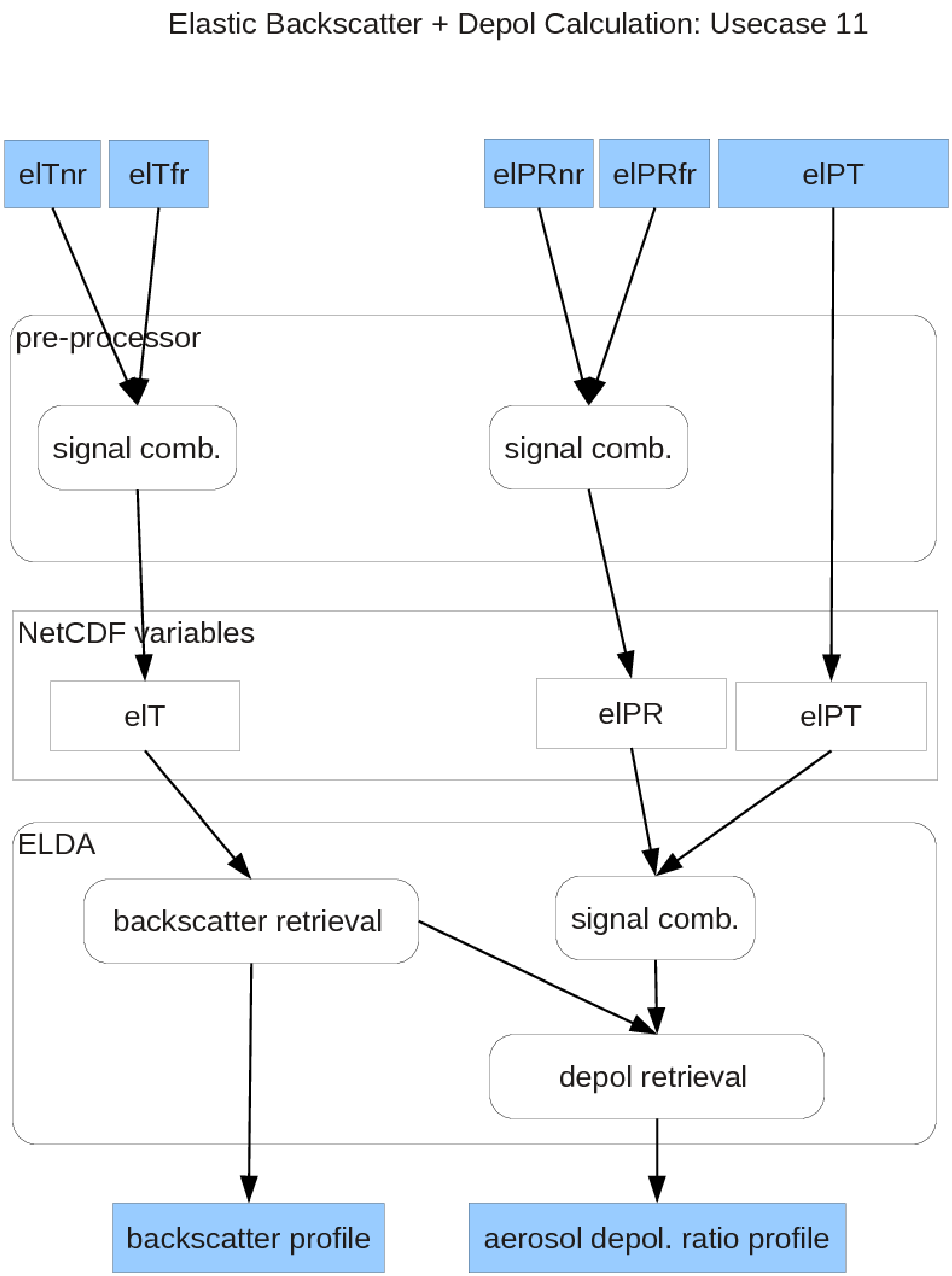

Elastic Backscatter + Depol Calculation: Usecase 6
elTnr
elTfr
elPR
elPT
pre-processor
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

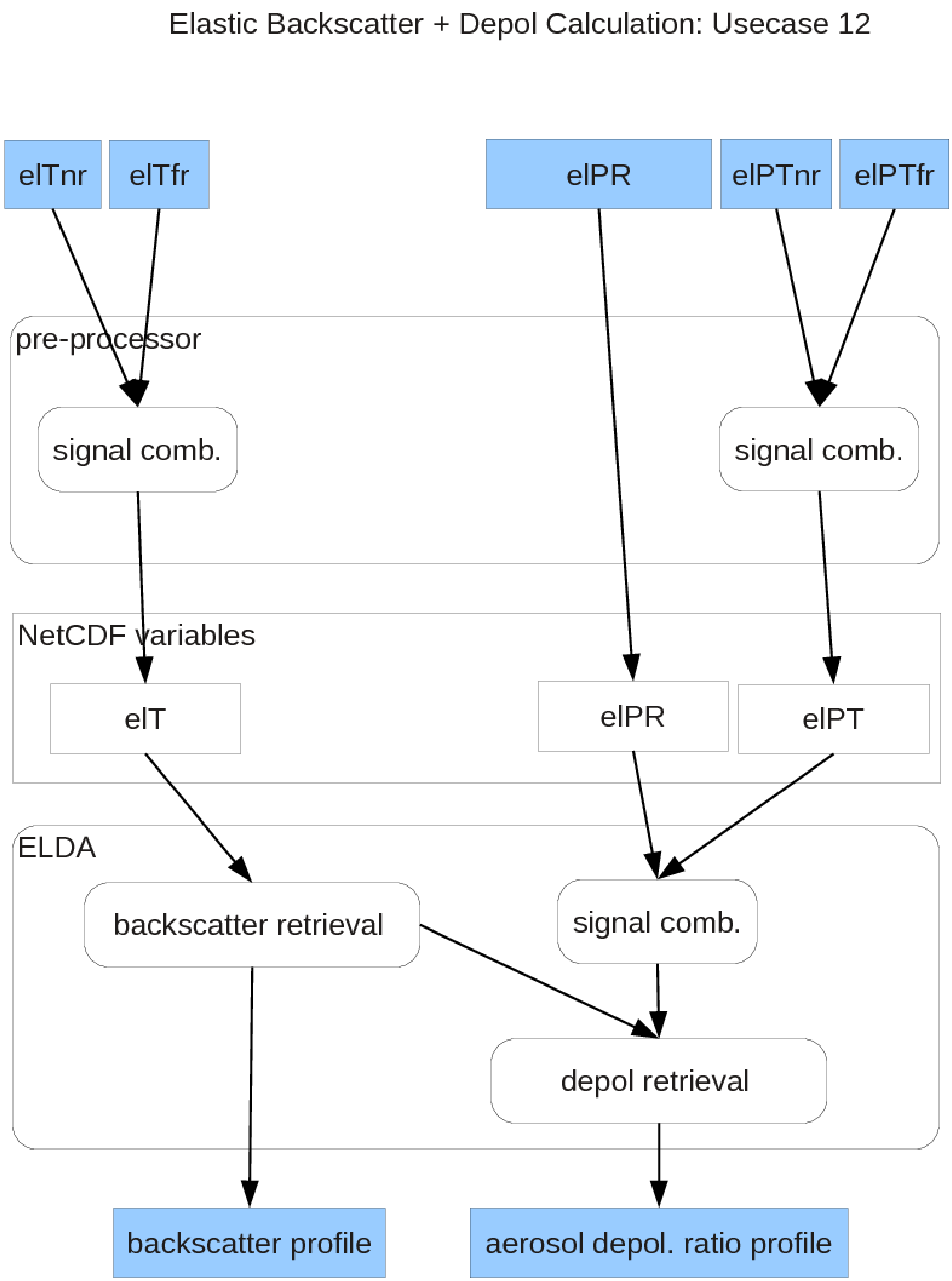

Elastic Backscatter + Depol Calculation: Usecase 7
elTnr
elTfr
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

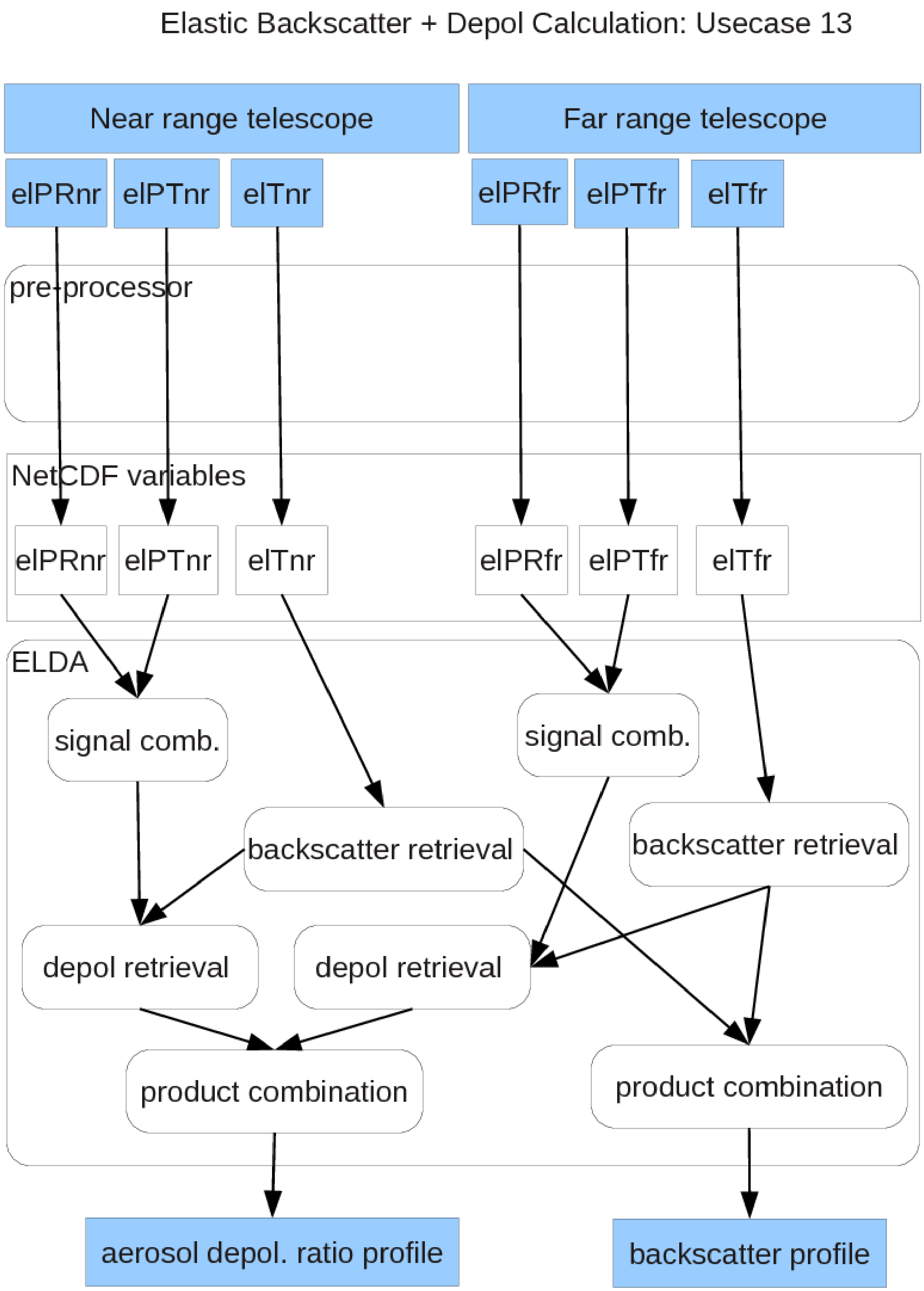

Elastic Backscatter + Depol Calculation: Usecase 8
elT
elPRnr
elPRfr
elPT
pre-processor
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

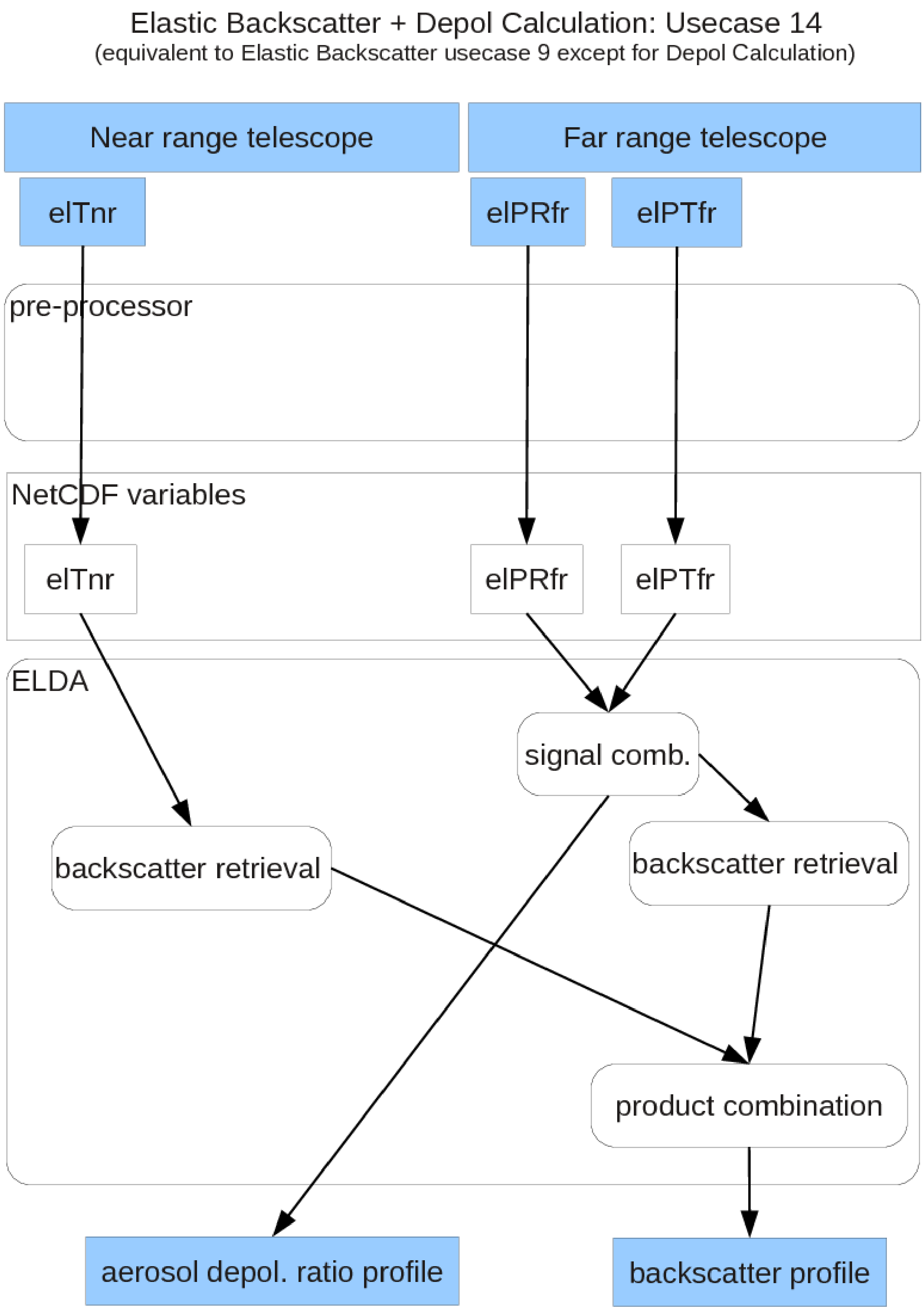

Elastic Backscatter + Depol Calculation: Usecase 9
elT
elPR
elPTnr
elPTfr
pre-processor
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile


Elastic Backscatter + Depol Calculation: Usecase 10
elT
elPRnr
elPRfr
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

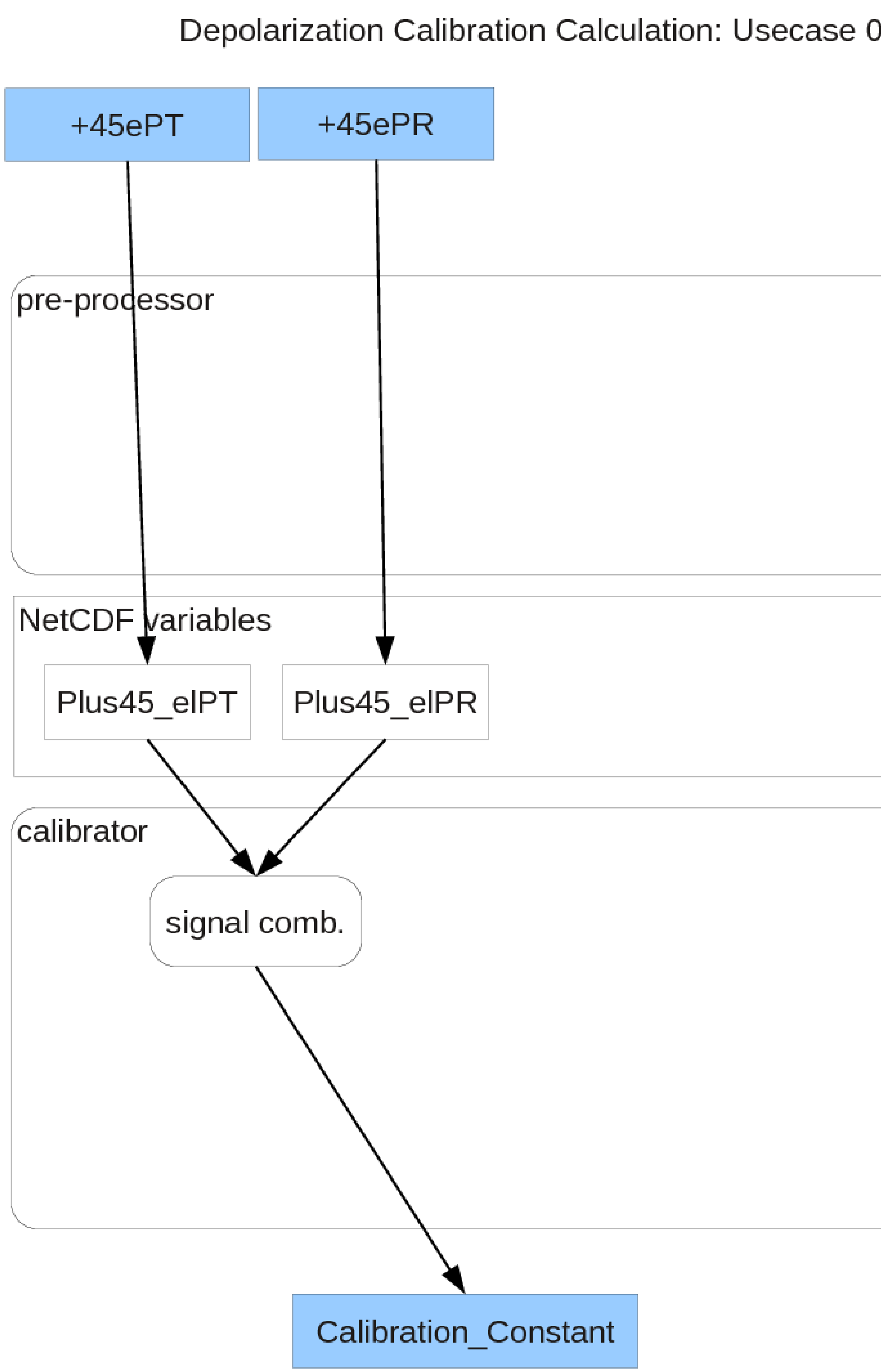

Elastic Backscatter + Depol Calculation: Usecase 11
elTnr
elTfr
elPRnr
elPRfr
elPT
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

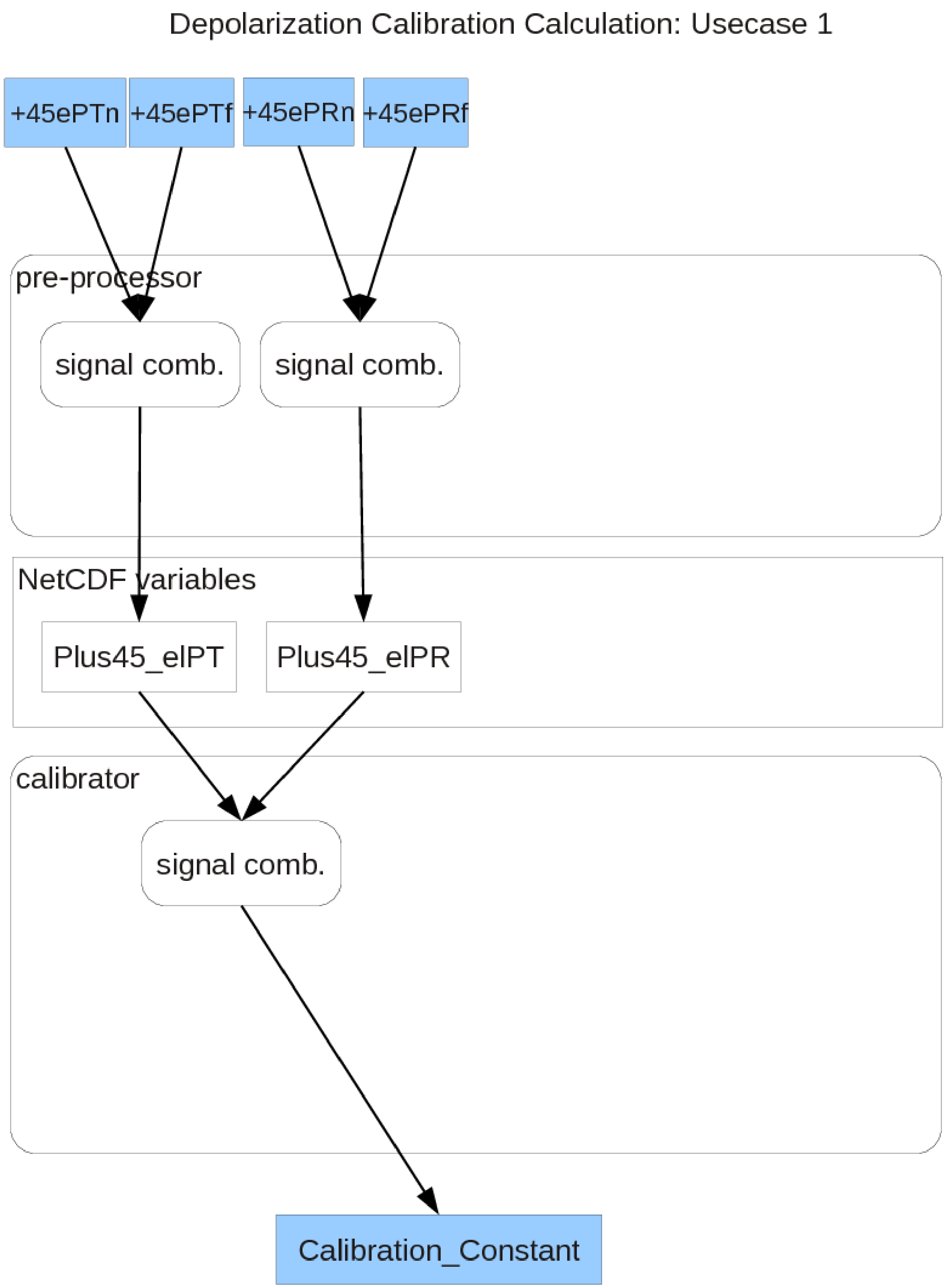

Elastic Backscatter + Depol Calculation: Usecase 12
elTnr
elTfr
elPR
elPTnr
elPTfr
pre-processor
signal comb.
signal comb.
NetCDF variables
elT
elPR
elPT
ELDA
backscatter retrieval
signal comb.
depol retrieval
backscatter profile
aerosol depol. ratio profile

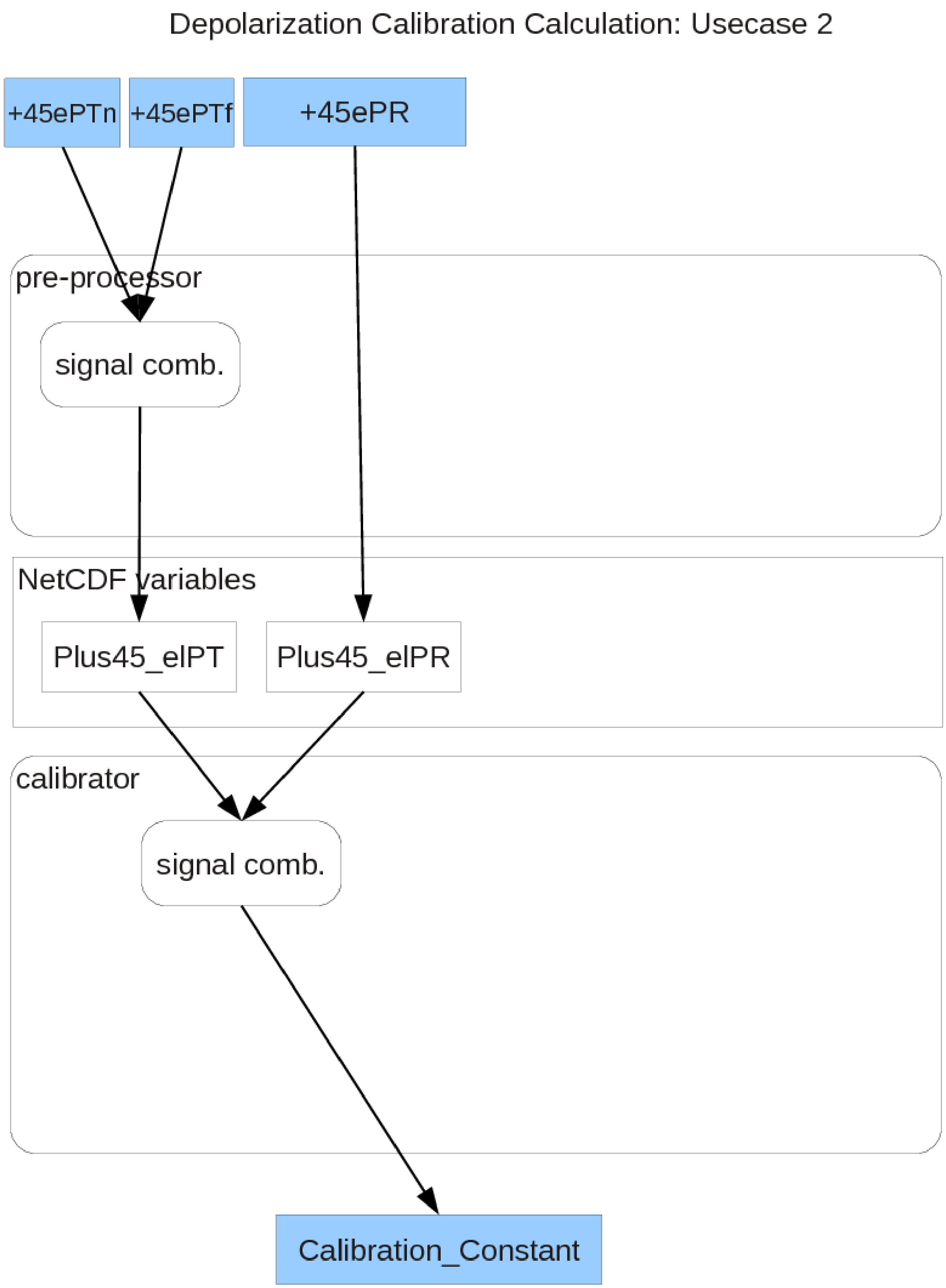

Elastic Backscatter + Depol Calculation: Usecase 13
Near range telescope
Far range telescope
elPRnr
elPTnr
elTnr
elPRfr
elPTfr
elTfr
pre-processor
NetCDF variables
elPRnr
elPTnr
elTnr
elPRfr
elPTfr
elTfr
ELDA
signal comb.
signal comb.
backscatter retrieval
backscatter retrieval
depol retrieval
depol retrieval
product combination
product combination
aerosol depol. ratio profile
backscatter profile

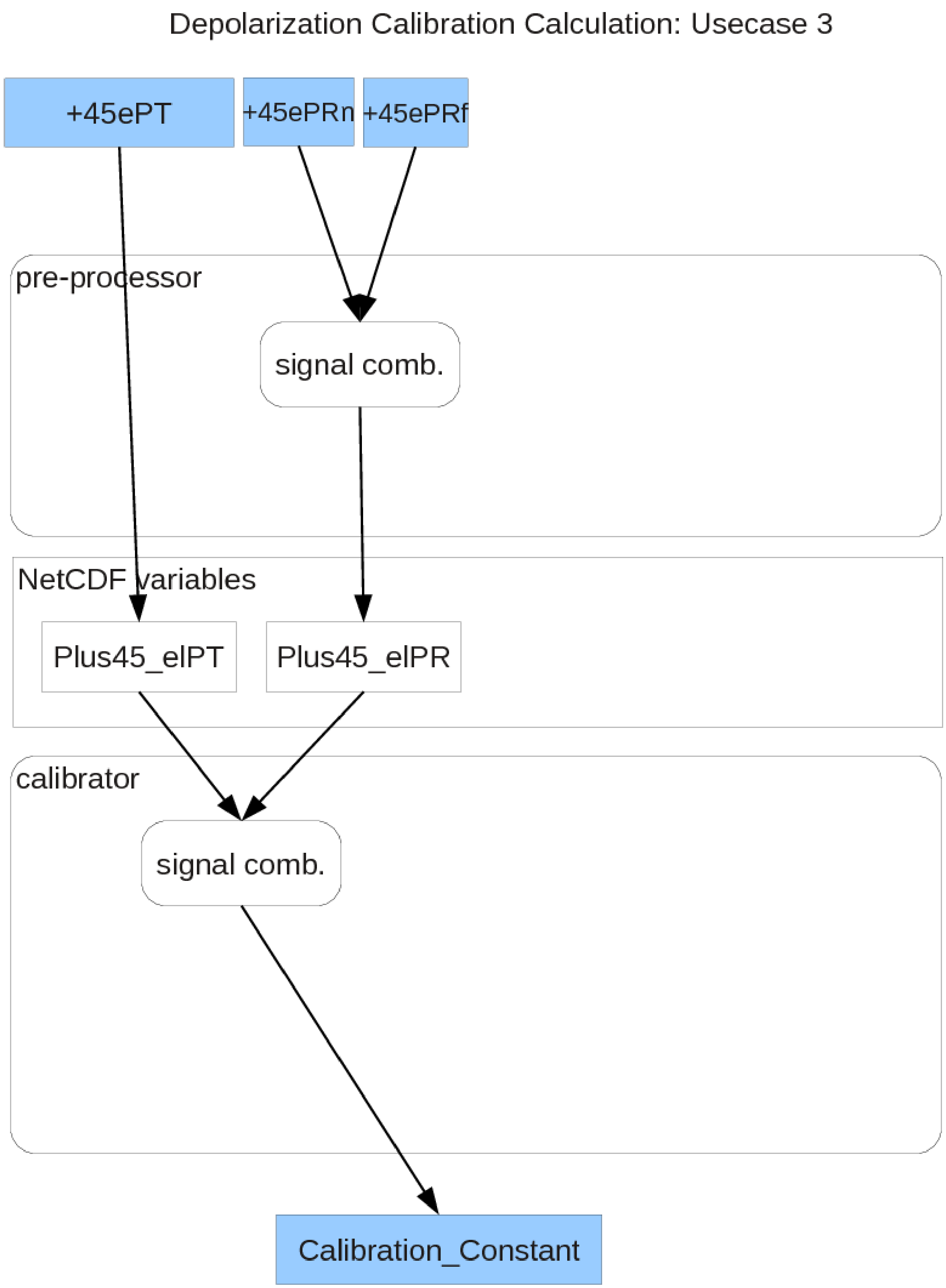

Elastic Backscatter + Depol Calculation: Usecase 14
(equivalent to Elastic Backscatter usecase 9 except for Depol Calculation)
Near range telescope
Far range telescope
elTnr
elPRfr
elPTfr
pre-processor
NetCDF variables
elTnr
elPRfr
elPTfr
ELDA
signal comb.
backscatter retrieval
backscatter retrieval
product combination
aerosol depol. ratio profile
backscatter profile

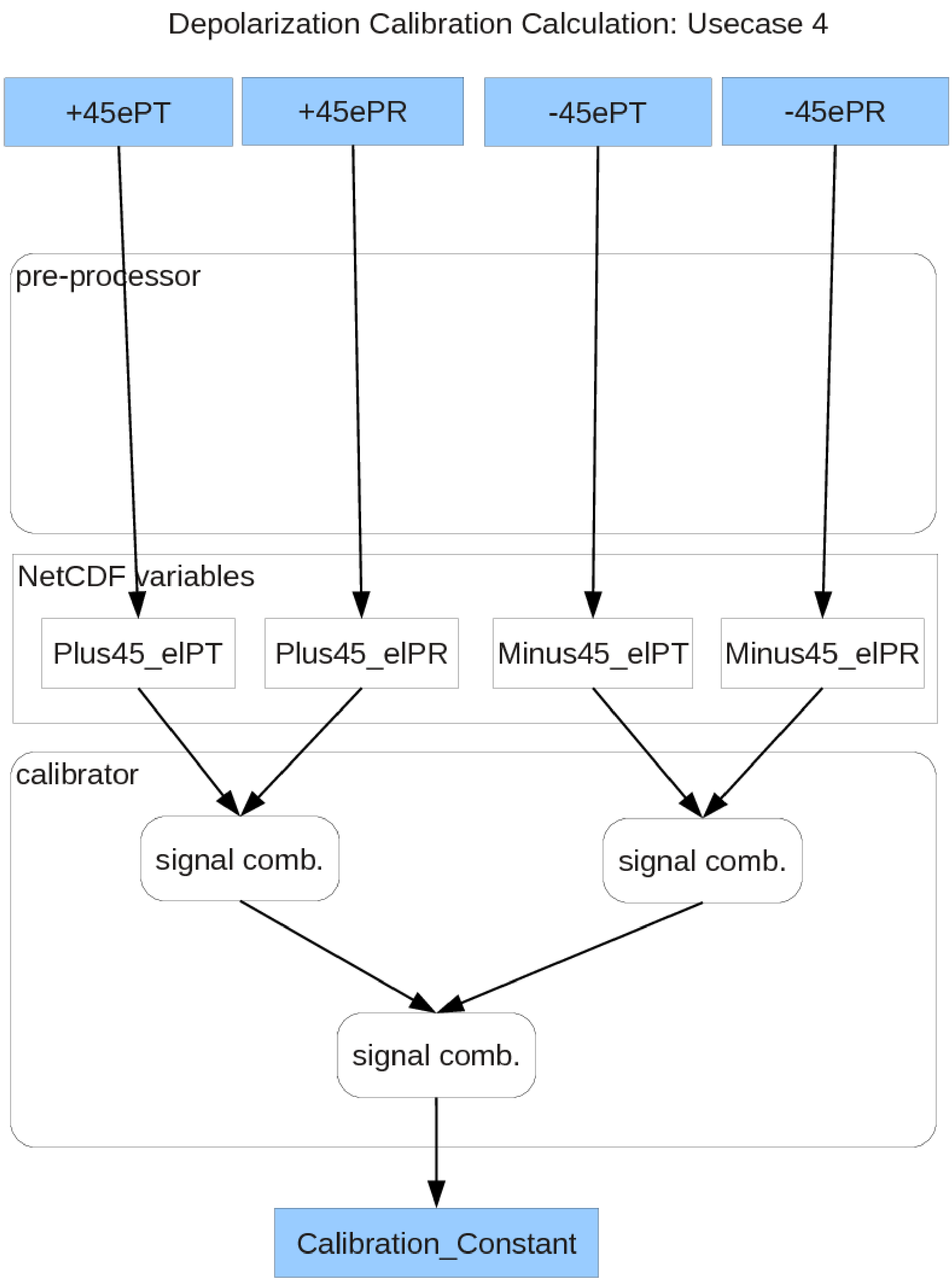
3.6.6 Depolarization calibration

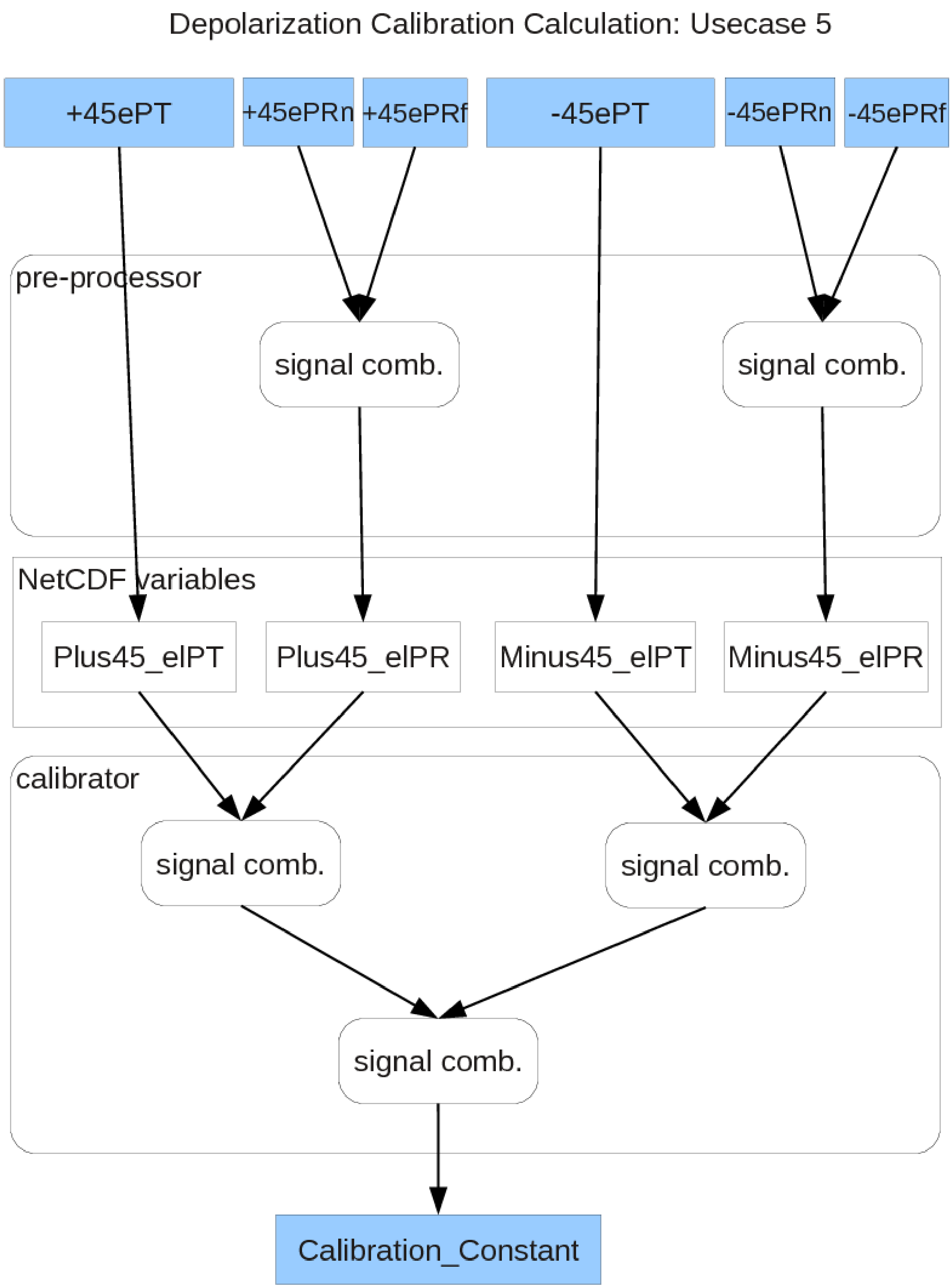


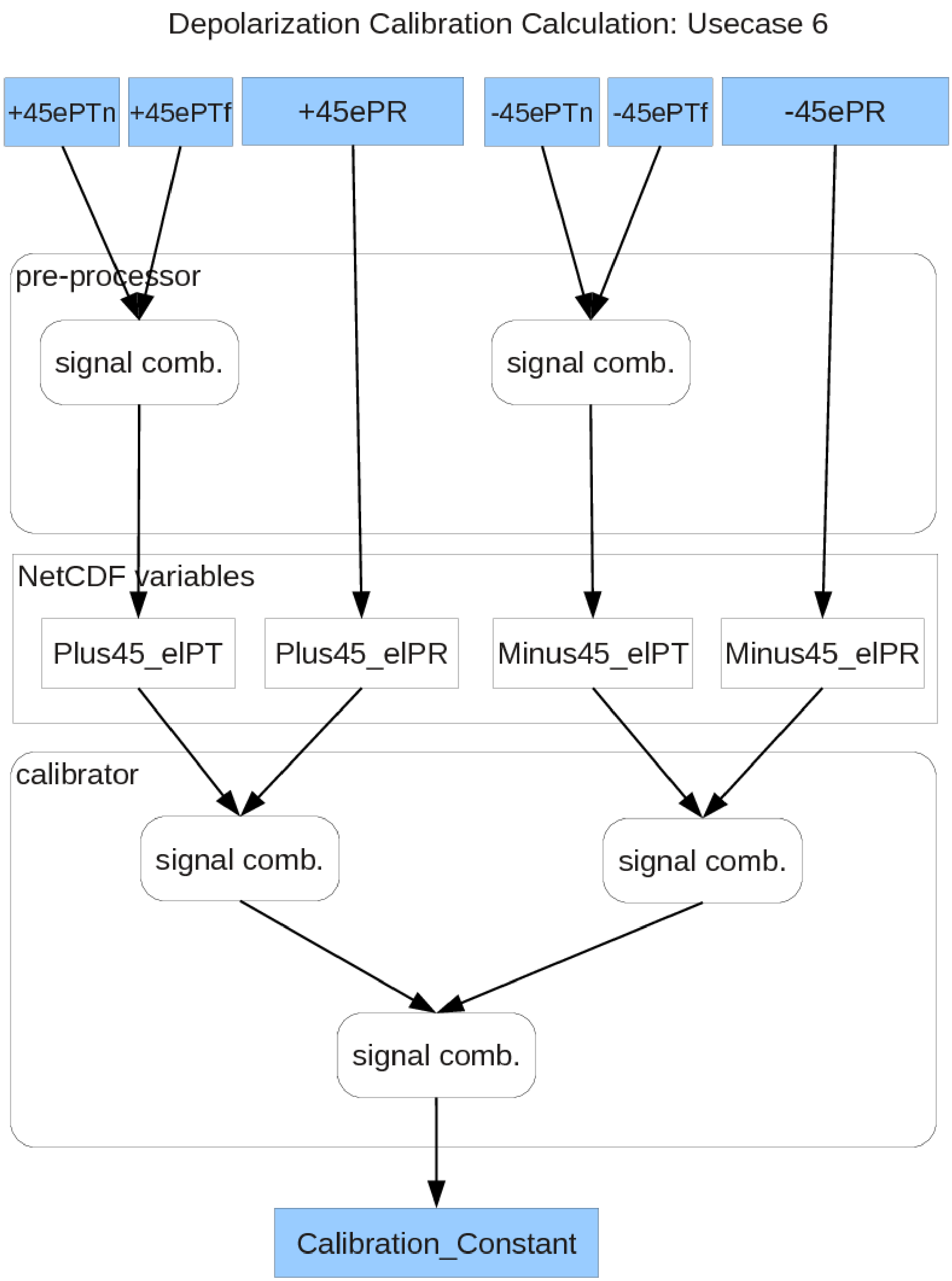

Depolarization Calibration Calculation: Usecase 1
+45ePTn
+45ePTf
+45ePRn
+45ePRf
pre-processor
signal comb.
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
calibrator
signal comb.
Calibration_Constant

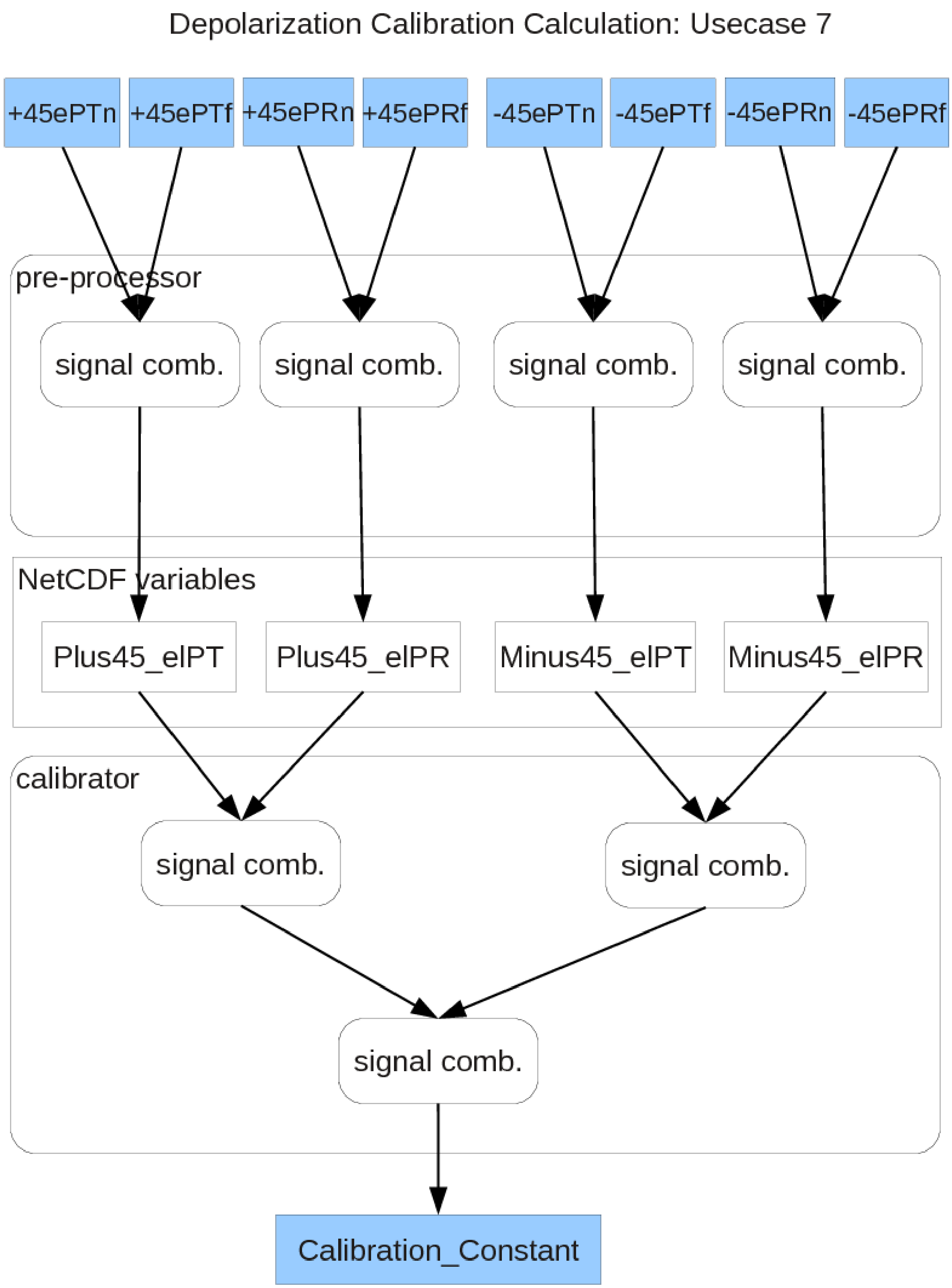

Depolarization Calibration Calculation: Usecase 2
+45ePTn
+45ePTf
+45ePR
pre-processor
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
calibrator
signal comb.
Calibration_Constant

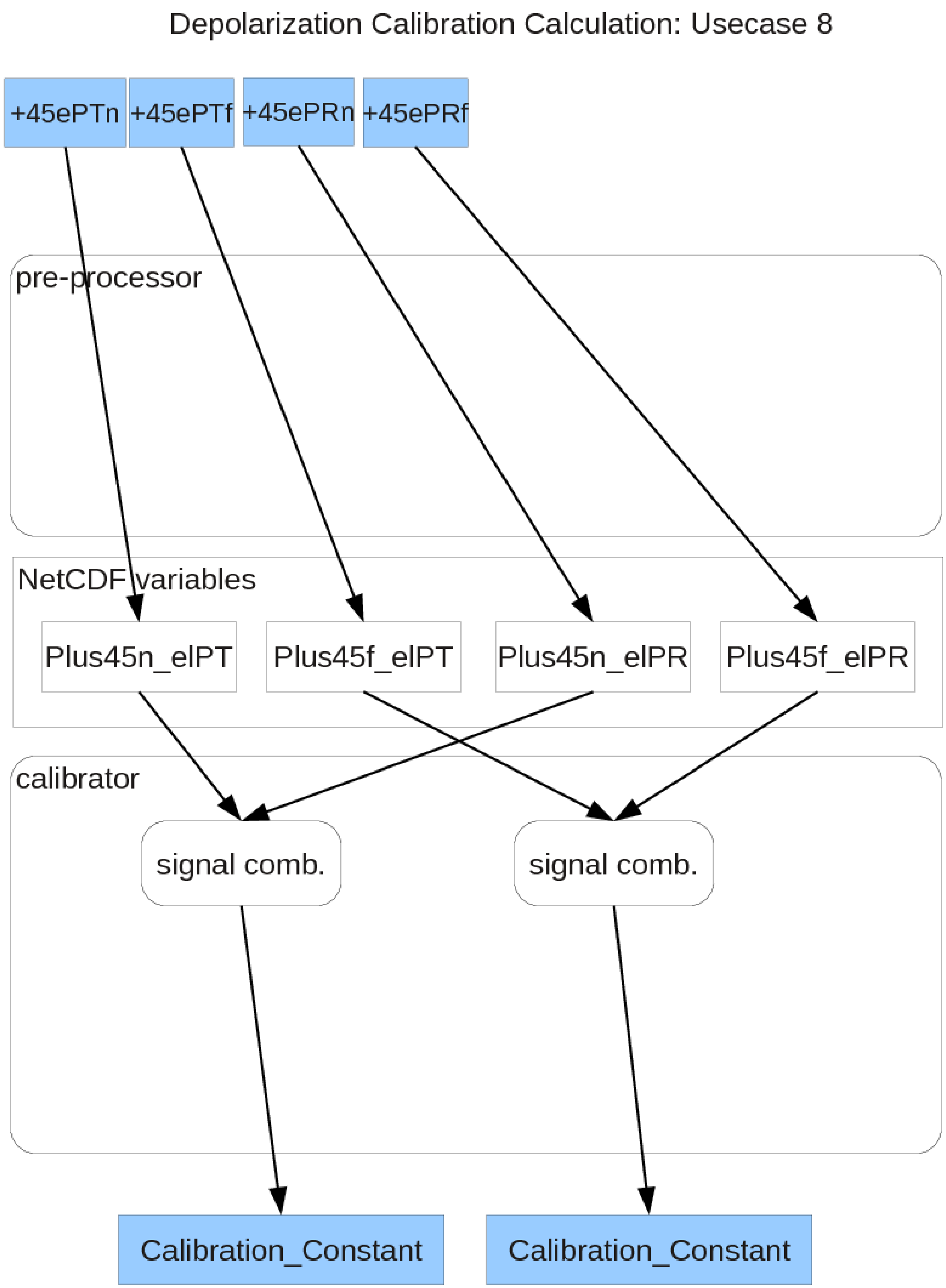

Depolarization Calibration Calculation: Usecase 3
+45ePT
+45ePRn
+45ePRf
pre-processor
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
calibrator
signal comb.
Calibration_Constant

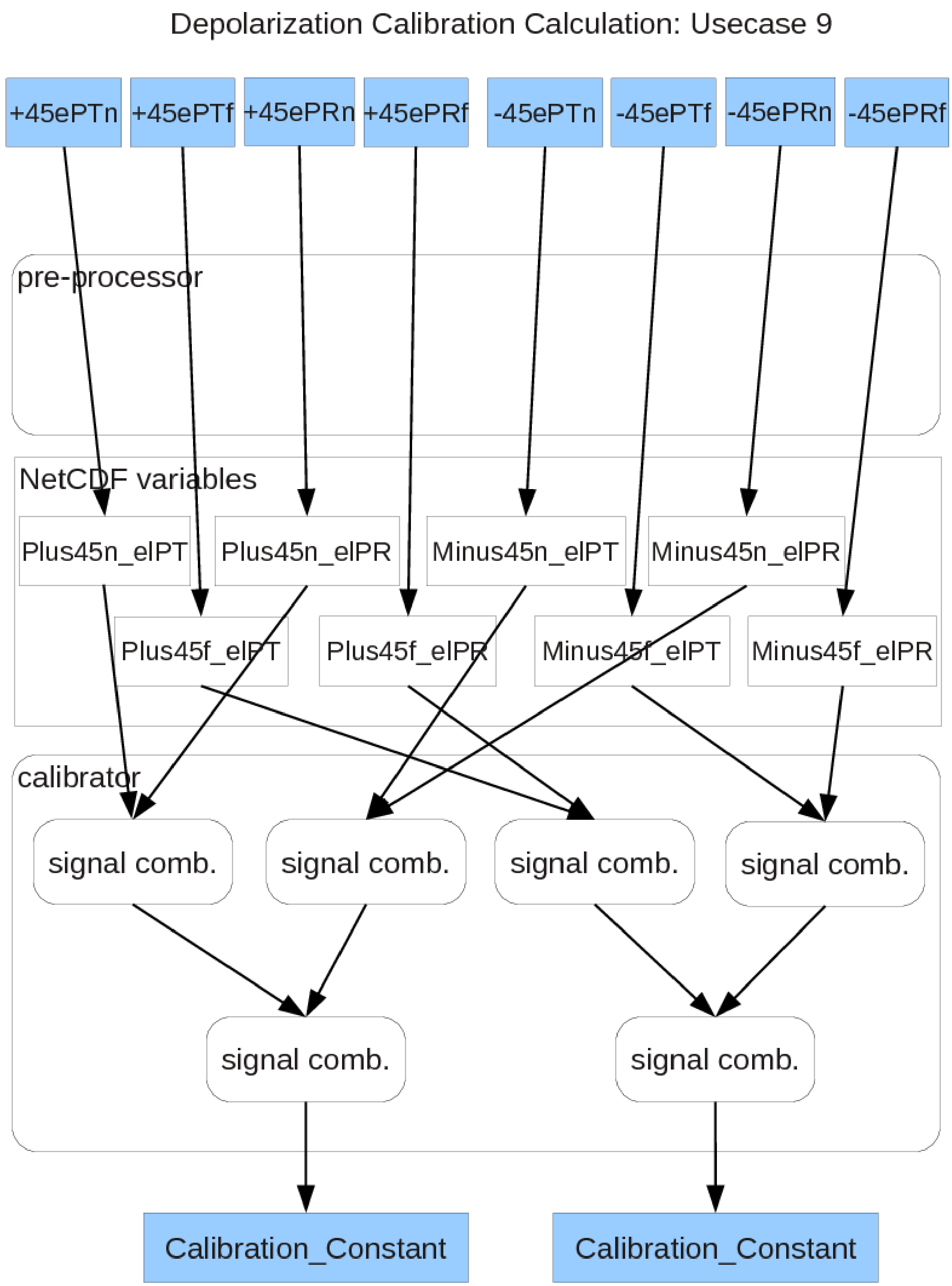

Depolarization Calibration Calculation: Usecase 4
+45ePT
+45ePR
-45ePT
-45ePR
pre-processor
NetCDF variables
Plus45_elPT
Plus45_elPR
Minus45_elPT
Minus45_elPR
calibrator
signal comb.
signal comb.
signal comb.
Calibration_Constant

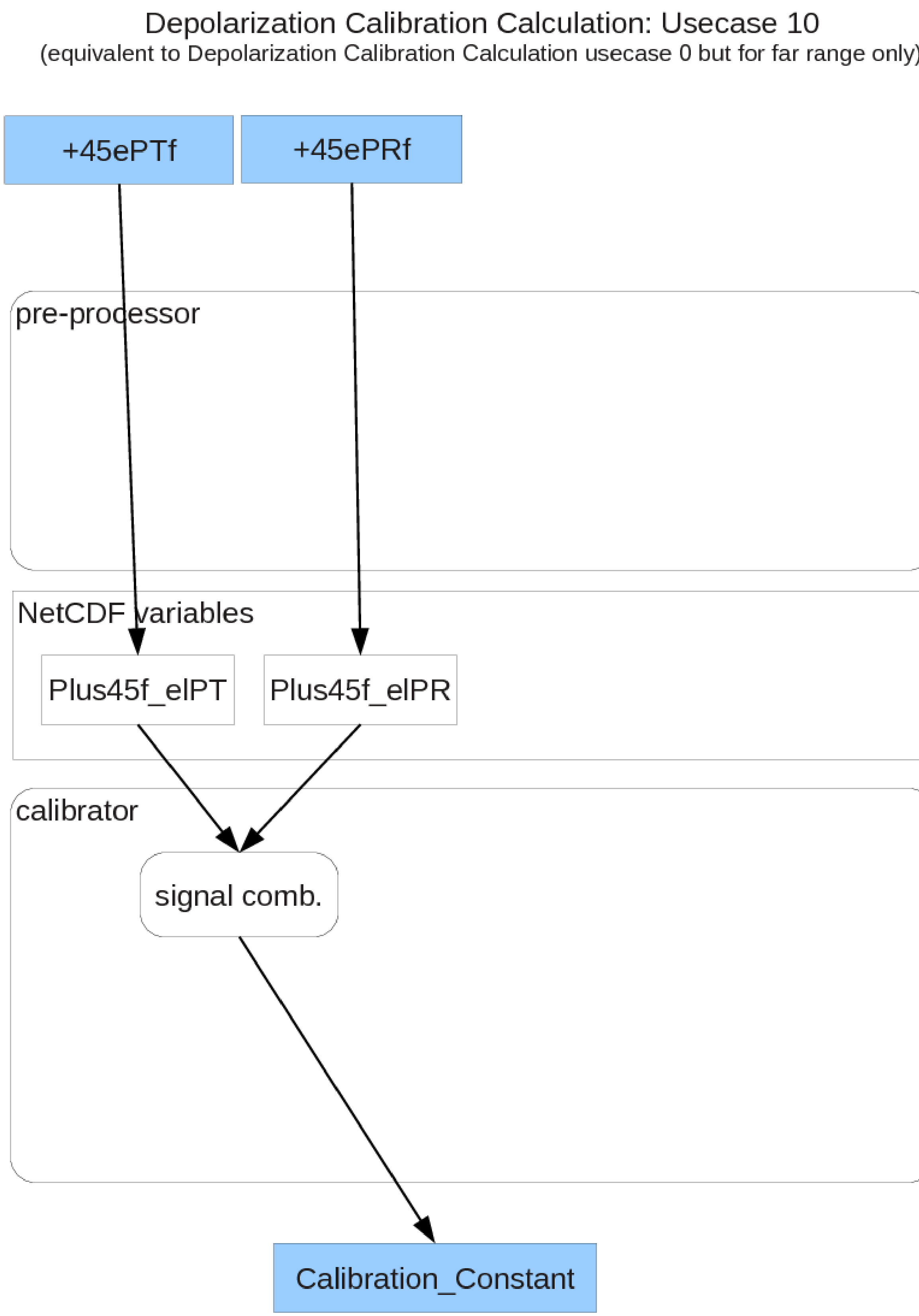

Depolarization Calibration Calculation: Usecase 5
+45ePT
+45ePRn
+45ePRf
-45ePT
-45ePRn
-45ePRf
pre-processor
signal comb.
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
Minus45_elPT
Minus45_elPR
calibrator
signal comb.
signal comb.
signal comb.
Calibration_Constant

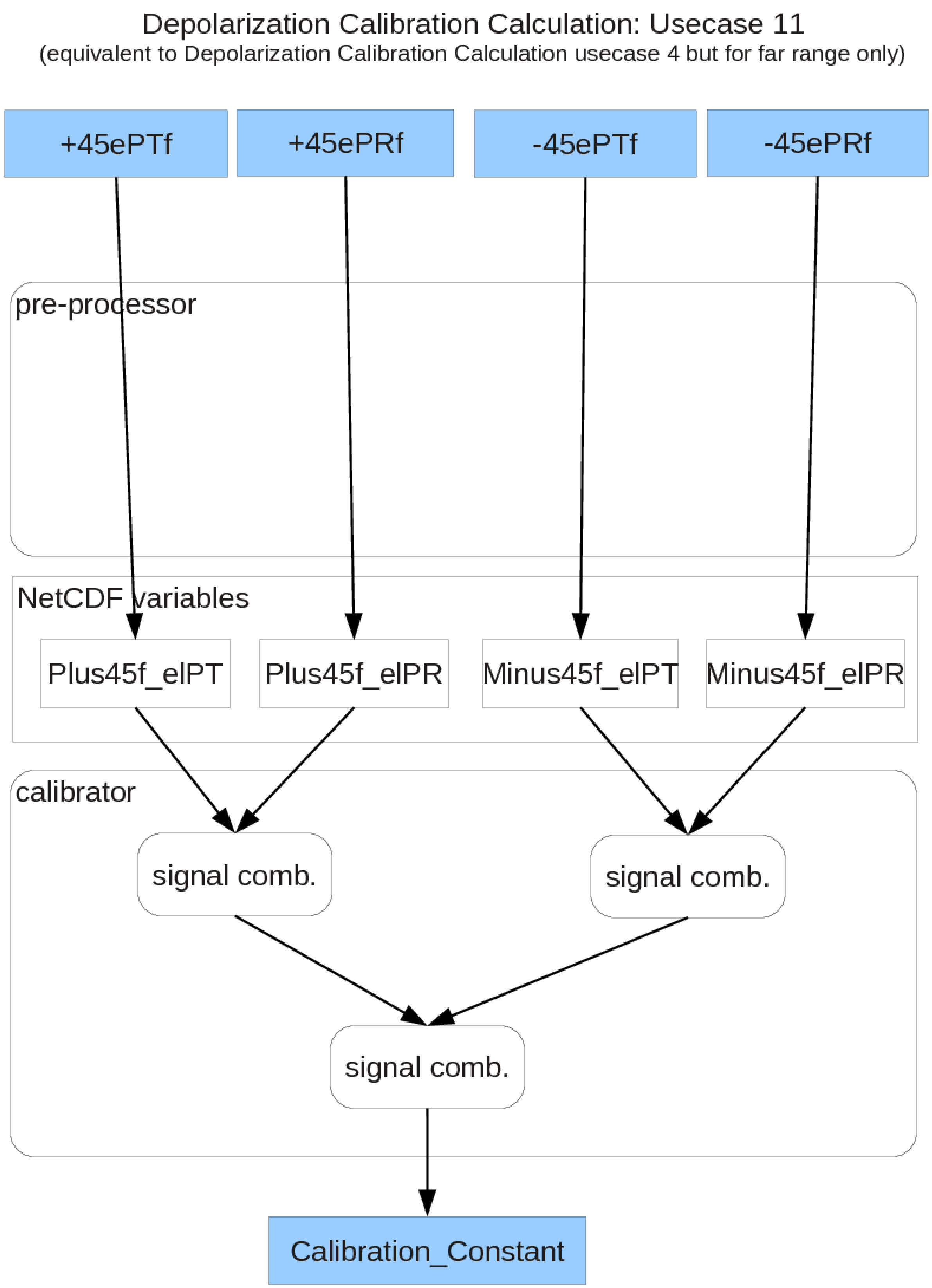

Depolarization Calibration Calculation: Usecase 6
+45ePTn
+45ePTf
+45ePR
-45ePTn
-45ePTf
-45ePR
pre-processor
signal comb.
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
Minus45_elPT
Minus45_elPR
calibrator
signal comb.
signal comb.
signal comb.
Calibration_Constant


Depolarization Calibration Calculation: Usecase 7
+45ePTn
+45ePTf
+45ePRn
+45ePRf
-45ePTn
-45ePTf
-45ePRn
-45ePRf
pre-processor
signal comb.
signal comb.
signal comb.
signal comb.
NetCDF variables
Plus45_elPT
Plus45_elPR
Minus45_elPT
Minus45_elPR
calibrator
signal comb.
signal comb.
signal comb.
Calibration_Constant


Depolarization Calibration Calculation: Usecase 8
+45ePTn
+45ePTf
+45ePRn
+45ePRf
pre-processor
NetCDF variables
Plus45n_elPT
Plus45f_elPT
Plus45n_elPR
Plus45f_elPR
calibrator
signal comb.
signal comb.
Calibration_Constant
Calibration_Constant


Depolarization Calibration Calculation: Usecase 9
+45ePTn
+45ePTf
+45ePRn
+45ePRf
-45ePTn
-45ePTf
-45ePRn
-45ePRf
pre-processor
NetCDF variables
Plus45n_elPT
Plus45n_elPR
Minus45n_elPT
Minus45n_elPR
Plus45f_elPT
Plus45f_elPR
Minus45f_elPT
Minus45f_elPR
calibrator
signal comb.
signal comb.
signal comb.
signal comb.
signal comb.
signal comb.
Calibration_Constant
Calibration_Constant


Depolarization Calibration Calculation: Usecase 10
(equivalent to Depolarization Calibration Calculation usecase 0 but for far range only)
+45ePTf
+45ePRf
pre-processor
NetCDF variables
Plus45f_elPT
Plus45f_elPR
calibrator
signal comb.
Calibration_Constant


Depolarization Calibration Calculation: Usecase 11
(equivalent to Depolarization Calibration Calculation usecase 4 but for far range only)
+45ePTf
+45ePRf
-45ePTf
-45ePRf
pre-processor
NetCDF variables
Plus45f_elPT
Plus45f_elPR
Minus45f_elPT
Minus45f_elPR
calibrator
signal comb.
signal comb.
signal comb.
Calibration_Constant

3.7 Product-Channel Connections

The definition of each SCC product requires the specification of all the lidar channels needed to calculate it. For example, for a Raman backscatter product it is necessary to speficy which elastic and Raman lidar channels need to be used to compute the particle backscatter profile.

The list of the channels to be provided depends on the product type and also on the product usecase.

In particular, the number and the type of the lidar channels to be linked to a product for a particular usecase is given by the input channels (i.e. the channels you find on the top of each usecase schema) given in the definition of the corresponding usecase.

For example, from the schema defining the usecase 5 of Raman backscatter product we find that 4 channels need to be linked:

- 2 elastic channels (one for near range and another for the far range - *elTnr* and *elTfr*)
- 2 Raman channels (again one for near range and one for the far range *vrRN2nr* of *vrRN2fr*).

Lidar channels are linked to a particular product by using the tab 'Product/channel connections' available in each product settings page (from admin section). To make the link two channel characteristics should be provided:

- channel ID
- Signal Type

The channel ID is automatically provided when you click on the small magnifying glass available on the right of channel ID field (tab Product/channel connections) and select the channel you want to use in the pop up list (if the channel ID to be included is already known it is possible to type it directly).

The signal type should be selected among the available options according to the role of the specific channel in the product calculation. To facilitate the consistency between the definition of signal type and what is required by the selected product usecase, the input channels reported in each usecase schema are named according to the available signal types.

Coming back to the example of the usecase 5 of Raman backscatter, for the near range and far range elastic channels the signal types *elTnr* and *elTfr* should be used. In the same way for the near range and far range Raman channels the signal types *vrRN2nr* and *vrRN2fr* should be selected.

> **Warning:** If the definition of the channels in "Product/channel connections" is not consistent with what expected from the specified product usecase, an error reporting inconsistent product definition is reported by ELPP module.

Sygnal types are defined as it follows:

Elastic channels

elT elastic lidar channel (total)

elTnr near range elastic lidar channel (total)

elTfr far range elastic lidar channel (total)

elTunr ultra near range elastic lidar channel (total)

elPR reflected polarization component of elastic lidar channel

elPRnr near range reflected polarization component of elastic lidar channel

elPRfr far range reflected polarization component of elastic lidar channel

elPT transmitted polarization component of elastic lidar channel

elPTnr near range transmitted polarization component of elastic lidar channel

elPTfr far range transmitted polarization component of elastic lidar channel

Raman channels

vrRN2 ro-vibration Raman channel due to atmospheric N2 molecule

vrRN2nr near range ro-vibrational Raman channel due to atmospheric N2 molecule

vrRN2fr far range ro-vibrational Raman channel due to the atmospheric N2 molecule

pRRlow low quantum number pure rotational Raman channel. Typically used together with high quantum number pure rotational Raman channel(s) to perform atmospheric temperature measurements

pRRlownr near range low quantum number pure rotational Raman channel. Typically used together with high quantum number pure rotational Raman channel(s) to perform atmospheric temperature measurements

pRRlowfr far range low quantum number pure rotational Raman channel. Typically used together with high quantum number pure rotational Raman channel(s) to perform atmospheric temperature measurements

pRRhigh high quantum number pure ritational Raman channel. Typically used together with low quantun number pure rotational Raman channel(s) to perform atmospheric temperature measurements

pRRhighnr near range high quantum number pure ritational Raman channel. Typically used together with low quantun number pure rotational Raman channel(s) to perform atmospheric temperature measurements

pRRhighfr far range high quantum number pure ritational Raman channel. Typically used together with low quantun number pure rotational Raman channel(s) to perform atmospheric temperature measurements

Polarization calibration channels

+45elPR plus 45 degrees rotated (in the polarization plane) reflected polariazion component of elastic lidar channel

+45elPRnr plus 45 degrees rotated (in the polarization plane) near range reflected polarization component of elastic lidar channel

+45elPRfr plus 45 degrees rotated (in the polarization plane) far range reflected polarization component of elastic lidar channel

+45elPT plus 45 degrees rotated (in the polariation plane) transmitted component of elastic lidar channel*

+45elPTnr plus 45 degrees rotated (in the polarization plane) near range transmitted polarization component of elastic lidar channel

+45elPTfr plus 45 degrees rotated (in the polarization plane) far range transmitted polarization component of elastic lidar channel

-45elPR minus 45 degrees rotated (in the polarization plane) reflected polariazion component of elastic lidar channel

-45elPRnr minus 45 degrees rotated (in the polarization plane) near range reflected polarization component of elastic lidarchannel

-45elPRfr minus 45 degrees rotated (in the polarization plane) far range reflected polarization component of elastic lidar channel

-45elPT minus 45 degrees rotated (in the polariation plane) transmitted component of elastic lidar channel

-45elPTnr minus 45 degrees rotated (in the polarization plane) near range transmitted polarization component of elastic lidar channel

-45elPTfr minus 45 degrees rotated (in the polarization plane) far range transmitted polarization component of elastic lidarchannel

Elastic channels are requested for the definition of product types:

- Raman backscatter
- Elastic backscatter
- Raman backscatter and linear depolarization
- Elastic backscatter and linear depolarization
- Lidar ratio and extinction

Raman channels are requested for the definition of the product types:

- Raman backscatter
- Raman backscatter and linear depolarization
- Extinction only
- Lidar ratio and extinction

Calibration channels are requested for the definition of the product types:

- Linear polarization calibration

3.8 Adding other equipment

CHAPTER 4

Data processing

Contents:

4.1 Overview

The "Data processing" section of the SCC is used to upload measurements and ancillary files, monitor the processing progress, and view the output results of the SCC.

The data processing documentation consist of three subsections:

- The procedures of submitting new data for processing are described in the *Uploading measurements* section.
- The procedures of submitting ancillary files (i.e. sounding, overlap, and lidar ratio files) that are used in the processing are described in the *Ancillary files* section.
- The procedure to find and explore existing measurements and ancillary files are detailed in the *Browse uploaded measurements* section.

4.1.1 Quick start

Note: The following discussion assumed that you have already set-up the system settings in the "Admin" section (see *Station administration* for details).

Upload a measurement

In order to start processing lidar data, you will need to upload them on the SCC server. The files should be in netcdf format, following the specific format described in *The SCC input netCDF file format*. There is no restriction imposed in the filename, but we suggest to use the format <measurement_ID>.nc. You can upload using the "Quick upload" link in the menu.

When you submit the processed measurements in the quick upload form, you will need to specify the system configuration that was used to perform the measurement. The *channels* that are defined in the file should be the ones defined in the selected system. The uploading of ancillary files or the selection of categories for the measurement are optional.

When the file is uploaded, it is checked if it conforms to the needed netCDF format. The check guarantees that the mandatory variables are present in the file and that they have the correct format. Note that the check is not exhaustive, in the sense it does not guarantee that the file is able to be correctly processed; the aim of this check is to detect as early as possible common problems in the file. If some errors are found, the file will be deleted from the server.

Ancillary files

If ancillary files are needed for the processing two things should be done:

1. **The ancillary file names should be defined in the submitted measurement file**. In contrast to the measurement files, the file name of the ancillary files should follow a specified format, described in *The SCC input netCDF file format*. In brief, a sounding file should have a name rs_YYYYMMDDccNN.nc, a overlap file a name ov_YYYYMMDDccNN.nc, and a lidar ratio file a name lr_YYYYMMDDccNN.nc, where YYYY is a year, MM is the month, DD is the day, cc is the EARLINET station call sign and NN is a number.
2. **The ancillary files should be uploaded on the server**. The uploading can be done in the same form as before, or independently in the "Upload ancillary" form. If an ancillary file required by a measurement is not present in the database, this file will appear as "Missing" and the processing of the measurement will not start and will be marked as "pending". When the missing file is uploaded, its status will change to "OK" and the measurement will be processed.

Monitor processing progress

When uploading finishes successfully, you will be transfered to the **measurement status page**. There you can monitor the progress of the processing through the status symbols. The three symbols correspond to the phase of processing:

Uploading It indicates if everything needed for processing is provided; practically this indicates if the measurement file and all needed ancillary files have been uploaded.

Preprocessing This indicates the progress of the preprocessor module.

Optical processing This indicates if progress of the optical processing module (ELDA).

These three indicators currently support four states:

Gray (Not started) A gray indicator means that the process did not start yet.

Orange (In progress) An orange indicator means that the process is currently performed.

Green (Success) A green indicator means that the process finished successfully.

Red (Fail) A red indicator means that the process failed.

While the processing is in progress the page will automatically refresh every 10 seconds.

View output

When the processing of your measurement finishes successfully, you can browse and download the results.

To **browse** the results, you must select the "Output" tab at the bottom part of the page. There you will find links to view graphs of the resulting optical products.

To **download** the results, you can use the links on the "File actions" submenu. Depending on the state of the processing you can downlaod *preprocessed singals*, *optical products*, and *optical product plots*.

4.2 Uploading measurements

New measurements can be uploaded using the "Quick Upload" link in the data processing submenu. There, you will need to fill and submit the provided form.

4.2.1 Form fields

System (Required) This field specifies the system configuration that was used to perform the measurement. The available systems can be managed in the "Station Admin" section of the site. You can later change the system used in the processing of the measurement again in the "Station Admin" section.

Data file (Required) In this field you have to select the measurement file that will be processed. The file has to be in the specific format described in *The SCC input netCDF file format*.

Sounding file (Optional) In this optional field you can submit a sounding file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which sounding file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Overlap file (Optional) In this optional field you can submit a overlap file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which overlap file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Lidar ratio file (Optional) In this optional field you can submit a lidar ratio file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which lidar ratio file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Comments (Optional) In this field you can add a small comment (up to 100 characters) to the measurement.

Categories (Optional) You can assign a number of categories to the uploaded measurement. The categories you assign can be lated changed in the in the "Station Admin" section of the site. You can select multiple categories by holding down the "Control" key (PC) or "Command" key (MAC) while selecting.

Note: Uploading the measurement file can take considerable time, depending on the size of the file and the speed of your Internet connection. In the current version of the site, no feedback is provided during the upload. You can get information on your upload progress depending on your browser.

Firefox You can install this add-on.

Chrome A progress indicator is built-in.

IE 9- No progress bar is available (as far as we know).

4.2.2 Form Validation

The aim of form validation is to detect errors in your submitted data as early as possible. In this way you can save time by correcting errors before the processing of your data fails.

The data you provide in the form are check in two stages, before you submit and after you submit the form.

Before you submit the form

Before you submit the form the following checks are performed:

- All the necessary fields (i.e. "system" and "data file") need to be submitted.
- The ancillary file names have to be in the appropriate format described in *The SCC input netCDF file format*. In brief, a sounding file should have a name rs_YYYYMMDDccNN.nc, a overlap file a name ov_YYYYMMDDccNN.nc, and a lidar ratio file a name lr_YYYYMMDDccNN.nc, where YYYY is a year, MM is the month, DD is the day, cc is the EARLINET station call sign and NN is a number.

After you submit the form

After you submit the form:

- The uploaded **data file** needs to have a unique name.
- The **measurement id** defined in the file should *not* exist in the SCC.
- All channels declared in the measurement file should correspond to declared channels in the SCC.
- The submitted measurement file should have the correct netCDF format. All mandatory variables and attributes should be present in the file. All variables and attributes should have the correct type (float, integer, etc). This check is not exhaustive, and it is possible that some errors are **not** detected. For example "conditionally" mandatory variables are not checked (as, for example "LR_Input", that is mandatory if elastic backscatter retrievals have to be done).

If ancillary files are submitted

Additionally, if ancillary files are also uploaded, the following checks are performed:

- The submitted measurement file should have the correct netCDF format. All mandatory variables and attributes should be present in the file. All variables and attributes should have the correct type (float, integer, etc).
- A file with the same filename should not exist in the SCC or, if it exists, it should have status "Missing" or "Error". If a file with the same filename exists, and its status is "OK" the submitted file is rejected.

4.3 Ancillary files

Together with your measurement data, you can submit ancillary files that will be used in the processing procedure. The files that are currently supported are **sounding files**, **overlap files**, and **lidar ratio files**.

These files can be submitted either together with a measurement, using the "Quick upload" form, or on their own, using the "Upload ancillary" form. Both methods are equivalent and will not affect which files is used for processing your measurements. The file that will be used are defined in the *measurement file* using the appropriate attributes as described in the *The SCC input netCDF file format* section.

If one of your measurements requires ancillary files for its processing, the actual processing of the data will start only after all the required files have been uploaded. The files that are missing will appear in the ancillary file list with "Missing" status. When all these files have been uploaded the processing of the measurement will start automatically.

4.3.1 Upload ancillary

In the "Upload ancillary" form you need to specify to submit at least one ancillary file.

Form fields

Station (Required) You need to specify the EARLINET station that is related to this file. In most cases you will just have to select your station.

Sounding file (Optional) In this optional field you can submit a sounding file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which sounding file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Overlap file (Optional) In this optional field you can submit a overlap file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which overlap file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Lidar ratio file (Optional) In this optional field you can submit a lidar ratio file that will be used by your measurements in the processing. The file needs to follow the netCDF file format specified in the pdf file found in the *The SCC input netCDF file format* section. The file you submit will not be necessarily connected with the submitted measurement; which lidar ratio file will be used in the processing of each measurement is specified in the measurement's file. The file you upload will, nevertheless, remain on the SCC and will be used whenever needed.

Form Validation

The aim of form validation is to detect errors in your submitted data as early as possible. In this way you can save time by correcting errors before the processing of your data fails.

When the files are submitted the following checks are performed:

- At least one file needs to be submitted.
- The submitted measurement file should have the correct netCDF format. All mandatory variables and attributes should be present in the file. All variables and attributes should have the correct type (float, integer, etc).
- A file with the same filename should not exist in the SCC or, if it exists, it should have status "Missing" or "Error". If a file with the same filename exists, and its status is "OK" the submitted file is rejected.

4.3.2 Browse ancillary files

You can browse existing and missing ancillary files in the "Ancillary files" page of the Data processing sub menu.

You can filter the results of the list using the **Filter** box above the form. The filter searches for matches in all columns of the table, i.e you can use it to filter the results by filename, stations, data, and status.

You can view more information about the ancillary file following the link on the name of the file.

4.3.3 View ancillary file details

In the ancillary file status page you can view more detailed information about the selected file. The following info are provided.

Status The status of the file can have three values. **Ok** means that the file has been uploaded successfully and is ready to be used in the processing. **Missing** status means that one or more of the submitted measurements have requested this file, but the file has not been submitted yet. An **Error** status indicates that the file has been submitted but there was an error while reading its content; you will need to correct its content and resubmit the file.

Interpolation This indicates the interpolation type that will be used on the content of the ancillary file. You can change the interpolation type in the admin section following the "Edit in admin" link in the File actions sub menu. The algorithms currently implemented are "Linear interpolation" and "Natural Cubic Spline". If no value is selected the natural cubic spline algorithm will be used.

Station The EARLINET station that is the related to this ancillary file

Submitted on The date and time that the file was submitted. If the file is missing, but is required by an uploaded measurement, this date indicated the date and time of the measurement upload.

At the bottom part of the page you can see a list of **Related measurements**. These are measurements that use the specific ancillary file for their processing.

4.4 Browse uploaded measurements

4.4.1 Search measurements

You can browse already submitted measurements using the "Search Measurements" link in the data processing sub-menu. All the fields in the form are optional.

Form fields

Station The EARLINET station call sign that performed measurement.

System The system that was used to process the measurement.

Start date The search start date. All measurements before that date will be filtered. The correct format is "YYYY-MM-DD HH:MM:SS".

Stop date The search stop date. All measurements after that date will be filtered. The correct format is "YYYY-MM-DD HH:MM:SS".

Upload status The measurement upload status. Four choices are possible (0, 1, -127, 127).

Preprocessing status The measurement preprocessing status. Four choices are possible (0, 1, -127, 127).

Optical processing status The measurement optical processing status. Four choices are possible (0, 1, -127, 127).

Categories The measurement categories. Only measurements that belong to all categories will be returned.

Results

You can filter the results of the list using the **Filter** box above the form. The filter searches for matches in all columns of the table, i.e you can use it to filter the results by id, system, start time and duration. You can also filter by status using a comma separated exit codes (ex. 127,127,-127).

The results can be sorted by clicking on the table labels.

4.4.2 View Measurement details

In the measurement status page you can view more detailed information about the selected file. The following info are provided.

Measurement info

System The system that was used to process the measurement. You can change the system used in the processing of the measurement in the admin section following the "Edit in admin" link in the File actions sub menu. If you change the selected system, the measurement will be reprocessed automatically, and all old output files will be deleted.

Start The start time of the measurement.

Stop The stop time of the measurement.

Sounding file The sounding file that will be used in processing the measurement. The status of the file is also provided (see the *View ancillary file details* section for further info).

Overlap file The overlap file file that will be used in processing the measurement. The status of the file is also provided (see the *View ancillary file details* section for further info).

Lidar ratio file The lidar ratio file that will be used in processing the measurement. The status of the file is also provided (see the *View ancillary file details* section for further info).

Categories The selected categories for this measurement. You can change the selected categories in the admin section following the "Edit in admin" link in the File actions sub menu.

Created on The date and time that the file was first submitted on the SCC.

Last update The date and time that this file was last updated.

Comment User comments for this file. You can add / modify the comment in the admin section following the "Edit in admin" link in the File actions sub menu.

The status of the measurement is represented by three icons. The icons have different colors depending on the status of each process.

Uploading (up arrow icon) It indicates if everything needed for processing is provided; practically this indicates if the measurement file and all needed ancillary files have been uploaded.

Preprocessing (gear icon) This indicates the progress of the preprocessor module.

Optical processing (graph icon) This indicates if progress of the optical processing module (ELDA).

These three indicators currently support four states:

Gray (Not started) A gray indicator means that the process did not start yet.

Orange (In progress) An orange indicator means that the process is currently performed.

Green (Success) A green indicator means that the process finished successfully.

Red (Fail) A red indicator means that the process failed.

File actions

The following actions are available, depending on the state of the processing.

Reprocess The preprocessed and optical products will be recalculated. All previously produced files will be deleted.

Rerun optical module Only the optical products will be recalculated. All previously produced optical files will be deleted.

Download pre-processed files Download all preprocessed files in a zip file. The action is available only if the pre-processor finished successfully.

Download optical products Download all optical files in a zip file. The action is available only if the optical module (ELDA) finished successfully.

Download plots Download all optical files plots in a zip file. The action is available only if the optical module (ELDA) finished successfully.

4.5 Preprocessor exit codes

After a measurement is processed by the preprocessor, an exit code is provided that can help you understand if everything went OK, and if not, what was the problem. The implemented exit codes are the following.

0 Finished without errors

1 Failure in memory allocation

2 Configuration file not found

3 Configuration file error: INPUT_DIR not defined

4 Configuration file error: OUTPUT_DIR not defined

5 Configuration file error: Cannot read in INPUT_DIR

6 Configuration file error: Cannot write on OUTPUT_DIR

7 Configuration file error: Cannot read in SOUNDINGS_DIR

8 Configuration file error: Cannot read in OVERLAP_DIR

9 Configuration file error: Cannot read in LIDARRATIO_DIR

10 Configuration file error: Cannot write on LOG_DIR

12 Wrong usage in calling the script

13 Cannot create output directory

14 An internal error occurred

15 Configuration file error: Cannot read in CLOUDMASK_DIR

16 Configuration file error: Cannot read in CLOUDNET_DIR

17 Configuration file error: Cannot write on RAW_SKIPPING_DIR

18 Cannot create cloudnet output directory

19 Cannot read CA file

20 Found one or more channels defined in the product definition but not linked to the used system configuration. Please check that all the channels in your products are also linked to the used system configuration

21 Cannot connect to SCC_DB

22 Failed in quering SCC_DB

23 Failed in fetching query of SCC_DB

24 Found error(s) in SCC_DB for the submitted Measurement_ID

25 Measurement ID not found in the selected database

26 Measurement_ID not unique in the selected database

27 Unknown usecase in SCC_DB

28 One or more mandatory datetime field(s) are not set into the database

29 Found mismatch among usecase, product type and given channels

30 All the products associated to a system configuration should be processed with the same processor module.

40 Unknown product type in SCC_DB

41 Cannot open NetCDF input file

42 Dimension "time" not found in NetCDF input file

43 Dimension "channels" or "channel" not found in NetCDF input file

44 Dimension "points" not found in NetCDF input file

45 Dimension "scan_angles" not found in NetCDF input file

46 Global attribute "Measurement_ID" not found or not correctly defined in NetCDF input file

47 Incorrect definition of global attribute "Measurement_ID" in NetCDF input file

48 Global attribute "RawData_Start_Date" not found in NetCDF input file

49 Global attribute "RawData_Start_Time_UT" not found in NetCDF input file

50 Global attribute "RawData_Stop_Time_UT" not found in NetCDF input file

51 Variable "Laser_Pointing_Anlge" not found and/or not defined correctly in NetCDF input file

52 "Laser_Pointing_Angle" variable cannot be undefined and should be between 0 and 90.

53 Variable "Laser_Pointing_Angle_of_Profiles" not found and/or not defined correctly in NetCDF input file

54 Variable "Laser_Shots" not found and/or not defined correctly in NetCDF input file

55 Found negative or not defined value in "Laser_Shots" array

56 Cannot write intermediate NetCDF file

57 Variables "Pol_Calib_Range_Min" and/or "Pol_Calib_Range_Max" not found and/or not defined correctly in NetCDF input file

58 Invalid values for variables "Pol_Calib_Range_Min" and/or "Pol_Calib_Range_Max" in NetCDF input file

70 Generic NetCDF error

71 Dimension "time" cannot be zero

72 Dimension "channels" cannot be zero

73 Dimension "points" cannot be zero

74 Natural cubic spline interpolation not possible

75 lfit: no parameters to be fitted

76 gaussj: Singular Matrix

77 Not correct type in NetCDF variable definition

78 Not correct number of dimensions in NetCDF variable definition

79 Not correct type in NetCDF global attribute definition

80 Found invalid value of latitude. Check the database value or the value reported in the raw NetCDF input file

81 Found invalid value of longitude. Check the database value or the value reported in the raw NetCDF input file

82 Found invalid value of altitude. Check the database value or the value reported in the raw NetCDF input file

83 Found invalid value of emission wavelength. Check the database value or the value reported in the raw NetCDF input file

84 Found invalid value of detection wavelength. Check the database value or the value reported in the raw NetCDF input file

101 Found wrong value for integration time in SCC_DB

102 Overlap file not registered in the database. Please register it and then re-run the SCC

103 Lidarratio file not registered in the database. Please register it and then re-run the SCC

104 Sounding file not registered in the database. Please register it and then re-run the SCC

105 Not all required depolarization calibrations is available in SCC_DB. Please submit at least one depolarizatio calibration measurement for each required calibration and link them to the product to be calibrated

106 Found no valid values for depolarization constant in the SCC_DB

107 Unkown depolarization calibration calculation method

108 No depolarization cross-talk correction factors (G and H) are available in SCC_DB for the depolarization channels. Please submit them

109 Found no valid values for depolarization cross-talk correction factors in the SCC_DB

110 No correction factor (K) for depolarization calibration is available in SCC_DB. Please provide at least one value for correction factor and link it to the product to be calibrated

111 Found no valid values for depolarization constant correction factor in the SCC_DB

112 Unkown depolarization correction factor calculation method

113 Found SCC_DB inconsistencies in the definition of transmitted depolarization channels. Please be sure that all the products containing the reflected depolarization channel includes the corresponding transmitted depolarization channel as well. Check also the signal type and the scattering type are set correctly.

114 Found SCC_DB inconsistencies in the definition of reflected depolarization channels. Please be sure that all the products containing the transmitted depolarization channel includes the corresponding reflected depolarization channel as well. Check also the signal type and the scattering type are set correctly.

115 For one or more polarization calibration parameters you have selected to use the option "Use the most recent available calibration previous to the measurement date". No calibrations according to this criteria has been found. So please select another calibration options of provide at least one calibration measurements with a date previous to the current measurement date.

121 Dimension "nb_of_time_scales" not found in NetCDF input file

122 Incorrect definition of global attribute "RawData_Start_Date" in NetCDF input file

123 Incorrect definition of global attribute "RawData_Start_Time_UT" in NetCDF input file

124 Incorrect definition of global attribute "RawData_Stop_Time_UT" in NetCDF input file

125 Variables “channel_string_ID” and/or “channel_ID” not found and/or not defined correctly in the NetCDF input file

126 Missing one or more channels in NetCDF input file

127 Found wrong value(s) in variable “Background_Low” end/or “Background_High”

128 Variable “Background_Low” not found and/or not defined correctly in NetCDF input file

129 Variable “Background_High” not found and/or not defined correctly in NetCDF input file

130 Variable “id_timescale” not found and/or not defined correctly in NetCDF input file

131 Variable “Raw_Data_Start_Time” not found and/or not defined correctly in NetCDF input file

132 Variable “Raw_Data_Stop_Time” not found and/or not defined correctly in NetCDF input file

133 Variable “Raw_Lidar_Data” not found and/or not defined correctly in NetCDF input file

134 Found not integer values in photoncounting signal

135 Variable “Molecular_Calc” not found and/or not defined correctly in NetCDF input file

136 Variable “Molecular_Calc” has not valid value

137 Variable “Pressure_at_Lidar_Station” not found and/or not defined correctly in NetCDF input file

138 Variable “Temperature_at_Lidar_Station” not found and/or not defined correctly in NetCDF input file

139 Found wrong value(s) for channel first signal range bin. Please check your SCC channels configuration of the variable “First_Signal_Rangebin” in NetCDF input file

140 Variable “DAQ_Range” not found and/or not defined correctly in NetCDF input file

143 Variable “Raw_Bck_Start_Time” not found and/or not defined correctly in NetCDF input file

144 Variable “Raw_Bck_Stop_Time” not found and/or not defined correctly in NetCDF input file

145 Global attribute “RawBck_Start_Date” not found in NetCDF input file

146 Global attribute “RawBck_Start_Time_UT” not found in NetCDF input file

147 Global attribute “RawBck_Stop_Time_UT” not found in NetCDF input file

148 Found not compatible values of variable “id_timescale”. All the values of “id_timescale” should be less then the dimension “nb_of_time_scales”

150 Global attribute “Sounding_File_Name” not found in NetCDF input file

151 Sounding NetCDF input file not found

152 Dimension “points” not found in Sounding NetCDF input file

153 Global attribute “Sounding_Start_Date” not found in the Sounding NetCDF input file

154 Global attribute “Sounding_Start_Time” not found in the Sounding NetCDF input file

155 Global attribute “Latitude_degrees_north” not found in the Sounding NetCDF input file

156 Global attribute “Longitude_degrees_east” not found in the Sounding NetCDF input file

157 Global attribute “Altitude_meter_asl” not found in the Sounding NetCDF input file

158 Variable “Altitude” not found and/or not defined correctly in the Souding NetCDF input file

159 Variable “Temperature” not found and/or not defined correctly in the Souding NetCDF input file

160 Variable “Pressure” not found and/or not defined correctly in the Souding NetCDF input file

161 Sounding file error: "Altitude" array should contain altitudes in ascendent order (from the lowest point to the highest one)

162 Cannot found variable "LR_Input" within NetCDF input file. This variable is mandatory for elastic backscatter retrievals

163 Found invalid value(s) for Variable "LR_Input"

164 Global attribute "LR_File_Name" not found in NetCDF input file

165 Lidar ratio NetCDF input file not found

166 Dimension "points" not found in lidar ratio NetCDF input file

167 Dimension "products" not found in lidar ratio NetCDF input file

168 Global attribute "Lidar_Station_Name" not found in the lidar ratio NetCDF input file

169 Variable "Altitude" not found and/or not defined correctly in the lidar ratio NetCDF input file

170 Variable "product_ID" not found and/or not defined correctly in the lidar ratio NetCDF input file

171 Variable "Lidar_Ratio" not found and/or not defined correctly in the lidar ratio NetCDF input file

172 Variable "product_id" in the lidar ratio NetCDF input file contains data that are not consistent with the SCC database values

173 Variable "Lidar_Ratio" in the lidar ratio NetCDF input file contains undefined values within the valid altitude range defined by the corresponding product. Please remove them.

174 Lidarratio file error: "Altitude" array should contain altitudes in ascendent order (from the lowest point to the highest one)

175 Overlap NetCDF input file not found

176 Dimension "points" not found in Overlap NetCDF input file

177 Dimension "channels" not found in Overlap NetCDF input file

178 Global attribute "Lidar_Station_Name" not found in the Overlap NetCDF input file

179 Global attribute "Overlap_Measurement_Date" not found in the Overlap NetCDF input file

180 Variable "Altitude" not found and/or not defined correctly in the Overlap NetCDF input file

181 Variable "channel_ID" not found and/or not defined correctly in the Overlap NetCDF input file

182 Variable "Overlap_Function" not found and/or not defined correctly in the Overlap NetCDF input file

183 Overlap file error: "Altitude" array should contain altitudes in ascendent order (from the lowest point to the highest one)

184 Sounding file error: "Temperature" array should contain atmospheric temperature values in degrees Celsius

185 Sounding file error: "Pressure" array should contain atmospheric pressure values in hPa

186 Lidar ratio file error: "Lidar_Ratio" array should contain valid lidar ratio values in sr

187 Overlap file error: "Overlap_Function" array should contain valid overlap correction values

188 Variable "Lidar_Ratio_Error" not defined correctly in the lidar ratio NetCDF input file

189 Dimension "time" is zero in the output file. This may be due to few cloud-free profile within integration time or the integration time is too small. Please check that thare are enought (>2) cloud-free profiles within the integration time in the submitted timeseries

190 Unable to use Poisson statistic in Montecarlo simulation. Found not integer values

191 Found wrong value(s) for "Background_Mode" in NetCDF file or in SCC_DB

192 Found wrong value(s) for variable "Dead_Time_Corr_Type" in NetCDF file or in SCC_DB

193 Cannot apply dead time correction. Please check the dead time value and the your photoncouning raw data

194 Found negative number of counts in lidar data!

195 Found wrong value(s) in variable "Raw_Bck_Start_Time" end/or "Raw_Bck_Stop_Time" and/or in laser repetition rate value

196 Background profiles don"t contain the same number of valid data point of the corresponding lidar profile

197 Cannot calculate errors using Montecarlo simulation within trigger delay correction

198 Linear interpolation not possible

199 Too few cloud-free lidar profiles within the integration time. Please check the integration time is not too small or/and there are enought cloud-free regions in the submitted timeseries

200 Cannot calculate the errors after time integration

201 Found timescales not syncronized

202 Cannot calculate errors using Montecarlo simulation within interpolation routine

203 Gluing between near range and far range signals not possible. No suitable overlap region can found

204 Gluing between near range and far range signals not possible. Poor linear correlation

205 Gluing between near range and far range signals not possible. Too noisy signal(s)

206 Gluing between near range and far range signals not possible. Slope test not passed

207 All the profiles within the submitted timeseries seem contaminated by low clouds. Cannot calculate aerosol optical properties

211 Found wrong value(s) for scattering type id

212 Found wrong value(s) for detection type id

213 Found wrong value(s) for range type id

214 Too few lidar points to calculate atmospheric background. At least 10 lidar rangebins are required within the background subtraction range.

215 Raw time resolution is greater than the selected integration time

216 Gluing between near range and far range signals not possible. Too large difference between signals in gluing point

217 Unkown interpolation type.

218 Found wrong value(s) for signal type id or product channel(s) missing in system channel connections

219 Found range resolution in raw lidar data greater than the value in SCC_DB

220 Error in calculating molecular depolarization ratio

221 Error in getting interferencial filter information for depolarization channels

222 Found too few valid datapoints. Please check the variable "Raw_Lidar_Data" in the raw input file contains at least 15 valid points for all the channels

223 Found not physical values after the interpolation of the sounding file. Try to change the interpolation type (to linear) to fix the problem.

224 Found not physical values after the interpolation of the lidar ratio file. Try to change the interpolation type (to linear) to fix the problem.

225 Found not physical values after the interpolation of the overlap file. Try to change the interpolation type (to linear) to fix the problem.

226 Found incompatible value for the emission wavelength in one or more lidar channels

227 Found not physical values after the interpolation of the model data. Try to change the interpolation type (to linear) to fix the problem.

228 Polarization calibration product and the corresponding product to calibrate refer to different stations!

230 Found invalid value of laser repetition rate. Check the database value or the value reported in the raw NetCDF input file

231 All negative heights above lidar station in sounding file. Please be sure that the heights above sounding station and the sounding station altitude are reported correctly in the sounding file. Check also that your lidar station altitude is correct.

232 All negative heights above lidar station in GDAS file. This shouldn"t happen! Try to force another model or use standard atmosphere or external sounding.

233 Found invalid value of detection wavelength fwhm. Check the database value or the value reported in the raw NetCDF input file

234 Found invalid value of range resolution. Check the database value or the value reported in the raw NetCDF input file

235 Cannot open cloud mask NetCDF file

236 Dimension "time" not found or not defined correctly in the cloudmask NetCDF file

237 Dimension "level" not found or not defined correctly in the cloudmask NetCDF file

238 Variable "time" and/or "time_bounds" not found or not defined correctly in the cloudmask NetCDF file

239 Variable "altitude" not found or not defined correctly in the cloudmask NetCDF file

240 Variable "latitude" not found or not defined correctly in the cloudmask NetCDF file

241 Variable "longitude" not found or not defined correctly in the cloudmask NetCDF file

242 Variable "station_altitude" not found or not defined correctly in the cloudmask NetCDF file

243 Variable "automatic_cloud_mask" or "manual_cloud_mask" not found or not defined correctly in the cloudmask NetCDF file

244 No cloud mask found for this measurement id. Please be sure the cloudmaks module ran succesfully

245 Global attribute "measurement_ID" not found or not defined correctly in the cloudmask NetCDF file

246 Cloud mask data not consistent with raw lidar data!

247 Unknown distribution type for MonteCarlo simulation

248 Polarization calibration: found inconsistency between wavelengths to calibrate and calibration wavelengths

249 Error in getting model data from CloudNet

250 It seems your station is not enabled to get model data from CloudNet. So please set Molecular_Calc=0 or 1 in your raw input file.

251 CloudNet model data have been downloaded succesfully but it seems it contains invalid height value (null or negative values).

254 General error (most probably segmentation fault)

255 Timeout

CHAPTER 5

Lidar retrievals

5.1 1. Algorithm Theoretical Basis

The European Aerosol Research Lidar Network, EARLINET, was founded in 2000 as a research project for establishing a quantitative, comprehensive, and statistically significant database for the horizontal, vertical, and temporal distribution of aerosols on a continental scale.

ACTRIS/EARLINET stations are typically able to retrieve aerosol optical properties, such as extinction and backscatter coefficients, lidar ratio, optical depth, and the Angstrom exponent if Raman lidar signals are available. In cases when only elastic lidar signals are used, backscatter and a backscatter-related Angstrom exponent are derived.

The calculation of all these quantities is done after the cloud masking procedure and it is therefore attributed exclusively to aerosol particles.

5.1.1 1.1 Physical meaning of the retrieved properties

When laser radiation with power P_L at wavelength λ_L is sent into the atmosphere, part of the radiation is backscattered. The optical power $P(\lambda, \lambda_L, z)$ of the backscattered radiation received from the distance z at wavelength λ depends on atmospheric composition through two parameters: the backscattering coefficient and the extinction coefficient, and is described by the lidar equation:

$$P(\lambda, \lambda_L, z) \sim \frac{P_L}{z^2}\beta(\lambda, \lambda_L, z) \exp\left[-\int_0^z \alpha(\lambda, \xi) d\xi\right] \exp\left[-\int_0^z \alpha(\lambda_L, \xi) d\xi\right] \tag{5.1}$$

The backscattering coefficient β is the fraction of incident radiation backscattered for unitary solid angle and for unitary length [m-1sr-1]. It depends on the kind of scattering process and on both emission (λ_L) and detection (λ) wavelength.

It is due to contributions of both molecules (m) and particles (p) of atmosphere:

$$\beta(\lambda, \lambda_L, z) = \beta_m(\lambda, \lambda_L, z) + \beta_p(\lambda, \lambda_L, z) \tag{5.2}$$

The extinction coefficient is defined as the energy flux reduction per unitary path [m-1].

It gives a measurement of the energy loss of the laser beam in the atmosphere.

It is due to contributions of both molecules (m) and particles (p) of atmosphere deriving from both the scattering (s) and absorption (a) processes:

$$\alpha(\lambda, z) = \alpha_{m,a}(\lambda, z) + \alpha_{m,s}(\lambda, z) + \alpha_{p,a}(\lambda, z) + \alpha_{p,s}(\lambda, z) \tag{5.3}$$

The extinction coefficient integrated over a spatial path provides the optical depth:

$$\tau(\lambda, z) = \int_0^z \alpha(\lambda, \xi) d\xi \tag{5.4}$$

The Raman configuration allows for the retrieval of the range-resolved particle lidar ratio. The lidar ratio is defined as the ration between the particle extinction coefficient and the particle backscatter coefficient:

$$S(\lambda, z) = \frac{\alpha(\lambda, z)}{\beta(\lambda, z)} \tag{5.5}$$

It is a parameter strongly related to the microphysical properties of the aerosols: shape, size distribution, chemical composition, relative humidity. Unlike α and β, S doesn't depend on atmospheric aerosol load, but only on aerosol type.

The combination of the particle extinction at different wavelengths allows for the calculation of the Angstrom exponent:

$$k_\alpha(\lambda_1, \lambda_2, z) = \frac{\ln\left[\frac{\alpha(\lambda_1, z)}{\alpha(\lambda_2, z)}\right]}{\ln\left[\frac{\lambda_2}{\lambda_1}\right]} \tag{5.6}$$

This quantity is size dependent assuming larger values for smaller particles and ranges between -1 for very big particles and 4 for molecules.

Similarly to the Angstrom exponent, the backscatter related Angstrom exponent can be calculated as:

$$k_\beta(\lambda_1, \lambda_2, z) = \frac{\ln\left[\frac{\beta(\lambda_1, z)}{\beta(\lambda_2, z)}\right]}{\ln\left[\frac{\lambda_2}{\lambda_1}\right]} \tag{5.7}$$

As for the Angstrom exponent this quantity is size dependent assuming larger values for smaller particles. However, it has to be noted that it is even more sensitive than Angstrom exponent to the size of the particles, because the backscatter itself is more size-related than the extinction coefficient.

5.1.2 1.2 Basic concepts for the retrieval of aerosol optical properties

ACTRIS/EARLINET is mainly based on Raman lidar stations, i.e. lidars equipped with elastic channel (detection channel at the same wavelength of transmitted laser beam) and an additional channel for detecting the N2 Raman-shifted signal. This additional channel allows the direct measurement of the aerosol extinction (Ansmann et al., 1990). This means having the capability of independent retrieval of extinction and backscatter coefficient in good signal-to-noise ratio conditions, using the retrieved extinction in the elastic lidar equation reported above. Whenever this is not possible an assumption about the relationship between extinction and backscatter is needed for solving the lidar equation affecting the overall uncertainty of the aerosol backscatter coefficient. Within ACTRIS/EARLINET, aerosol extinction profiles are reported only when the Raman channel capability is used and therefore only with direct assessed measurement of the extinction coefficient profile.

Solving the N2 Raman lidar equation involves a derivative respect to the range of the logarithm of the signal. This procedure is complex from mathematical point of view and needs for specific smoothing approaches. Within EARLINET many efforts have been done for comparing the different suitable procedures (Pappalardo et al., 2004): the linear fit has been identified as the most appropriate one. Two options are available the weighted and not weighted linear fit.

For what concerns the aerosol backscatter coefficient profiles, the SCC can provide aerosol products in a flexible way choosing from a set of possible pre-defined analysis procedures: it enables the retrieval of particle backscatter coefficients with the elastic technique by using both the Klett method (Klett, 1981; Fernald, 1984) and the iterative algorithm (Di Girolamo et al., 1995), but also the computation of particle backscatter coefficient profiles after the Raman method (Ansmann et al., 1992).

Statistical errors are calculated starting from the statistical errors affecting the lidar detected signals: the statistical errors affecting the optical properties can be calculated using the Monte Carlo or error propagation law. The provided errors do not include the uncertainties related to the assumption made in the retrieval algorithms like: the uncertainty to the atmospheric molecular profile, the wavelength dependence of the extinction, the absence of aerosol in the backscatter calibration range and more relevant the lidar ratio values assumption in the elastic backscatter method and the calibration of the depolarization channels. The quantification of the resulting overall error is still under investigation and object of studies within ACTRIS/EARLINET. The current approach is to reduce as much as possible such errors improving the quality assurance procedures together with the calibration centre and selecting the best possible approaches (e.g. intensifying the scheduling of the depolarization calibration procedures).

CHAPTER 6

Depolarization

6.1 1. Particle Linear Depolarization Ratio Implementation

The most important improvement included in the SCC v4.0 is the implementation of a new optical product which is the particle linear depolarization ratio.

Important: If your lidar system is not equipped with any polarization channels **NO** changes are required. In this case, the SCC v4.0 should work using the same input files and the same database configurations you have used with the SCC v3.11. Anyway as in the SCC v4.0 several bugs have been fixed, it is recommended to re-run all the measurement IDs you have submitted. For doing that you just need to reprocess all your data without the need to submit raw data files already uploaded on the server.

6.1.1 1.1 Background

The calculation of the volume linear depolarization ratio profile (*VLDR*) and particle linear depolarization ratio profile (*PLDR*) needs two different steps:

1. the calibration of the polarization sensitive lidar channels;
2. the calculation of the *VLDR* or *PLDR* itself.

The SCC allows the user to make both the above points. In particular the calibration step is made by a completely new module called **ELDEC** (Earlinet Lidar Depolarization Calibrator) which computes the *apparent calibration factor* η^* out of the pre-processed data provided by the standard **ELPP** (Earlinet Lidar Pre-Processor) module and it records it in the SCC database (SCC_DB). Once logged into the SCC_DB this factor can be used whenever it is necessary.

The raw lidar calibration measurements should be put in a NetCDF file which has the same structure as the "standard" raw SCC NetCDF input file (for more details see sections 2 and 3.2).

New signal types have been introduced to take into account special channel configurations used for calibration purposes.

Moreover new product types for both calibration and *PLDR* calculation have been defined. As, in principle, it is possible to calculate the *PLDR* only when the aerosol backscatter coefficient profile is available the following new products have been defined:

1. *Linear polarization calibration (factor η) (product_type_id=6);*
2. *Raman backscatter and linear depolarization ratio (product_type_id=7);*
3. *Elastic backscatter and linear depolarization ratio (product_type_id=8).*

The first product in the above list is used only for calibration while the other two are used for the calculation of *PLDR*. Basically, in most of the cases, the products 2 and 3 are equivalent to the corresponding backscatter product types with the exception that also the following new variables are available:

```
double VolumeDepol(Length) ;
double ErrorVolumeDepol(Length) ;
      ErrorVolumeDepol:long_name = "absolute error of VolumeDepol" ;
double ParticleDepol(Length) ;
double ErrorParticleDepol(Length) ;
      ErrorParticleDepol:long_name = "absolute error of ParticleDepol" ;
```

6.1.2 1.2 Polarization calibration

An important point is the definition of reliable *PLDR* calibration procedures. Within EARLINET the following calibration procedures are currently used:

a) Rayleigh calibration;

b) +45 calibration method, or Δ90 calibration method (made by +45 and -45 measurements);

c) 3 signals (total, cross and parallel).

It is well known that method a) could produce easily large errors on *PLDR* which cannot be controlled. For this reason only the methods b) and c) can be used to provide reliable polarization calibrations and so only those methods will be implemented in the SCC.

For what it concerns the method c) it, basically, requires to solve the equation:

$$\alpha_s P_s + \alpha_p P_p = P \tag{6.1}$$

in two different atmospheric layers with considerably different *VLDR*. So to calibrate in this way the implementation of automatic layer identification in the SCC is required. As at moment this feature is not yet available within the SCC **ONLY** the method b) is considered.

6.1.3 1.3 SCC procedure to calculate the PLDRP

According to what mentioned before the SCC calculates the *PLDR* through the following steps:

1. The user needs to create a new system configuration in the SCC_DB including only lidar channels used for the calibration. One (or more) *Linear polarization calibration (product_type_id=6)* product should be associated to this new configuration (see section 3.2 for more details);
2. This new system configuration should contain only the polarization channels in the configuration used for the calibration (for example rotated in the polarization plane of +45 degrees). A channel in calibration measurement configuration should have a **DIFFERENT** channel ID from the channel ID corresponding to the same channel in standard measurement configuration. For example, if a system has two polarization channels which in standard measurement configuration correspond to the channel ID=1 and 2 respectively, the same physical channels under calibration measurement configuration should correspond to different channel IDs (let's say ID=3 and 4 for the

+45 degrees polarization rotated channels and ID=5 and 6 for the -45 degrees polarization rotated ones in case D90 calibration method is used). Moreover, the polarization channels should be labeled correctly using the new signal types available (*+45elPT, +45elPR, -45elPT, -45elPR, +45elPTnr, +45elPTfr, +45elPRnr, +45elPRfr, -45elPTnr, -45elPTfr, -45elPRnr, -45elPRfr).* For more details see section 3.2;

3. In SCC v4.0 the polarization channels are **NOT** labeled on the base of their polarization state (as it was done in the SCC v3.11) but **ALWAYS** as transmitted and reflected channels. So the channels that in SCC v3.11 were labeled as *elCP, elCPnr, elCPfr, elPP, elPPnr elPPfr* will be labeled in SCC v4.0 as *elPR, elPRnr elPRfr elPT, elPTnr elPTfr* where the letter *T* stands from transmitted and the letter *R* for reflected.

Warning:

In switching from the SCC v3.11 to SCC v4.0 the following modifications have been made on ALL channels of ALL registered

elPP→elPR

elCP→elPT

elPPnr→elPRnr

elPPfr→ elPRfr

elCPnr→ elPTnr

elCPfr→ elPTfr

Please be sure these modifications reflect to your actual lidar setup(cross channels are transmitted and parallel channels are reflected);

4. The user needs to submit a file (same format as raw SCC input file) containing the raw data for the lidar channels defined at the point 1 (see section 3.2 for more details);

5. The file at point 4 is pre-processed by **ELPP** module which applies the standard pre-processing procedures applied to "standard" lidar data;

6. The pre-processed files are then processed by the new modules **ELDEC** which calculates η^* *the apparent calibration factor* and logs it into the SCC_DB;

7. The user needs to create a new system configuration in the SCC_DB (which should be different from the one used for the calibration) and associate it the new product *Raman backscatter and linear depolarization ratio product_type_id=7)* or *Elastic backscatter and linear depolarization ratio (product_type_id=8).* Alternatively the calculation of those products can be added to an already existing lidar configuration as long as it is different from the calibration one;

8. The product defined at point 7 should be linked to the product containing the polarization calibration (defined at point 1) in a way that the *apparent calibration factor* can be selected from the SCC_DB (see section 3.3 and in particular figure 3.4);

9. The user needs to submit another SCC raw data file containing the "standard" measurements;

10. Finally **ELPP** and **ELDA** will produce a b-file containing backscatter coefficient profile and *PLDR*. In particular this calculation is made in two different steps: from the pre-processed lidar polarization signals, and taking into account the *apparent calibration factor* and the *calibration factor correction K* (defined as option of *Linear polarization calibration* product*)* written into the SCC_DB, an "apparent" *VLDR* δ^* is calculated. Even if δ^* is a calibrated quantity it can be still affected by possible systematic errors due to not perfect optics or alignment of the system;

11. To take into account these errors a corrected *VLDR* (δ) is calculated using the *polarization cross-talk correction parameters G* and *H* calculated on the base of Müller matrix formalism. These cross-talk correction parameters (*G* and *H*) are stored in the SCC_DB for each lidar channels (see section 3.1 in particular figure 3.2). Finally

the *PLDR* is calculated using the backscatter coefficient profile and the molecular LDRP calculated by ELPP considering the center wavelength and bandwidth of the channels interference filter.

The *apparent calibration factor* η^* is calculated by the **ELDEC** module as the geometrical mean of the ratio of the +/-45 degrees reflected to the +/- 45 degrees transmitted signals within an altitude calibration range defined by the users in the raw data input files.

In case of +45 calibration method η^* is calculated by:

$$\eta^* = \frac{I_R}{I_T}(+45) \tag{6.2}$$

While in case of $\Delta 90$ calibration method:

$$\eta^* = \sqrt{\frac{I_R}{I_T}(+45)\frac{I_R}{I_T}(-45)} \tag{6.3}$$

ELDA module calculates the “apparent” *VLDR*:

$$\delta^* = \frac{K}{\eta^*} \cdot \frac{I_R}{I_T} \tag{6.4}$$

the *VLDR*

$$\delta = \frac{\delta^*(G_T + H_T) - (G_R + H_R)}{(G_R - H_R) - \delta^*(G_T - H_T)} \tag{6.5}$$

and the *PLDR*

$$\delta_\alpha = \frac{(1 + \delta_m)\delta R - (1 + \delta)\delta_m}{(1 + \delta_m)R - (1 + \delta)} \tag{6.6}$$

where:

- η^* is the *apparent calibration factor* calculated by **ELDEC**
- *K* is the *calibration factor correction* defined as polarization product option
- I_T and I_R are the transmitted and the reflected signals in the polarization detection set-up
- $G_{T,R}$ and $H_{T,R}$ are *polarization cross-talk correction parameters* for the transmitted and reflected signals used to correct for systematic errors. Both these factors are defined in the SCC_DB for each lidar channel.
- δ_m is the molecular linear depolarization ratio calculated by ELPP
- *R* is the backscatter ratio

Please note once again that the polarization channels are described in terms of transmitted and reflected signals. This means that according to different lidar instrumental configurations, the transmitted or the reflected channel can contain total, perpendicular or parallel polarized signals.

In order to retrieve the backscatter profile the total signal must be obtained combining the transmitted and reflected polarized signals. The following formula is used:

$$I_{total} \propto \frac{\frac{\eta^*}{K} H_R I_T - H_T I_R}{H_R G_T - H_T G_R} \tag{6.7}$$

The formulas above are general and can be adapted to all possible polarization lidar configurations selecting the right polarization cross-talk correction parameters (see Table 1.1).

Let’s suppose, for example, we have the perpendicular polarized lidar signal on the transmitted channel and the parallel polarized on reflected channel. For an ideal system (no diattenuation and cross-talk) we have:

$$G_T = 1, \qquad H_T = -1, \qquad G_R = 1, \qquad H_R = 1$$

If, on the other hand, we have the perpendicular polarized lidar signal on reflected channel and the total polarized on the transmitted for and ideal system we have:

$$G_T = 1, \qquad H_T = 0, \qquad G_R = 1, \qquad H_R = -1$$

Table 1.1: Polarization cross-talk correction parameters for ideal systems

Laser polarization	Detected in lidar channel			
	Transmitted		Reflected	
	G_T	H_T	G_R	H_R
total	1	0	1	0
parallel	1	1	1	1
cross	1	-1	1	-1

The *apparent calibration factor* (η^*), *the calibration factor correction* (K) and the *polarization cross-talk correction parameters* are stored by **ELPP** module in the intermediate NetCDF files using the following variables:

- `Polarization_Channel_Gain_Factor` (*apparent calibration factor* - η^*)
- `Polarization_Channel_Gain_Factor_Correction` (*calib. factor corr. – K*)
- `G_T`
- `H_T`
- `G_R`
- `H_R`

Finally new usecases have been defined to take into account all the possible lidar configurations. The details on that are provided as a separate file.

6.2 2. Changes of the SCC input format

The following minor changes have been applied to raw SCC data format:

1. The optional variable *ID_Range* has been **REMOVED**;
2. The **OPTIONAL** variable `int Signal_Type(channels)` has been added. The possible values are the same available in the SCC_DB:

   `0` → `elT`

   `1` → `elTnr`

   `2` → `elTfr`

   `3` → `vrRN2`

   `4` → `vrRN2nr`

   `5` → `vrRN2fr`

   `6` → `elPR`

   `7` → `elPT`

   `8` → `pRRlow`

   `9` → `pRRhigh`

   `10` → `elPRnr`

11 → `elPRfr`

12 → `elPTnr`

13 → `elPTfr`

14 → `vrRH2O`

15 → `pRRhighnr`

16 → `pRRhighfr`

17 → `pRRlownr`

18 → `pRRlowfr`

19 → `vrRH2Onr`

20 → `vrRH2Ofr`

21 → `elTunr`

22 → `+45elPT`

23 → `+45elPR`

24 → `-45elPT`

25 → `-45elPR`

26 → `+45elPTnr`

27 → `+45elPTfr`

28 → `+45elPRnr`

29 → `+45elPRfr`

30 → `-45elPTnr`

31 → `-45elPTfr`

32 → `-45elPRnr`

33 → `-45elPRfr`

> **Warning:** This variable is found in the SCC input file the corresponding settings in the SCC database will be **OVERWRITTEN**. Unless you don't have any valid reason to overwrite the database value this variable should not be used.

3. The variables:

```
double Pol_Calib_Range_Min(channels)
double Pol_Calib_Range_Max(channels)
```

have been added. Both these variable are **MANDATORY** for any calibration raw dataset. These variable should be included only the polarization calibration measurements and should specify the altitude range (meters) in which the polarization calibration should be made. For more details see section 3.3;

4. The variable `Depolarization_Factor` has been **REMOVED**.

The SCC v3.11 used this variable to get polarization calibration factor for the calculation of the total signal out of cross and parallels ones. As the SCC v4.0 is able to calculate the same parameter by itself, the use of this variable is *NOT* possible anymore. The recommended way to get a valid and quality assured depolarization calibration factor is to submit to the SCC v4.0 a polarization calibration dataset and let the SCC to calculate such factor.

To make this change more smooth and to provide the users with the possibility to continue to analyze their data with the SCC v4.0 even if a calibration dataset has not been submitted yet, it will be possible for a **LIMITED** period of time to submit the calibration constant via the SCC web interface. The SCC will keep track of the used calibration method (automatic or manual).

> **Warning:** After this transition period **ONLY** automatic calibration will be allowed!

5. The new **OPTIONAL** variable:

   ```
   string channel_string_ID(channels)
   ```

   has been introduced.

   Starting from SCC v4.0 the lidar channel can be identified not only by using integers (as it happened until SCC v3.11) but also by using strings.

   The procedure implemented in the SCC v4.0 to recognize the lidar channel within the raw lidar data is fully backward compatible (old format files are accepted as they are by SCC v4.0).

> **Warning:** Please note that the definition of the new string variable requires netCDF-4 format! The type *string* is not supported in netCDF-3 format!

6.3 3. Real Example

This section describes all the practical steps the users need to follow to switch from SCC v3.11 to new SCC v4.0.

IMPORTANT If your lidar system is not equipped with any polarization channels **NO** changes are required. In this case, the SCC v4.0 should work using the same input files and the same database configurations you have used with the SCC v3.11. Anyway as in the SCC v4.0 several bugs have been fixed,it is recommended to re-run all the measurement IDs you have submitted. For doing that you just need to reprocess all your data without the need to submit raw data files already uploaded on the server.

The practical example reported below describes the modifications required to use the SCC v4.0 for lidar systems equipped with polarization channels. Lidar systems not equipped with polarization channels do not require any modification to switch to SCC v4.0.

6.3.1 3.1 Modification of polarization channel parameters

In what it follows it is assumed you already have registered one or more lidar configurations in the SCC database and that such configurations have been already used to produce optical products (aerosol extinction and/or backscatter coefficients) by means of the SCC v3.11.

Let's assume your 3+2 system is registered in the SCC database and the settings used by the SCC v3.11 are the ones summarized in table 3.1.

Table 3.1 Example of configuration in SCC v3.11

Channel Name	Channel ID	Channel Type	nighttime	daytime
355	1	elT	x	x
387	2	vrRN2	x	
532 cross	3	elCP	x	x
532 parallel	4	elPP	x	x
607	5	vrRN2	x	
1064	6	elT	x	x

We assume there are 2 system configurations called "nighttime" and "daytime". The nighttime configuration contains all the available lidar channels (in order to calculate, for example, the aerosol extinction at 355 and 532nm and the aerosol backscatter at 355, 532 and 1064nm) while in daytime conditions only elastic channels are used (only elastic backscatter coefficients are generated).

To make these settings working with SCC v4.0 it is needed to modify :underline:ONLY‘ the products properties involving the polarization channels (532 cross and parallel). All the products not involving the polarization channels **DO NOT** need any modification and should work in the SCC v4.0 exactly as they did in SCC v3.11. In the example above the aerosol extinction and backscatter coefficient at 355nm, the extinction at 532nm as well as the backscatter coefficient at 1064nm do not required any modification. Let's focus on the modifications needed for the calculation of backscatter at 532nm.

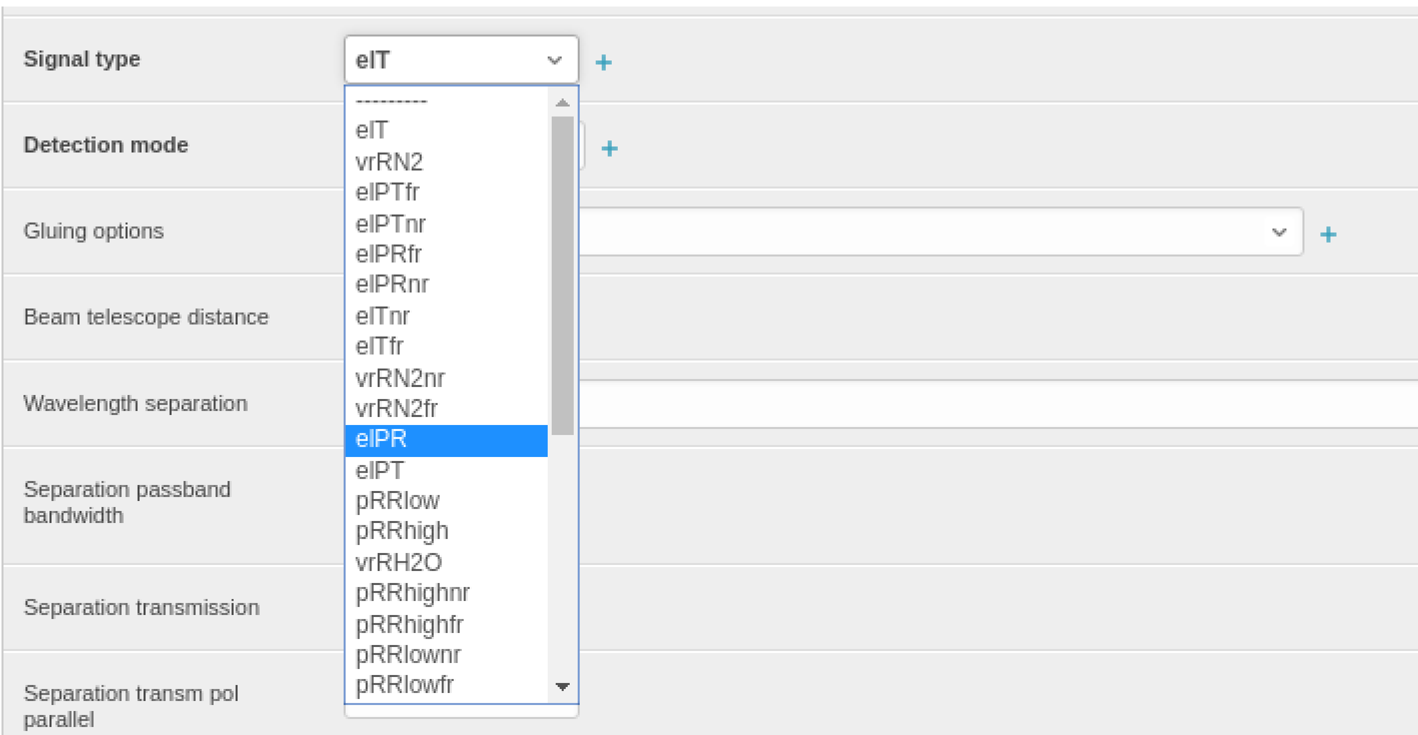


Fig. 1: **Figure 3.1**: How to select signal types

The first modification concerns the settings of the channel type for the 532 cross and 532 parallel polarization channels. Starting from SCC v4.0 polarization channels are identified as transmitted and reflected polarization channels and not on the base of their polarization state. So if we suppose that the cross polarized channel is transmitted by a polarizer beam splitter cube, and the parallel is reflected, the value reported in table 3.1 should be modified as they appear in table 3.2. So using the SCC web interface, the signal type of the 532 cross channel should be changed from `elCP` to `elPT` and in the same way the 532 parallel channel should be changed from `elPP` to `elPR` (see figure 3.1).

Table 3.2: The same of table 3.1 but with new channel types introduced in SCC v4.0

Channel Name	Channel ID	Channel Type	nighttime	daytime
355	1	elT	x	x
387	2	vrRN2	x	
532 cross	3	**elPT**	x	x
532 parallel	4	**elPR**	x	x
607	5	vrRN2	x	
1064	6	elT	x	x

The other change about the polarization channels required to run the SCC v4.0 is the definition of the polarization crosstalk parameters for all the polarization channels available. Such parameters can be defined for each polarization channel using the SCC web interface (see figure 3.2). In particular among the channel parameters there is a new tab called *Polarization crosstalk parameters* where it is possible to insert the values from for the parameters *G* and *H* and the corresponding statistical and systematic errors if available. In case you have measured *G* and *H* for your polarization channels please insert the corresponding values there. Otherwise you can insert the ideal values as reported in table 1.1.

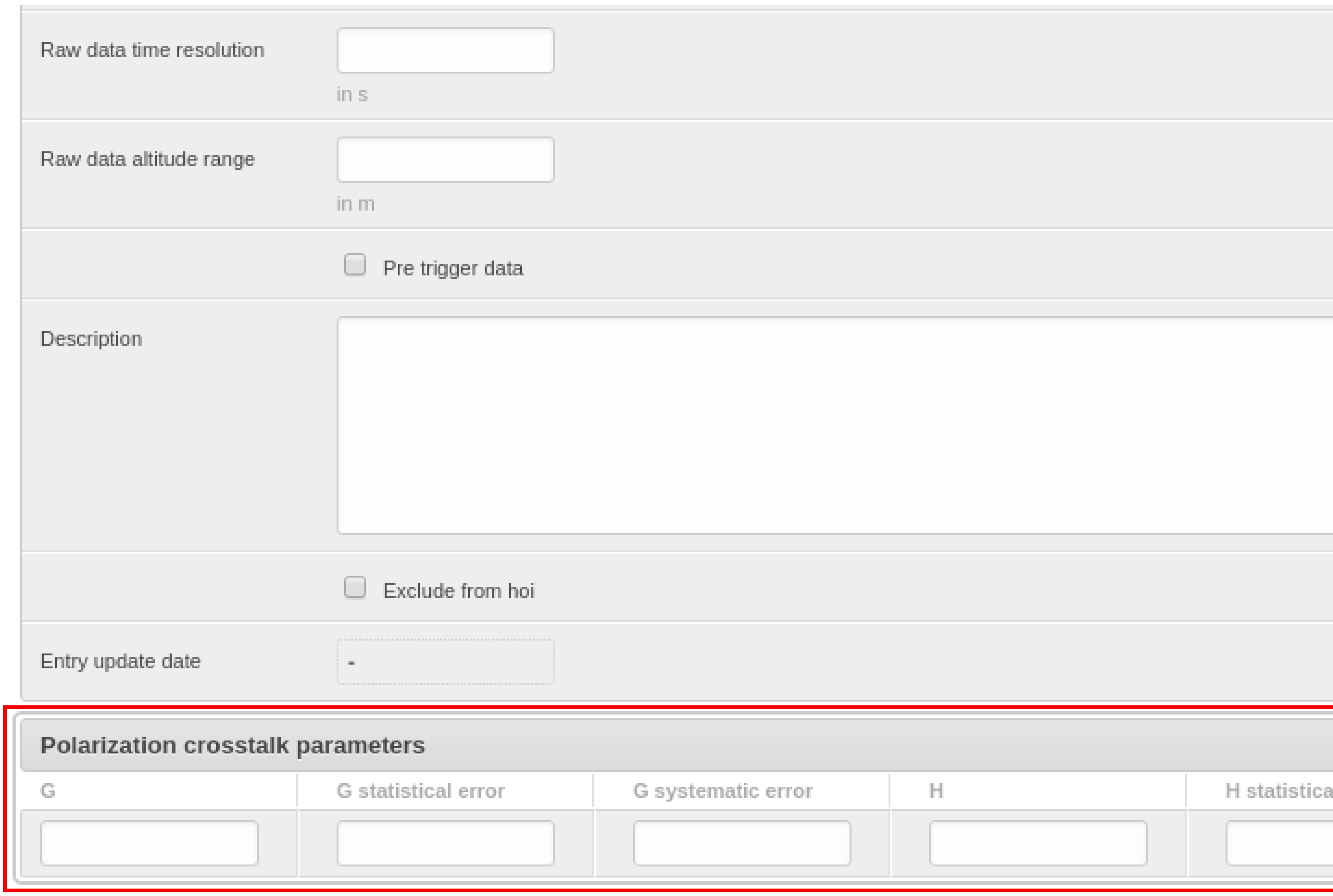


Fig. 2: **Figure 3.2:** Polarization crosstalk parameters tab in channel properties (SCC v4.0).

6.3.2 3.2 Definition of new calibration configuration and product

In this section we will see how to set the polarization calibration parameters: the calibration constant (called η^* in section 1.3) and the correction to calibration constant (called K in section 1.3). In order to provide such parameters you need to define a new system configuration to be used **ONLY** for calibration purposes. Such new configuration should include the polarization channels in the measurement configuration used for the calibration. Let's suppose we want to use the $\Delta 90$ calibration method.

In this case we need to define a new configuration (called for example "depol_calibration") as reported in the table 3.3. As you can see the configuration "depol_calibration" includes 4 "new" channels. Actually the channels "532 cross +45 degrees" (channel ID=10) and "532 cross -45 degrees" (channel ID=12) refer to the same physical channel "532 cross" reported with channel ID=3 in table 3.2. Anyway we need to define two new channel IDs to identify the "532 cross" channel in the two polarization rotated configurations (+45 and -45 degrees) needed to apply the D90 calibration method. The same is true for the "532 parallel" channel. The polarization rotated channels should be labeled with the corresponding signal type as reported in table 3.3 (see figure 3.1).

Table 3.3: Polarization calibration configurations assuming $\Delta 90$ calibration method

Channel Name	Channel ID	Channel Type	depol_calibration
532 cross +45 degrees	10	+45elPT	x
532 parallel +45 degrees	11	+45elPR	x
532 cross -45 degrees	12	-45elPT	x
532 parallel -45 degrees	13	-45elPR	x

Finally we should add to the configuration "depol_calibration" a product "*Linear polarization calibration"* to be used for the calibration. According to the example given above and to the usecase document attached we should use an usecase=4 for this example.

Other "*Linear polarization calibration"* options to be specified are reported in figure 3.3. The most important factor you should insert here is the *Pol calibration correction factor* (K). The ideal value for this parameter is 1. Anyway if you have measured the parameter K please fill in the measured value and the corresponding measurement errors.

As you can see it is possible to fill in only the K correction factor and not the calibration constant η^*.

Actually for a **LIMITED** period of time it will be possible to fill in also the constant η^* using a temporary option shown in figure 3.4. This has been done to provide the users with the possibility to continue to use the SCC even if an automatic calibration made by the SCC was not submitted yet. Anyway after a transition period it will be **NOT** possible to provide calibration constant using this procedure and the parameter η^* can be calculated **ONLY** by the SCC as result of the submission of a proper calibration raw input dataset. The format of this input file is the same as the standard SCC input file. The only difference is that is should contain calibration measurements instead of standard measurements. Following our example, such file should contain the measurement performed at +45 and -45 degrees at 532nm. Also the channel IDs in the file should reflect the ones reported in table 3.3.

Moreover this raw input file has to contain the variables:

```
double Pol_Calib_Range_Min(channels)
double Pol_Calib_Range_Max(channels)
```

where to specify the altitude ranges in meters in which the polarization calibration should be done.

According to the table 3.3 this file should be something similar to:

```
dimensions:
   channels = 4 ;
   nb_of_time_scales = 1 ;
   points = 16380 ;
   scan_angles = 1 ;
```

(continues on next page)

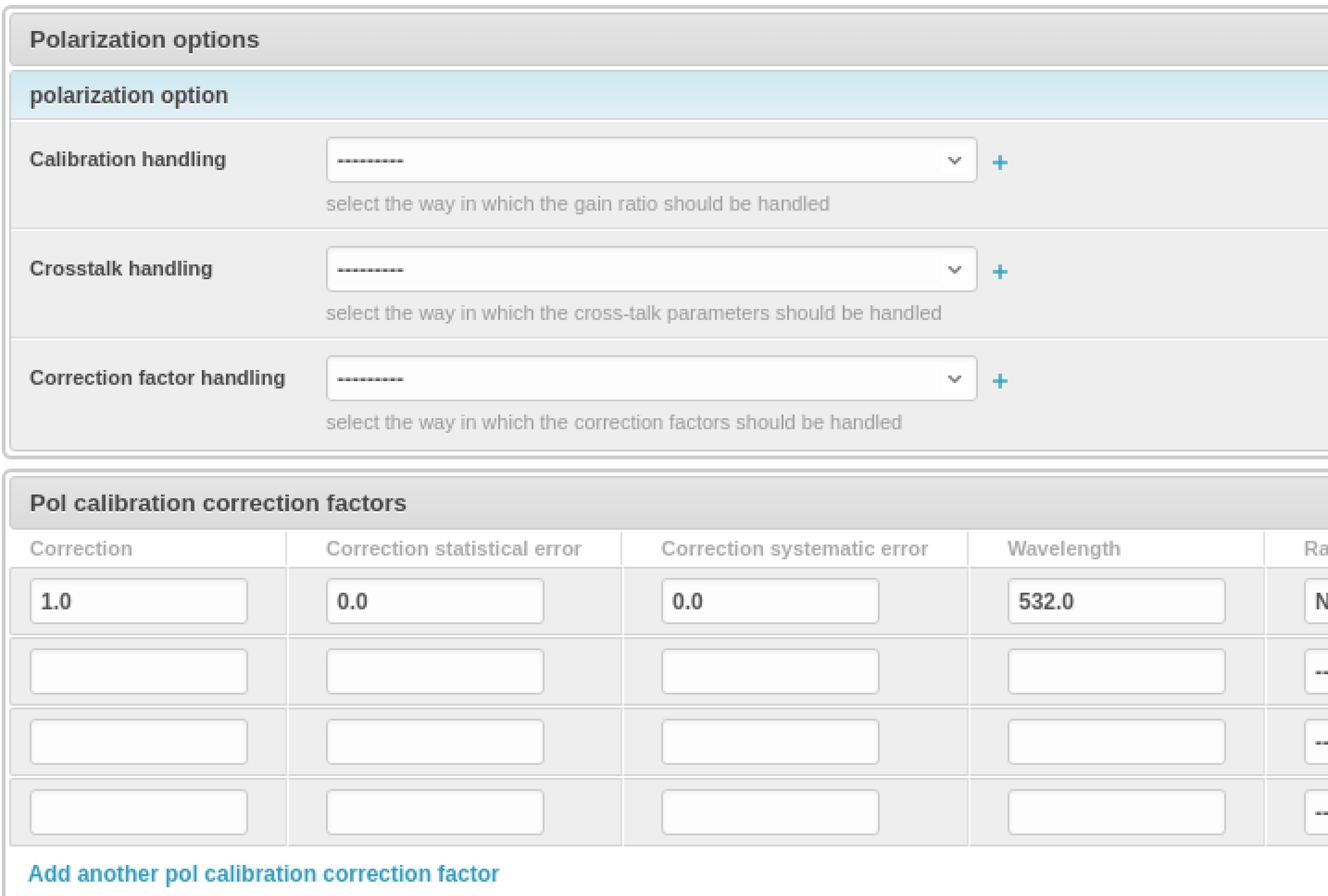
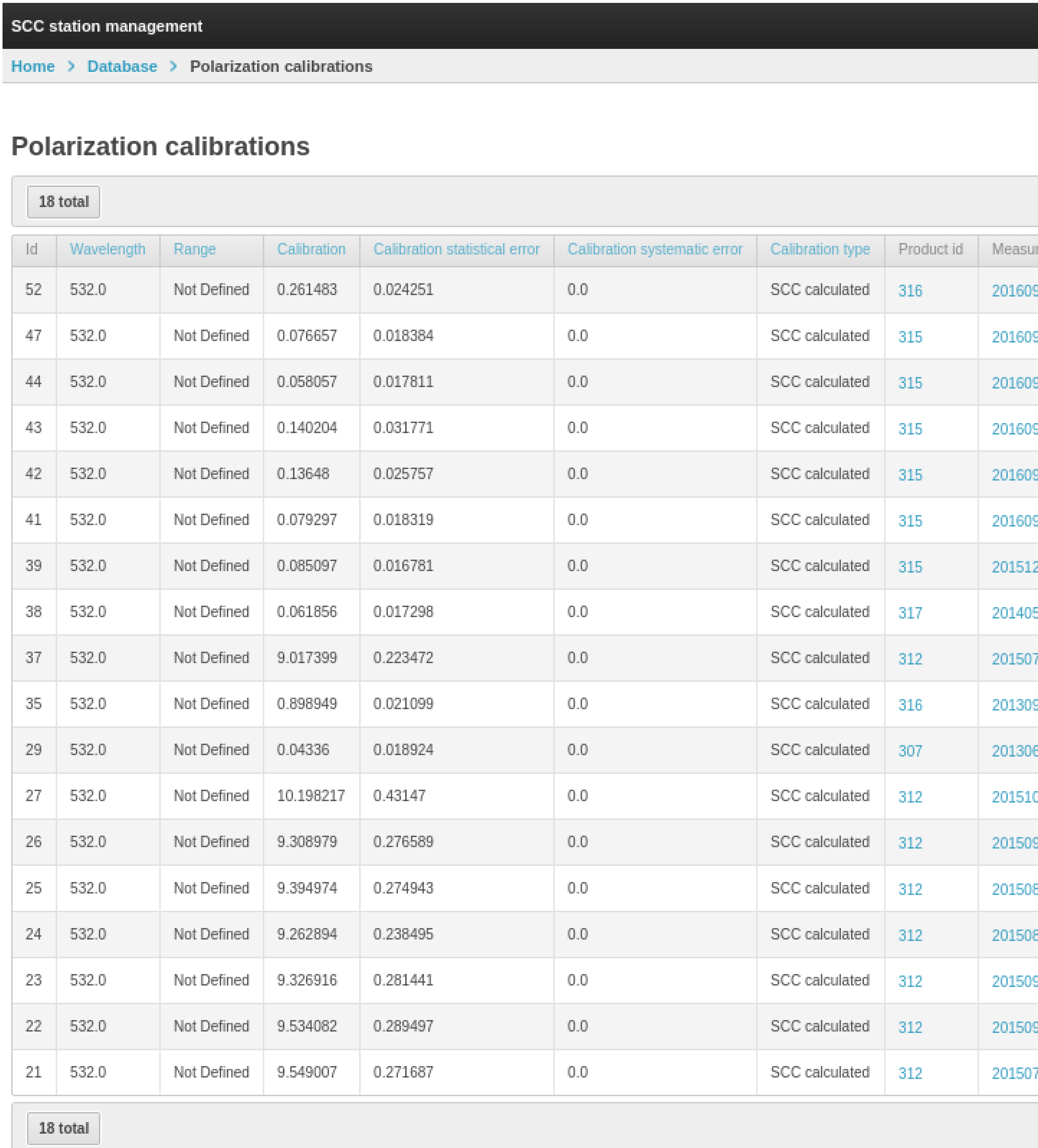


Fig. 3: **Figure 3.3:** Options for *Linear polarization calibration product*.



Id	Wavelength	Range	Calibration	Calibration statistical error	Calibration systematic error	Calibration type	Product id	Measur
52	532.0	Not Defined	0.261483	0.024251	0.0	SCC calculated	316	201609
47	532.0	Not Defined	0.076657	0.018384	0.0	SCC calculated	315	201609
44	532.0	Not Defined	0.058057	0.017811	0.0	SCC calculated	315	201609
43	532.0	Not Defined	0.140204	0.031771	0.0	SCC calculated	315	201609
42	532.0	Not Defined	0.13648	0.025757	0.0	SCC calculated	315	201609
41	532.0	Not Defined	0.079297	0.018319	0.0	SCC calculated	315	201609
39	532.0	Not Defined	0.085097	0.016781	0.0	SCC calculated	315	201512
38	532.0	Not Defined	0.061856	0.017298	0.0	SCC calculated	317	201405
37	532.0	Not Defined	9.017399	0.223472	0.0	SCC calculated	312	201507
35	532.0	Not Defined	0.898949	0.021099	0.0	SCC calculated	316	201309
29	532.0	Not Defined	0.04336	0.018924	0.0	SCC calculated	307	201306
27	532.0	Not Defined	10.198217	0.43147	0.0	SCC calculated	312	201510
26	532.0	Not Defined	9.308979	0.276589	0.0	SCC calculated	312	201509
25	532.0	Not Defined	9.394974	0.274943	0.0	SCC calculated	312	201508
24	532.0	Not Defined	9.262894	0.238495	0.0	SCC calculated	312	201508
23	532.0	Not Defined	9.326916	0.281441	0.0	SCC calculated	312	201509
22	532.0	Not Defined	9.534082	0.289497	0.0	SCC calculated	312	201509
21	532.0	Not Defined	9.549007	0.271687	0.0	SCC calculated	312	201507

Fig. 4: **Figure 3.4:** To provide polarization calibration (η^*) values manually just use the button "Add polarization calibration" in the upper-right corner. This option will be available only for a limited period of time. After that only SCC calculated calibration constants will be accepted.

(continued from previous page)

```
   time = UNLIMITED ; // (3 currently)
variables:
   int channel_ID(channels) ;
   double Background_Low(channels) ;
   double Background_High(channels) ;
   int id_timescale(channels) ;
   double Laser_Pointing_Angle(scan_angles) ;
   int Molecular_Calc ;
   int Laser_Pointing_Angle_of_Profiles(time, nb_of_time_scales) ;
   int Raw_Data_Start_Time(time, nb_of_time_scales) ;
   int Raw_Data_Stop_Time(time, nb_of_time_scales) ;
   int Laser_Shots(time, channels) ;
   double Raw_Lidar_Data(time, channels, points) ;
   double Pressure_at_Lidar_Station ;
   double Temperature_at_Lidar_Station ;
   double Pol_Calib_Range_Min(channels) ;
   double Pol_Calib_Range_Max(channels) ;

// global attributes:
         :System = "mysystem" ;
         :Longitude_degrees_east = 15.723771 ;
         :RawData_Start_Time_UT = "220000" ;
         :RawData_Start_Date = "20130620" ;
         :Measurement_ID = "20130620po00" ;
         :Altitude_meter_asl = 760. ;
         :RawData_Stop_Time_UT = "230333" ;
         :Latitude_degrees_north = 40.601039 ;

data:
 channel_ID = 10, 11, 12, 13 ;

 Background_Low = 30000, 30000, 30000, 30000 ;

 Background_High = 50000, 50000, 50000, 50000 ;

 id_timescale = 0, 0, 0, 0 ;

 Laser_Pointing_Angle = 0 ;

 Molecular_Calc = 0 ;

 Laser_Pointing_Angle_of_Profiles =
  0,
  0,
  0 ;

 Raw_Data_Start_Time =
  0,
  300,
  600 ;

 Raw_Data_Stop_Time =
  210,
  510,
  810 ;

 Laser_Shots =
```

(continues on next page)

(continued from previous page)

```
 1200, 1200, 1200, 1200,
 1200, 1200, 1200, 1200,
 1200, 1200, 1200, 1200 ;

Pressure_at_Lidar_Station = 1010 ;

Temperature_at_Lidar_Station = 14 ;

Pol_Calib_Range_Min = 1000, 1000, 1000, 1000 ;

Pol_Calib_Range_Min = 2000, 2000, 2000, 2000 ;

Raw_Lidar_Data = ......;
```

The file above assume the following calibration measurements have been done:

1. First +45 degrees acquisition followed by a corresponding -45 degrees acquisition

   a. Measurement at +45 degrees

      Start Time: 20130620 22:00:00

      Stop Time: 20130620 22:01:00

      Shots: 1200

   b. Measurement at -45 degrees

      Start Time: 20130620 22:02:30

      Stop Time: 20130620 22:03:30

      Shots: 1200

2. Second +45 degrees acquisition followed by a corresponding -45 degrees acquisition

   a. Measurement at +45 degrees

      Start Time: 20130620 22:05:00

      Stop Time: 20130620 22:06:00

      Shots: 1200

   b. Measurement at -45 degrees

      Start Time: 20130620 22:07:30

      Stop Time: 20130620 22:08:30

      Shots: 1200

3. Third +45 degrees acquisition followed by a corresponding -45 degrees acquisition

   a. Measurement at +45 degrees

      Start Time: 20130620 22:10:00

      Stop Time: 20130620 22:11:00

      Shots: 1200

   b. Measurement at -45 degrees

Start Time: 20130620 22:12:30

Stop Time: 20130620 22:13:30

Shots: 1200

As you can see there are 3 cycles of consecutive measurements at +45 and -45 degrees. That way the dimension `time` is set to 3.

The first +/-45 degrees measurement starts at "20130620 22:00:00" (start time of the first +45 measurement) and stops at "20130620 22:03:30" (stop time of the fist -45 measurement). As a consequence, according to the values of the global attributes `RawData_Start_Date` and `RawData_Start_Time_UT` we have to set:

`Raw_Data_Start_Time[0]=0` (start of the first +45 measurement in seconds since `RawData_Start_Time_UT`)

`Raw_Data_Stop_Time[0]=210` (stop of the first -45 measurement in seconds since `RawData_Start_Time_UT`)

Following a similar procedure for the other 2 cycles we have:

`Raw_Data_Start_Time[1]=300` (start of the second +45 measurement in seconds since `RawData_Start_Time_UT`)

`Raw_Data_Stop_Time[1]=510` (stop of the second -45 measurement in seconds since `RawData_Start_Time_UT`)

`Raw_Data_Start_Time[2]=600` (start of the third +45 measurement in seconds since `RawData_Start_Time_UT`)

`Raw_Data_Stop_Time[2]=810` (stop of the third -45 measurement in seconds since `RawData_Start_Time_UT`)

Moreover, according to the order of the channels in the `channel_ID` variable, the `Raw_Lidar_Data` array should be filled as it follows:

`Raw_Lidar_Data[0][0][points]` → 1^{st} measured transmitted signal at +45 degrees

`Raw_Lidar_Data[0][1][points]` → 1^{st} measured reflected signal at +45 degrees

`Raw_Lidar_Data[0][2][points]` → 1^{st} measured transmitted signal at -45 degrees

`Raw_Lidar_Data[0][3][points]` → 1^{st} measured reflected signal at -45 degrees

`Raw_Lidar_Data[1][0][points]` → 2^{nd} measured transmitted signal at +45 degrees

`Raw_Lidar_Data[1][1][points]` → 2^{nd} measured reflected signal at +45 degrees

`Raw_Lidar_Data[1][2][points]` → 2^{nd} measured transmitted signal at -45 degrees

`Raw_Lidar_Data[1][3][points]` → 2^{nd} measured reflected signal at -45 degrees

`Raw_Lidar_Data[2][0][points]` → 3^{rd} measured transmitted signal at +45 degrees

`Raw_Lidar_Data[2][1][points]` → 3^{rd} measured reflected signal at +45 degrees

`Raw_Lidar_Data[2][2][points]` → 3^{rd} measured transmitted signal at -45 degrees

`Raw_Lidar_Data[2][3][points]` → 3^{rd} measured reflected signal at -45 degrees

Once this file has been created it needs to be submitted to the SCC and linked to the configuration "depol_calibration". The result of the SCC analysis on this file will be the calculation of the calibration constant h^* that will be logged into the SCC database and can be used to calibrate Raman/Elastic backscatter products (see section 3.3).

6.3.3 3.3 Definition of "Raman/Elastic backscatter and linear depolarization ratio"

In order to calculate the *PLDR* we need to modify the polarization related products linked to the "standard" measurement configurations (the configuration called "nighttime" and/or "daytime" in table 3.2).

Let's suppose we have defined the following products (defined already in SCC v3.11):

Table 3.4: Example of products configuration in SCC v3.11

Product Name	Product ID	Product Type	nighttime	daytime
Raman backscatter 355nm	1	Raman backscatter	x	
Extinction 387nm	2	Extinction	x	
Raman backscatter 532nm	3	Raman backscatter	x	
Extinction 532nm	4	Extinction	x	
Elastic backscatter 355nm	5	Elastic backscatter		x
Elastic backscatter 532nm	6	Elastic backscatter		x
Elastic backscatter 1064nm	7	Elastic backscatter	x	x

Product ID=1, 2, 4, 5, 7 do not need any modification as they do not involve polarization channels. The only product that need to be modified are the Product ID=3 and 6. To produce b532 files containing also *PLDR* we need to modify the "nighttime" and "daytime" configurations to include a product of type "Raman backscatter and linear depolarization ratio" or "Elastic bakscatter and linear depolarization ratio" respectively. So the configuration reported in table 3.4 should be changed to match what is included in table 3.5.

Table 3.5: The same of table 3.4 but with new product types introduced in SCC v4.0

Product Name	Product ID	Product Type	nighttime	daytime
Raman backscatter 355nm	1	Raman backscatter	x	
Extinction 387nm	2	Extinction	x	
Raman backscatter 532nm	10	**Raman backscatter and linear depolarization ratio**	x	
Extinction 532nm	4	Extinction	x	
Elastic backscatter 355nm	5	Elastic backscatter		x
Elastic backscatter 532nm	11	**Elastic backscatter and linear depolarization ratio**		x
Elastic backscatter 1064nm	7	Elastic backscatter	x	x

As you can see in table 3.5, the old product IDs=3 and 6 (present in table 3.4) have been replaced with the new product ID=10 and 11 to guarantee the calculation of *PLDR*.

It is important to set among the product options of the product ID=10 and 11 which calibration product we want to use for calibration (see section 3.2). This can be done using the SCC web interface setting the appropriate setting in the tab *Polarization calibration products* (see figure 3.4). According to the current example you should set here the calibration product defined in section 3.2.

Fig. 5: **Figure 3.5:** How to link a product to calibrate with a calibration product.

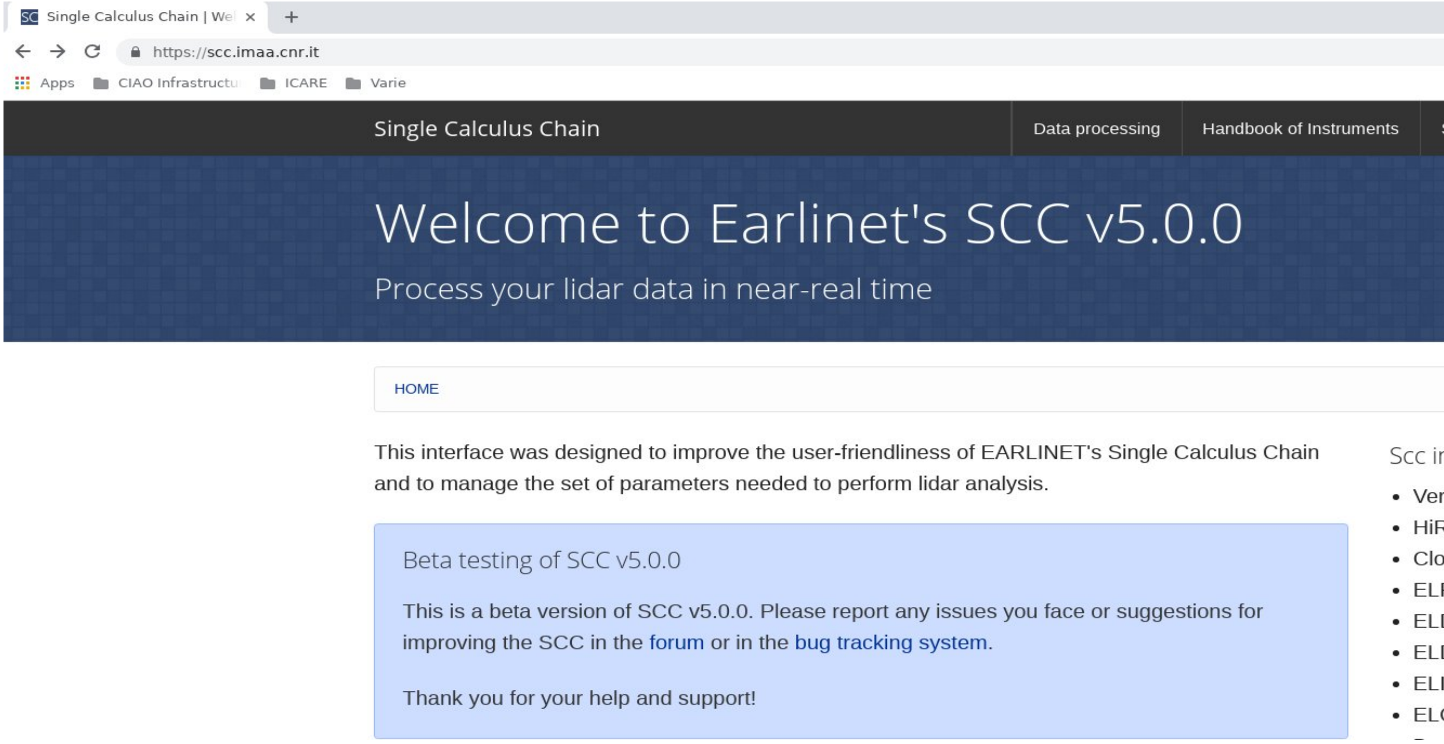
Warning: Please note that also *Raman/Elastic backscatter products* need to be linked to a calibration product because the calibration constant and the corresponding correction factor is needed to calculate the total signal out of the two polarization components even if the *PLDR* is not involved in the product calculation.

CHAPTER 7

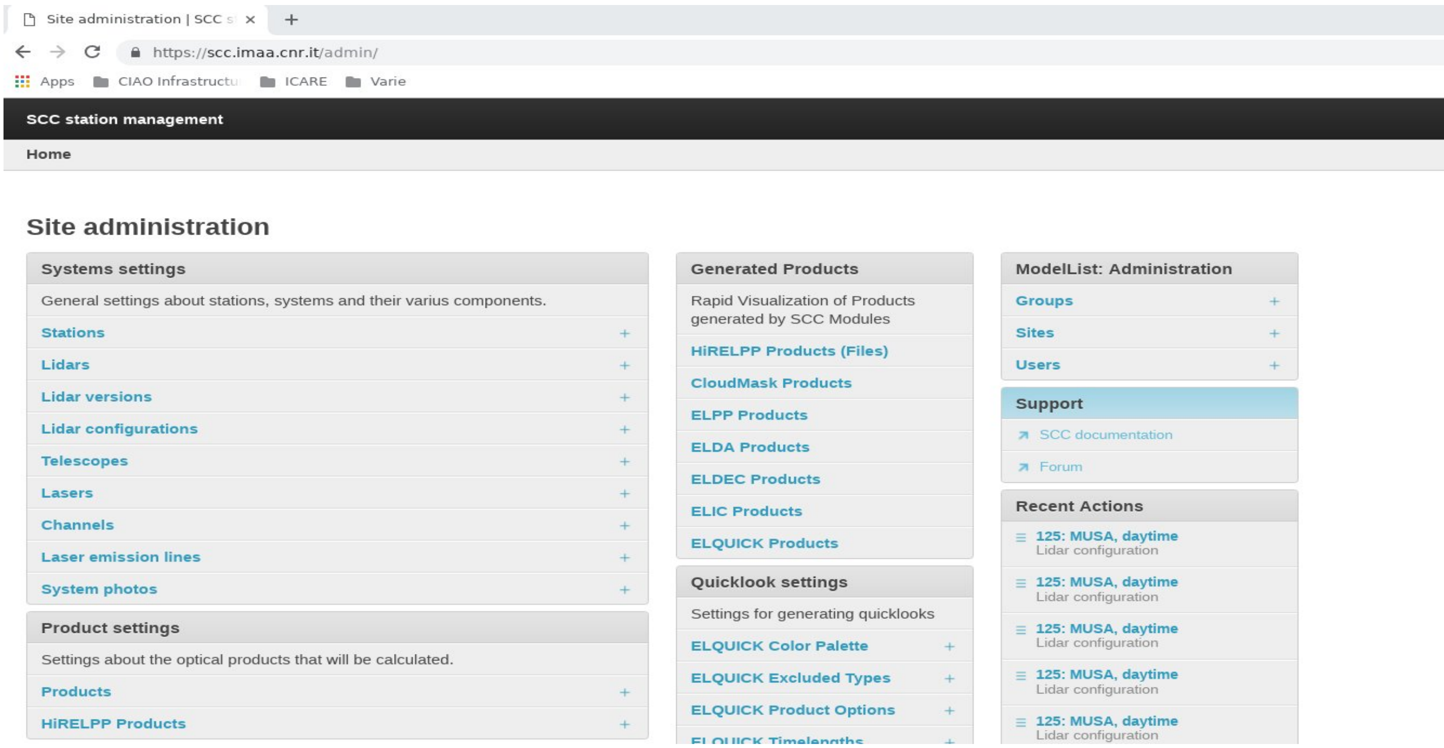
High resolution products

7.1 1. High Resolution products

1. Connect to https://scc.imaa.cnr.it
2. Log-in with your username and password
3. Click on "Station "Admin"



1. In "Product settings" tab click on the "+" at the right of "HiRELPP Products"

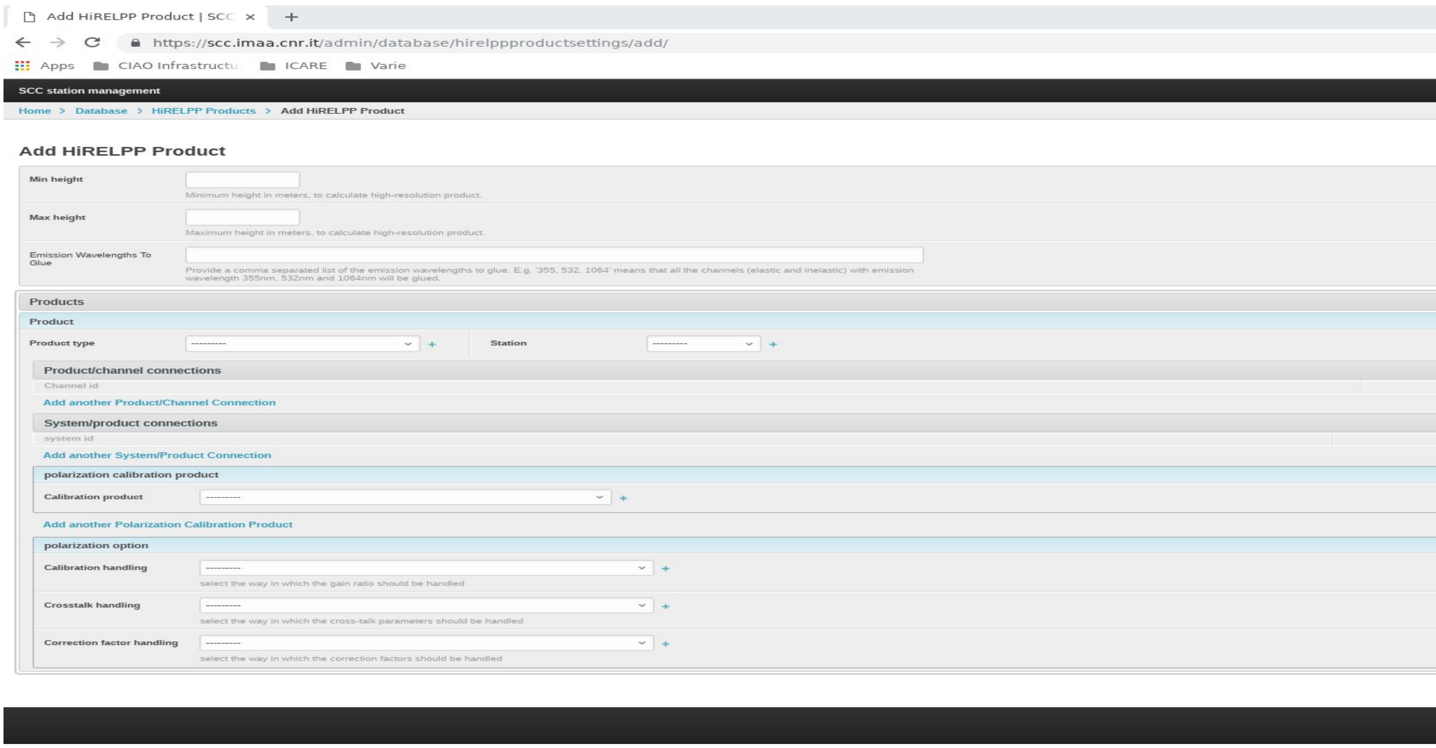


1. Select "Min height" and "Max height"
2. Select "Product types" and "Station"
3. Select all the channels involved in the product calculation in the "Product/channel Connection"

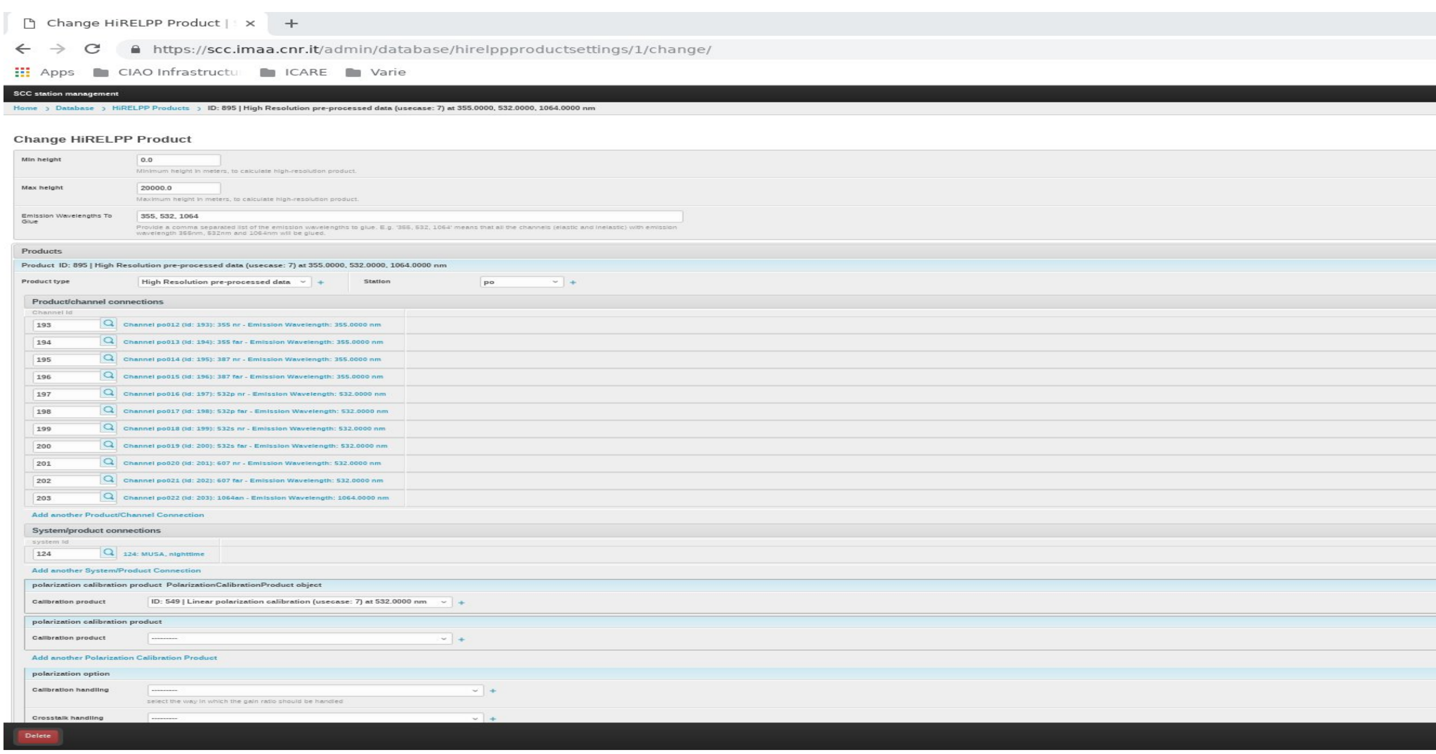
Important: HiRELPP products are designed to be multi-wavelength products. So do not define different HiRELPP products for different wavelengths but put all the wavelengths (channels) in the same HiRELPP product

1. Select the configuration at which the product should be connected in "System/Product Connection"
2. On the base of the emission wavelengths of the channels included in the products, specify the emission wavelengths of the channels that have to be glued (separated by comma). If you leave this field empty the glueing will be NOT performed on any channel.
3. (Optional) If among the channels included in the product, there are cross and parallel channels, select the depolarization calibration product(s) to use for their calibration under "polarization calibration product" and the "polarization option"
4. Finally press "Save"

This is how the HiRELPP product should look like:



Add HiRELPP Product | SCC
https://scc.imaa.cnr.it/admin/database/hirelppproductsettings/add/
Apps
CIAO Infrastructu
ICARE
Varie
SCC station management
Home > Database > HiRELPP Products > Add HiRELPP Product
Add HiRELPP Product
Min height
Minimum height in meters, to calculate high-resolution product.
Max height
Maximum height in meters, to calculate high-resolution product.
Emission Wavelengths To Glue
Provide a comma separated list of the emission wavelengths to glue. E.g. '355, 532, 1064' means that all the channels (elastic and inelastic) with emission wavelength 355nm, 532nm and 1064nm will be glued.
Products
Product
Product type
Station
Product/channel connections
Channel id
Add another Product/Channel Connection
System/product connections
system id
Add another System/Product Connection
polarization calibration product
Calibration product
Add another Polarization Calibration Product
polarization option
Calibration handling
select the way in which the gain ratio should be handled
Crosstalk handling
select the way in which the cross-talk parameters should be handled
Correction factor handling
select the way in which the correction factors should be handled



Change HiRELPP Product |
https://scc.imaa.cnr.it/admin/database/hirelppproductsettings/1/change/
Apps
CIAO Infrastructu
ICARE
Varie
SCC station management
Change HiRELPP Product
Min height
0.0
Max height
20000.0
Emission Wavelengths To Glue
355, 532, 1064
Products
Product type
High Resolution pre-processed data
Station
po
Product/channel connections
Add another Product/Channel Connection
System/product connections
124
124: MUSA, nighttime
Add another System/Product Connection
polarization calibration product
Calibration product
Add another Polarization Calibration Product
polarization option
Calibration handling
Crosstalk handling
Delete

CHAPTER 8

Handbook of instruments

Contents:

8.1 The Handbook of instruments

The SCC database contains all the information needed to complete the handbook of instruments. We hope that this method of communicating system information will be more efficient than exchanging traditional excel files.

8.1.1 Describing your systems

Unfortunately, the SCC does not contain the concept of a system, but only of System Configuration. This means that a station can have a single physical system, but use several configuration in the database (daytime channels, Raman channels, etc). For example, the Alomar station, could have three system configuration but only one system, the "ALOMAR Troposphere Lidar".

To construct the HOI system page, we group different configurations based on the **name** of the system. In the previous example, we need to combine all three configuration of ALOMAR in one system HOI entry. To do this we:

1. Group all system configurations with the same name as one system.
2. Get all distinct channels connected with any of these systems.
3. Get all lasers, telescopes, and emission lines for all these channels.
4. Show all of these to the HOI as one system.

8.1.2 Excluding system configurations or channels

In some cases, you may create some system configuration or channel just for testing purposes. You can exclude these from the handbook of instruments. To do this:

1. Login to the admin section.

2. Go to the page of the system/channel you want to exclude
3. Check the “Exclude from HOI” checkbox.

8.1.3 System from and to dates

If you had several old systems that are no longer working, or if you have performed a major upgrade to your system, it is a good idea to include this information to the HOI. Each system configuration includes the fields **Configuration from** and **Configuration to** that define the time period that the system was active.

CHAPTER 9

Input/Output file formats

In this section of the documentation you can find information about the input and output files formats of the SCC.

9.1 The SCC input netCDF file format

A more detailed version of this document can be found in this `pdf file`.

Note: You can check the format of the files you create using the linked script .

9.1.1 Rationale

The Single Calculus Chain (SCC) is composed by three different modules:

- pre-processing module (*ELPP*)
- optical processing module (*ELDA*)
- depolarization calibrator module (*ELDEC*)

To perfom aerosol optical retrievals the SCC needs not only the raw lidar data but also a certain number of parameters to use in both pre-processing and optical processing stages. The SCC gets these parameters looking at two different locations:

- Single Calculus Chain relational database (SCC_DB)
- Input files

There are some paramenters that can be found only in the input files (those ones changing from measurement to measurement), others that can be found only in the SCC_DB and other ones that can be found in both these locations. In the last case, if a particular parameter is needed, the SCC will search first in the input files and then in SCC_DB. If the parameter is found in the input files, the SCC will keep it without looking into SCC_DB.

The input files have to be submitted to the SCC in NetCDF format. At present the SCC can handle four different types of input files:

1. Raw Lidar Data
2. Sounding Data
3. Overlap
4. Lidar Ratio

As already mentioned, the *Raw lidar data* file contains not only the raw lidar data but also other parameters to use to perform the pre-processing and optical processing. The *Sounding Data* file contains the data coming from a correlative radiosounding and it is used by the SCC for molecular density calculation. The *Overlap* file contains the measured overlap function. The *Lidar Ratio* file contains a lidar ratio profile to use in elastic backscatter retrievals. The *Raw Lidar Data* file is of course mandatory and the *Sounding Data*, *Overlap* and *Lidar Ratio* files are optional. If *Sounding Data* file is not submitted by the user, the molecular density will be calculated by the SCC using model forecast/re-analysis or in case these are not available using the "US Standard Atmosphere 1976". If the *Overlap* file is not submitted by the user, the SCC will get the full overlap height from SCC_DB and it will produce optical results starting from this height. If *Lidar Ratio* file is not submitted by the user, the SCC will consider a fixed value for lidar ratio got from SCC_DB.

The user can decide to submit all these files or any number of them (of course the file *Raw Lidar Data* is mandatory). For example the user can submit together with the *Raw Lidar Data* file only the *Sounding Data* file or only the *Overlap* file.

This document provides a detailed example about the structure of the NetCDF input files to use for SCC data submission. All Earlinet groups should read it carefully because they have to produce such kind of input files if they want to use the SCC for their standard lidar retrievals.

Additionaly, the linked `pdf file` contains tables with all mandatory and optional variables for the netcdf files accepted by the SCC. Table 1 contains a list of dimensions, variables and global attributes that can be used in the NetCDF *Raw Lidar Data* input file. For each of them it is indicated:

- The name. For the multidimensional variables also the corresponding dimensions are reported
- A description explaining the meaning
- The type
- If it is mandatory or optional

As already mentioned, the SCC can get some parameters looking first in the *Raw Lidar Data* input file and then into SCC_DB. This means that to use the parameters stored in SCC_DB the optional variables or optional global attributes must not appear within *Raw Lidar Data* file. This is the suggested and recommended way to use the SCC. Please include optional parameters in the *Raw Lidar Data* only as an exception.

Tables 2, 3, and 4 report all the information about the structure of *Sounding Data*, *Overlap* and *Lidar Ratio* input files respectively.

9.1.2 Example

Let's now consider an example of *Raw Lidar Data* input file. Suppose we want to generate NetCDF input file corresponding to a measurement with the following properties:

Start Date	30th January 2009
Start Time UT	00:00:01
Stop Time UT	00:05:01
Station Name	Dummy station
Earlinet call-sign	ccc
Pointing angle	5 degrees with respect to the zenith

Moreover suppose that this measurement is composed by the following lidar channels:

1. 1064 lidar channel

Emission wavelength=1064nm	Detection wavelength=1064nm
Time resolution=30s	Number of laser shots=1500
Number of bins=3000	Detection mode=analog
Range resolution=7.5m	Polarization state=total

2. 532 cross lidar channel

Emission wavelength=532nm	Detection wavelength=532nm
Time resolution=60s	Number of laser shots=3000
Number of bins=5000	Detection mode=photoncounting
Range resolution=15m	Polarization state=cross (transmitted)

3. 532 parallel lidar channel

Emission wavelength=532nm	Detection wavelength=532nm
Time resolution=60s	Number of laser shots=3000
Number of bins=5000	Detection mode=photoncounting
Range resolution=15m	Polarization state=parallel (reflected)

4. 607 N_2 vibrational Raman channel

Emission wavelength=532nm	Detection wavelength=607nm
Time resolution=60s	Number of laser shots=3000
Number of bins=5000	Detection mode=photoncounting
Range resolution=15m	

Finally let's assume we have also performed dark measurements before the lidar measurements from the 23:50:01 UT up to 23:53:01 UT of 29th January 2009.

Dimensions

Looking at table 1 of the pdf file we have to fix the following dimensions:

```
points
channels
time
nb_of_time_scales
scan_angles
time_bck
```

The dimension `time` is unlimited so we don't have to fix it. We have 4 lidar channels so:

```
channels=4
```

Regarding the dimension `points` we have only one channel with a number of vertical bins equal to 3000 (the 1064nm) and all other channels with 5000 vertical bins. In cases like this the dimension `points` has to be fixed to the maximum number of vertical bins so:

```
points=5000
```

Moreover only one channel (1064nm) is acquired with a time resolution of 30 seconds, all the other channels have a time resolution of 60 seconds. This means that we have to define two different time scales. We have to set:

```
nb_of_time_scales=2
```

The measurement is performed only at one scan angle (5 degrees with respect to the zenith) so:

```
scan_angles=1
```

We have 3 minutes of dark measurements and two different time scales one with 60 seconds time resolution and the other one with 30 seconds time resolution. So we will have 3 different dark profiles for the channels acquired with the first time scale and 6 for the lidar channels acquired with the second time scale. We have to fix the dimension `time_bck` as the maximum between these values:

```
time_bck=6
```

Variables

In this section it will be explained how to fill all the possible variables either mandatory or optional of *Raw Lidar Data* input file.

- `Raw_Data_Start_Time(time, nb_of_time_scales)`

  This 2 dimensional mandatory array has to contain the acquisition start time (in seconds from the time given by the global attribute `RawData_Start_Time_UT`) of each lidar profile. In this example we have two different time scales: one is characterized by steps of 30 seconds (the 1064nm is acquired with this time scale) the other by steps of 60 seconds (532cross, 532parallel and 607nm). Moreover the measurement start time is 00:00:01 UT and the measurement stop time is 00:05:01 UT. In this case we have to define:

  ```
  Raw_Data_Start_Time =
    0, 0,
    60, 30,
    120, 60,
    180, 90,
    240, 120,
    _, 150,
    _, 180,
    _, 210,
    _, 240,
    _, 270 ;
  ```

  The order used to fill this array defines the correspondence between the different time scales and the time scale index. In this example we have a time scale index of 0 for the time scale with steps of 60 seconds and a time scale index of 1 for the other one.

- `Raw_Data_Stop_Time(time, nb_of_time_scales)`

The same as previous item but for the data acquisition stop time. Following a similar procedure we have to define:

```
Raw_Data_Stop_Time =
  60, 30,
  120, 60,
  180, 90,
  240, 120,
  300, 150,
  _, 180,
  _, 210,
  _, 240,
  _, 270,
  _, 300 ;
```

- `Raw_Lidar_Data(time, channels, points)`

  This 3 dimensional mandatory array has to be filled with the time-series of raw lidar data. The photoncounting profiles have to submitted in counts (so as integers) while the analog ones in mV. The order the user chooses to fill this array defines the correspondence between channel index and lidar data.

  For example if we fill this array in such way that:

`Raw_Lidar_Data(time,0,points)`	is the time-series of 1064 nm
`Raw_Lidar_Data(time,1,points)`	is the time-series of 532 cross
`Raw_Lidar_Data(time,2,points)`	is the time-series of 532 parallel
`Raw_Lidar_Data(time,3,points)`	is the time-series of 607 nm

  from now on the channel index 0 is associated to the 1064 channel, 1 to the 532 cross, 2 to the 532 parallel and 3 to the 607nm.

- `Raw_Bck_Start_Time(time_bck, nb_of_time_scales)`

  This 2 dimensional optional array has to contain the acquisition start time (in seconds from the time given by the global attribute `RawBck_Start_Time_UT`) of each dark measurements profile. Following the same procedure used for the variable `Raw_Data_Start_Time` we have to define:

```
Raw_Bck_Start_Time =
  0, 0,
  60, 30,
  120, 60,
  _, 90,
  _, 120,
  _, 150;
```

- `Raw_Bck_Stop_Time(time_bck, nb_of_time_scales)`

  The same as previous item but for the dark acquisition stop time. Following a similar procedure we have to define:

```
Raw_Bck_Stop_Time =
  60, 30,
  120, 60,
  180, 90,
  _, 120,
```

(continues on next page)

(continued from previous page)

```
  _, 150,
  _, 180 ;
```

- `Background_Profile(time_bck, channels, points)`
  This 3 dimensional optional array has to be filled with the time-series of the dark measurements data. The photoncounting profiles have to submitted in counts (so as integers) while the analog ones in mV. The user has to fill this array following the same order used in filling the array `Raw_Lidar_Data`:

`Background_Profile(time_bck,0,points)`	dark time-series at 1064 nm
`Background_Profile(time_bck,1,points)`	dark time-series at 532 cross
`Background_Profile(time_bck,2,points)`	dark time-series at 532 parallel
`Background_Profile(time_bck,3,points)`	dark time-series at 607 nm

- `channel_ID(channels)`
  This mandatory array provides the link between the channel index within the *Raw Lidar Data* input file and the channel ID in SCC_DB. To fill this variable the user has to know which channel IDs in SCC_DB correspond to his lidar channels. For this purpose the SCC, in its final version will provide to the user a special tool to get these channel IDs through a Web interface. At the moment this interface is not yet available and these channel IDs will be communicated directly to the user by the NA5 people.

  Anyway to continue the example let's suppose that the four lidar channels taken into account are mapped into SCC_DB with the following channel IDs:

1064 nm	channel ID=7
532 cross	channel ID=5
532 parallel	channel ID=6
607 nm	channel ID=8

  In this case we have to define:

```
channel_ID = 7, 5, 6, 8 ;
```

- `id_timescale(channels)`
  This mandatory array is introduced to determine which time scale is used for the acquisition of each lidar channel. In particular this array defines the link between the channel index and the time scale index. In our example we have two different time scales. Filling the arrays `Raw_Data_Start_Time` and `Raw_Data_Stop_Time` we have defined a time scale index of 0 for the time scale with steps of 60 seconds and a time scale index of 1 for the other one with steps of 30 seconds. In this way this array has to be set as:

```
id_timescale = 1, 0, 0, 0 ;
```

- `Laser_Pointing_Angle(scan_angles)`
  This mandatory array contains all the scan angles used in the measurement. In our example we have only one scan angle of 5 degrees with respect to the zenith, so we have to define:

```
Laser_Pointing_Angle = 5 ;
```

- `Laser_Pointing_Angle_of_Profiles(time, nb_of_time_scales)`
  This mandatory array is introduced to determine which scan angle is used for the acquisition of each lidar profile. In particular this array defines the link between the time and time scales indexes and the scan angle index. In our example we have a single scan angle that has to correspond to the scan angle index 0. So this array has to be defined as:

```
Laser_Pointing_Angle_of_Profiles =
  0, 0,
  0, 0,
  0, 0,
  0, 0,
  0, 0,
  _, 0,
  _, 0,
  _, 0,
  _, 0,
  _, 0 ;
```

- `Laser_Shots(time, channels)`
  This mandatory array stores the laser shots accumulated at each time for each channel. In our example the number of laser shots accumulated is 1500 for the 1064nm channels and 3000 for all the other channels. Moreover the laser shots do not change with the time. So we have to define this array as:

```
Laser_Shots =
 1500, 3000, 3000, 3000,
 1500, 3000, 3000, 3000,
 1500, 3000, 3000, 3000,
 1500, 3000, 3000, 3000,
 1500, 3000, 3000, 3000,
 1500, _, _, _,
 1500, _, _, _,
 1500, _, _, _,
 1500, _, _, _,
 1500, _, _, _ ;
```

- `Emitted_Wavelength(channels)`
  This optional array defines the link between the channel index and the emission wavelength for each lidar channel. The wavelength has to be expressed in nm. This information can be also taken from SCC_DB. In our example we have:

```
Emitted_Wavelength = 1064, 532, 532, 532 ;
```

- `Detected_Wavelength(channels)`
  This optional array defines the link between the channel index and the detected wavelength for each lidar channel. Here detected wavelength means the value of center of interferential filter expressed in nm. This information can be also taken from SCC_DB. In our example we have:

```
Detected_Wavelength = 1064, 532, 532, 607 ;
```

- `Raw_Data_Range_Resolution(channels)`
  This optional array defines the link between the channel index and the raw range resolution for each channel. If the scan angle is different from zero this quantity is different from the vertical resolution. More precisely if α is the scan angle used and Δz is the range resolution the vertical resolution is calculated as $\Delta z' = \Delta z \cos \alpha$. This

array has to be filled with Δz and not with $\Delta z'$. The unit is meters. This information can be also taken from SCC_DB. In our example we have:

```
Raw_Data_Range_Resolution = 7.5, 15.0, 15.0, 15.0 ;
```

- `Scattering_Mechanism(channels)`
  This optional array defines the scattering mechanism involved in each lidar channel. In particular the following values are adopted:

0	Total elastic backscatter
1	N_2 vibrational Raman backscatter
2	Cross polarization elastic backscatter
3	Parallel polarization elastic backscatter
4	H_2O vibrational Raman backscatter
5	Rotational Raman low quantum number
6	Rotational Raman high quantum number

This information can be also taken from SCC_DB. In our example we have:

```
Scattering_Mechanism = 0, 2, 3, 1 ;
```

- `Signal_Type(channels)`
  This optional array defines the type of signal involved in each lidar channel. In particular the following values are adopted:

0	Total elastic
1	Total elastic near range
2	Total elastic far range
3	N_2 vibrational Raman
4	N_2 vibrational Raman near range
5	N_2 vibrational Raman far range
6	Elastic polarization reflected
7	Elastic polarization transmitted
8	Rotational Raman line close to elastic line
9	Rotational Raman line far from elastic line
10	Elastic polarization reflected near range
11	Elastic polarization reflected far range
12	Elastic polarization transmitted near range
13	Elastic polarization transmitted far range
14	H_2O vibrational Raman backscatter
15	Rotational Raman line far from elastic line near range
16	Rotational Raman line far from elastic line far range
17	Rotational Raman line close to elastic line near range
18	Rotational Raman line close to elastic line far range
19	H_2O vibrational Raman backscatter near range
20	H_2O vibrational Raman backscatter far range
21	Total elastic ultra near range
22	+45 rotated elastic polarization transmitted

Continued on next page

Table 1 – continued from previous page

23	+45 rotated elastic polarization reflected
24	-45 rotated elastic polarization transmitted
25	-45 rotated elastic polarization reflected
26	+45 rotated elastic polarization transmitted near range
27	+45 rotated elastic polarization transmitted far range
28	+45 rotated elastic polarization reflected near range
29	+45 rotated elastic polarization reflected far range
30	-45 rotated elastic polarization transmitted near range
31	-45 rotated elastic polarization transmitted far range
32	-45 rotated elastic polarization reflected near range
33	-45 rotated elastic polarization reflected far range

This information can be also taken from SCC_DB. In our example we have:

```
Signal_Type = 0, 7, 6, 3 ;
```

- `Acquisition_Mode(channels)`
  This optional array defines the acquisition mode (analog or photoncounting) involved in each lidar channel. In particular a value of 0 means analog mode and 1 photoncounting mode. This information can be also taken from SCC_DB. In our example we have:

```
Acquisition_Mode = 0, 1, 1, 1 ;
```

- `Laser_Repetition_Rate(channels)`
  This optional array defines the repetition rate in Hz used to acquire each lidar channel. This information can be also taken from SCC_DB. In our example we are supposing we have only one laser with a repetition rate of 50 Hz so we have to set:

```
Laser_Repetition_Rate = 50, 50, 50, 50 ;
```

- `Dead_Time(channels)`
  This optional array defines the dead time in ns associated to each lidar channel. The SCC will use the values given by this array to correct the photoncounting signals for dead time. Of course for analog signals no dead time correction will be applied (for analog channels the corresponding dead time values have to be set to undefined value). This information can be also taken from SCC_DB. In our example the 1064 nm channel is acquired in analog mode so the corresponding dead time value has to be undefined. If we suppose a dead time of 10 ns for all other channels we have to set:

```
Dead_Time = _, 10, 10, 10 ;
```

- `Dead_Time_Corr_Type(channels)`
  This optional array defines which kind of dead time correction has to be applied on each photoncounting channel. The SCC will correct the data supposing a not-paralyzable channel if a value of 0 is found while a paralyzable channel is supposed if a value of 1 is found. Of course for analog signals no dead time correction will be applied and so the corresponding values have to be set to undefined value. This information can be also taken from SCC_DB. In our example the 1064 nm channel is acquired in analog mode so the corresponding has to be undefined. If we want to consider all the photoncounting signals as not-paralyzable ones: we have to set:

```
Dead_Time_Corr_Type = _, 0, 0, 0 ;
```

- `Trigger_Delay(channels)`
  This optional array defines the delay (in ns) of the middle of the first rangebin with respect to the output laser pulse for each lidar channel. The SCC will use the values given by this array to correct for trigger delay. This information can be also taken from SCC_DB. Let's suppose that in our example all the photoncounting channels are not affected by this delay and only the analog channel at 1064nm is acquired with a delay of 50ns. In this case we have to set:

```
Trigger_Delay = 50, 0, 0, 0 ;
```

- `Background_Mode(channels)`
  This optional array defines how the atmospheric background has to be subtracted from the lidar channel. Two options are available for the calculation of atmospheric background:

  1. Average in the far field of lidar channel. In this case the value of this variable has to be 1
  2. Average within pre-trigger bins. In this case the value of this variable has to be 0

  This information can be also taken from SCC_DB. Let's suppose in our example we use the pre-trigger for the 1064nm channel and the far field for all other channels. In this case we have to set:

```
Background_Mode = 0, 1, 1, 1 ;
```

- `Background_Low(channels)`
  This mandatory array defines the minimum altitude (in meters) to consider in calculating the atmospheric background for each channel. In case pre-trigger mode is used the corresponding value has to be set to the rangebin to be used as lower limit (within pre-trigger region) for background calculation. In our example, if we want to calculate the background between 30000 and 50000 meters for all photoncounting channels and we want to use the first 500 pre-trigger bins for the background calculation for the 1064nm channel we have to set:

```
Background_Low= 0, 30000, 30000, 30000 ;
```

- `Background_High(channels)`
  This mandatory array defines the maximum altitude (in meters) to consider in calculating the atmospheric background for each channel. In case pre-trigger mode is used the corresponding value has to be set to the rangebin to be used as upper limit (within pre-trigger region) for background calculation. In our example, if we want to calculate the background between 30000 and 50000 meters for all photoncounting channels and we want to use the first 500 pre-trigger bins for the background calculation for the 1064nm channel we have to set:

```
Background_High = 500, 50000, 50000, 50000 ;
```

- `Molecular_Calc`
  This mandatory variable defines the way used by SCC to calculate the molecular density profile. The following options are available:

  1. Automatic. In this case the value of this variable has to be 0. First the availablity of model forecast/re-analysis on Cloudnet data portal (https://cloudnet.fmi.fi/) is checked. In case no model data are found US Standard Atmosphere 1976 is used.
  2. Radiosounding. In this case the value of this variable has to be 1
  3. Model forecast/re-analysis. In this case the value of this variable has to be 2. Model data are made available by Cloudnet data portal (https://cloudnet.fmi.fi/).
  4. US Standard Atmosphere 1976. In this case the value of this variable has to be 4

If we decide to use the option 1. or 4. we have to provide also the measured pressure and temperature at lidar station level. Options 1. and 3. are available only for the stations registered on Cloudnet data portal for the delivery of the required model data. The Cloudnet station registration status can be checked by looking at the field 'Cloudnet Station ID' in the station settings. If this field is filled the corresponding station is registered otherwise it is not. In case the station is not registered you can ask to be registered by contacting SCC responsible. Usually model data are made available with a delay of 24h.

For the option 2. a radiosounding file has to be submitted separately in NetCDF format (the structure of this file is summarized in table 2 of the pdf file). Let's suppose we want to use the option 1. so:

```
Molecular_Calc = 0 ;
```

- `Pressure_at_Lidar_Station`
  Because we have chosen the automatic calculation mode to compute the molecular density profile we need to provide the pressure in hPa at lidar station level (in case model data are not available).

```
Pressure_at_Lidar_Station = 1010 ;
```

- `Temperature_at_Lidar_Station`
  Because we have chosen the automatic calculation mode to compute the molecular density profile we need to provide the temperature in C at lidar station level (in case model data are not available).

```
Temperature_at_Lidar_Station = 19.8 ;
```

- `LR_Input(channels)`
  This array is required only for lidar channels for which elastic backscatter retrieval has to be performed. It defines the lidar ratio to be used within this retrieval. Two options are available:

  1. The user can submit a lidar ratio profile. In this case the value of this variable has to be 0.
  2. A fixed value of lidar ratio can be used. In this case the value of this variable has to be 1.

  If we decide to use the option 1. a lidar ratio file has to be submitted separately in NetCDF format (the structure of this file is summarized in table). If we decide to use the option 2. the fixed value of lidar ratio will be taken from SCC_DB. In our example we have to give a value of this array only for the 1064nm lidar channel because for the 532nm we will be able to retrieve a Raman backscatter coefficient. In case we want to use the fixed value stored in SCC_DB we have to set:

```
LR_Input = 1,_,_,_ ;
```

- `DAQ_Range(channels)`
  This array is required only if one or more lidar signals are acquired in analog mode. It gives the analog scale in mV used to acquire the analog signals. In our example we have only the 1064nm channel acquired in analog mode. If we have used a 100mV analog scale to acquire this channel we have to set:

```
DAQ_Range = 100,_,_,_ ;
```

Global attributes

- `Measurement_ID`
  The submission of a raw data file to the SCC is possible only if it contains this global attribute which, in general, can be set to any alphanumeric string composed by 12 or 15 characters. The raw data file can be registered into the SCC database only if the corresponding measurement ID string is not already used by any other raw data file (of any station). To ensure the unicity of the measurement ID string the following format is highly recommened::

<start_date><station_code><start_time>

where:

<start_date>	8 digits measurement start date: YYYYMMDD (for example 20220620)
<station_code>	3 digits code (for example pot)
<star_time>	4 digits measurement start time (UTC): HHMM (for example 2226)

In our example we have to set:

```
Measurement_ID= "20090130ccc0000" ;
```

- `RawData_Start_Date`
  This mandatory global attribute defines the start date of lidar measurements in the format YYYYMMDD. In our case we have:

```
RawData_Start_Date = "20090130" ;
```

- `RawData_Start_Time_UT`
  This mandatory global attribute defines the UT start time of lidar measurements in the format HHMMSS. In our case we have:

```
RawData_Start_Time_UT = "000001" ;
```

- `RawData_Stop_Time_UT`
  This mandatory global attribute defines the UT stop time of lidar measurements in the format HHMMSS. In our case we have:

```
RawData_Stop_Time_UT = "000501" ;
```

- `RawBck_Start_Date`
  This optional global attribute defines the start date of dark measurements in the format YYYYMMDD. In our case we have:

```
RawBck_Start_Date = "20090129" ;
```

- `RawBck_Start_Time_UT`
  This optional global attribute defines the UT start time of dark measurements in the format HHMMSS. In our case we have:

```
RawBck_Start_Time_UT = "235001" ;
```

- `RawBck_Stop_Time_UT`
  This optional global attribute defines the UT stop time of dark measurements in the format HHMMSS. In our case we have:

```
RawBck_Stop_Time_UT = "235301" ;
```

Example of file (CDL format)

To summarize we have the following NetCDF *Raw Lidar Data* file (in CDL format):

```
dimensions:
        points = 5000 ;
        channels = 4 ;
        time = UNLIMITED ; // (10 currently)
        nb_of_time_scales = 2 ;
        scan_angles = 1 ;
        time_bck = 6 ;
variables:
        int channel_ID(channels) ;
        int Laser_Repetition_Rate(channels) ;
        double Laser_Pointing_Angle(scan_angles) ;
        int Signal_Type(channels);
        double Emitted_Wavelength(channels) ;
        double Detected_Wavelength(channels) ;
        double Raw_Data_Range_Resolution(channels) ;
        int Background_Mode(channels) ;
        double Background_Low(channels) ;
        double Background_High(channels) ;
        int Molecular_Calc ;
        double Pressure_at_Lidar_Station ;
        double Temperature_at_Lidar_Station ;
        int id_timescale(channels) ;
        double Dead_Time(channels) ;
        int Dead_Time_Corr_Type(channels) ;
        int Acquisition_Mode(channels) ;
        double Trigger_Delay(channels) ;
        int LR_Input(channels) ;
        int Laser_Pointing_Angle_of_Profiles(time, nb_of_time_scales) ;
        int Raw_Data_Start_Time(time, nb_of_time_scales) ;
        int Raw_Data_Stop_Time(time, nb_of_time_scales) ;
        int Raw_Bck_Start_Time(time_bck, nb_of_time_scales) ;
        int Raw_Bck_Stop_Time(time_bck, nb_of_time_scales) ;
        int Laser_Shots(time, channels) ;
        double Raw_Lidar_Data(time, channels, points) ;
        double Background_Profile(time_bck, channels, points) ;
        double DAQ_Range(channels) ;

// global attributes:
                :Measurement_ID = "20090130ccc0000" ;
                :RawData_Start_Date = "20090130" ;
                :RawData_Start_Time_UT = "000001" ;
                :RawData_Stop_Time_UT = "000501" ;
                :RawBck_Start_Date = "20090129" ;
                :RawBck_Start_Time_UT = "235001" ;
                :RawBck_Stop_Time_UT = "235301" ;

data:

 channel_ID = 7, 5, 6, 8 ;

 Laser_Repetition_Rate = 50, 50, 50, 50 ;

 Laser_Pointing_Angle = 5 ;

 Signal_Type = 0, 7, 6, 3 ;

 Emitted_Wavelength = 1064, 532, 532, 532 ;
```

(continues on next page)

(continued from previous page)

```
Detected_Wavelength = 1064, 532, 532, 607 ;

Raw_Data_Range_Resolution = 7.5, 15, 15, 15 ;

Background_Mode = 0, 1, 1, 1 ;

Background_Low = 0, 30000, 30000, 30000 ;

Background_High = 500, 50000, 50000, 50000 ;

Molecular_Calc = 0 ;

Pressure_at_Lidar_Station = 1010 ;

Temperature_at_Lidar_Station = 19.8 ;

id_timescale = 1, 0, 0, 0 ;

Dead_Time = _, 10, 10, 10 ;

Dead_Time_Corr_Type = _, 0, 0, 0 ;

Acquisition_Mode = 0, 1, 1, 1 ;

Trigger_Delay = 50, 0, 0, 0 ;

LR_Input = 1,_,_,_ ;

DAQ_Range = 100,_,_,_ ;

Laser_Pointing_Angle_of_Profiles =
 0, 0,
 0, 0,
 0, 0,
 0, 0,
 0, 0,
 _, 0,
 _, 0,
 _, 0,
 _, 0,
 _, 0 ;


Raw_Data_Start_Time =
 0, 0,
 60, 30,
 120, 60,
 180, 90,
 240, 120,
 _, 150,
 _, 180,
 _, 210,
 _, 240,
 _, 270 ;

Raw_Data_Stop_Time =
```

(continues on next page)

(continued from previous page)

```
  60, 30,
  120, 60,
  180, 90,
  240, 120,
  300, 150,
  _, 180,
  _, 210,
  _, 240,
  _, 270,
  _, 300 ;


 Raw_Bck_Start_Time =
  0, 0,
  60, 30,
  120, 60,
  _, 90,
  _, 120,
  _, 150;


Raw_Bck_Stop_Time =
  60, 30,
  120, 60,
  180, 90,
  _, 120,
  _, 150,
  _, 180 ;


 Laser_Shots =
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _ ;


 Raw_Lidar_Data = ...

 Background_Profile = ...
```

The name of the input file should have the following format:

```
Measurement_ID.nc
```

so in the example the filename should be 20090130ccc0000.nc.

Please keep in mind that in case you submit a file like the previous one all the parameters present in it will be used by the SCC even if you have different values for the same parameters within the SCC_DB. If you want to use the values already stored in SCC_DB (this should be the usual way to use SCC) the *Raw Lidar Data* input file has to be modified as follows:

```
dimensions:
        points = 5000 ;
        channels = 4 ;
        time = UNLIMITED ; // (10 currently)
        nb_of_time_scales = 2 ;
        scan_angles = 1 ;
        time_bck = 6 ;
variables:
        int channel_ID(channels) ;
        double Laser_Pointing_Angle(scan_angles) ;
        double Background_Low(channels) ;
        double Background_High(channels) ;
        int Molecular_Calc ;
        double Pressure_at_Lidar_Station ;
        double Temperature_at_Lidar_Station ;
        int id_timescale(channels) ;
        int Laser_Pointing_Angle_of_Profiles(time, nb_of_time_scales) ;
        int Raw_Data_Start_Time(time, nb_of_time_scales) ;
        int Raw_Data_Stop_Time(time, nb_of_time_scales) ;
        int Raw_Bck_Start_Time(time_bck, nb_of_time_scales) ;
        int Raw_Bck_Stop_Time(time_bck, nb_of_time_scales) ;
        int LR_Input(channels) ;
        int Laser_Shots(time, channels) ;
        double Raw_Lidar_Data(time, channels, points) ;
        double Background_Profile(time_bck, channels, points) ;
        double DAQ_Range(channels) ;

// global attributes:
                :Measurement_ID = "20090130ccc0000" ;
                :RawData_Start_Date = "20090130" ;
                :RawData_Start_Time_UT = "000001" ;
                :RawData_Stop_Time_UT = "000501" ;
                :RawBck_Start_Date = "20090129" ;
                :RawBck_Start_Time_UT = "235001" ;
                :RawBck_Stop_Time_UT = "235301" ;

data:

 channel_ID = 7, 5, 6, 8 ;

 Laser_Pointing_Angle = 5 ;

 Background_Low = 0, 30000, 30000, 30000 ;

 Background_High = 500, 50000, 50000, 50000 ;

 Molecular_Calc = 0 ;

 Pressure_at_Lidar_Station = 1010 ;

 Temperature_at_Lidar_Station = 19.8 ;

 id_timescale = 1, 0, 0, 0 ;
```

(continues on next page)

(continued from previous page)

```
LR_Input = 1,_,_,_ ;

DAQ_Range = 100,_,_,_ ;

Laser_Pointing_Angle_of_Profiles =
 0, 0,
 0, 0,
 0, 0,
 0, 0,
 0, 0,
 _, 0,
 _, 0,
 _, 0,
 _, 0,
 _, 0 ;


Raw_Data_Start_Time =
 0, 0,
 60, 30,
 120, 60,
 180, 90,
 240, 120,
 _, 150,
 _, 180,
 _, 210,
 _, 240,
 _, 270 ;

Raw_Data_Stop_Time =
 60, 30,
 120, 60,
 180, 90,
 240, 120,
 300, 150,
 _, 180,
 _, 210,
 _, 240,
 _, 270,
 _, 300 ;


Raw_Bck_Start_Time =
 0, 0,
 60, 30,
 120, 60,
 _, 90,
 _, 120,
 _, 150;


Raw_Bck_Stop_Time =
 60, 30,
 120, 60,
 180, 90,
 _, 120,
 _, 150,
```

(continues on next page)

(continued from previous page)

```
  _, 180 ;


 Laser_Shots =
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, 3000, 3000, 3000,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _,
  1500, _, _, _ ;


 Raw_Lidar_Data = ...

 Background_Profile = ...
```

This example file contains the minimum collection of mandatory information that has to be found within the *Raw Lidar Data* input file. If it is really necessary, the user can decide to add to these mandatory parameters any number of additional parameters considered in the previous example.

Finally, suppose we want to make the following changes with respect to the previous example:

1. use a sounding file for molecular density calculation instead of "US Standar Atmosphere 1976"
2. supply a lidar ratio profile to use in elastic backscatter retrieval instead of a fixed value
3. provide a overlap function for overlap correction

In this case we have to generate the following NetCDF additional files:

- `rs_20090130ccc0000.nc`
  The name of *Sounding Data* file has to be computed as follows:
  `"rs_"+Measurement_ID`
  The structure of this file is summarized in table 2 of the pdf.
- `ov_20090130ccc0000.nc`
  The name of *Overlap* file has to be computed as follows:
  `"ov_"+Measurement_ID`
  The structure of this file is summarized in table 3 of the pdf.
- `lr_20090130ccc0000.nc`
  The name of *Lidar Ratio* file has to be computed as follows:
  `"lr_"+Measurement_ID`
  The structure of this file is summarized in table 4 of the pdf.

Moreover we need to apply the following changes to the *Raw Lidar Data* input file:

1. Change the value of the variable `Molecular_Calc` as follows:

   ```
   Molecular_Calc = 1 ;
   ```

   Of course the variables `Pressure_at_Lidar_Station` and `Temperature_at_Lidar_Station` are not necessary anymore.

2. Change the values of the array `LR_Input` as follows:

```
LR_Input = 0,_,_,_ ;
```

3. Add the global attribute `Sounding_File_Name`

```
Sounding_File_Name = "rs_20090130ccc0000.nc" ;
```

4. Add the global attribute `LR_File_Name`

```
LR_File_Name = "lr_20090130ccc0000.nc" ;
```

5. Add the global attribute `Overlap_File_Name`

```
Overlap_File_Name = "ov_20090130ccc0000.nc" ;
```

9.2 SCC products filename

The filename of all SCC products has the following structure:

```
<station_code>_<product_type_ID>_<wavelength>_<productID>_<starttime>_<stoptime>_<measurementID>_<SCC_module>_<SCC\_version>.nc
```

where:

<*station_code*>	3 digits code (for example pot)
<*product_type_ID*>	numeric 3 digits (for example 002)
<*wavelength*>	4 digits reporting the wavelength in nm (for example 0532)
<*productID*>	numeric 7 digits (for example 0000213)
<*starttime*>	YYYYMMDDHHMM (for example 202102031832)
<*stoptime*>	YYYYMMDDHHMM (for example 202102031944)
<*measurementID*>	measurementID string (15 characters)
<*SCC_module*>	SCC module (hirelpp,cloudscreen,elpp,eldec,elda,elic)
<*SCC_version*>	SCC version (for example v5.2.0)

The field <*product_type_ID*> is codified as it follows:

Product_type_ID	Description
0	Raman Bacskcatter
1	Extinction only
2	Lidar Ration and Extinction
3	Elastic Backscatter
6	Linear Polarization Calibration
7	Raman Backscatter and Linear Depolarization Ratio
8	Elastic Backscatter and Linear Depolarization Ratio
9	High Resolution Pre-Processed data

The field <*wavelength*> is available only for ELEC and ELDA data products (for all the other SCC data products is missing). For CloudScreen products the fields <*product_type_ID*>, <*wavelength*>, <*productID*> are missing.

9.3 SCC products

9.3.1 Introduction

The Single Calculus Chain (SCC) is the standard EARLINET tool to perform automatic and quality checked analysis of raw lidar data. It is composed by the following modules:

- HiRELPP (High Resolution ELPP)
- CloudScreen (SCC cloud screen module)
- ELPP (EARLINET Lidar Pre-Processor)
- ELDA (EARLINET Lidar Data Analizer)
- ELDEC (EARLINET Lidar DEpolarization Calibrator)
- ELIC (EARLINET LIdar Calibrator)
- ELQUICK (EARLINET Lidar QUICJKlook)

9.3.2 HiRELPP

The HiRELPP module implements the corrections to be applied to the raw lidar signals before they can be used to derive higher level products. All the operations implemented in HiRELPP are designed to preserve both the vertical and time resolution as high as possible. Some instrumental effects (like for example, dead-time correction, trigger-delay correction, overlap correction, atmospheric and electronic background subtraction, low- and high-range automatic signal glueing) are corrected following the recommendations provided by the EARLINET quality assurance program.

Dead-time correction

The dead-time corresponds to a maximum count rate. The dead- time causes a non-linearity between the actual intensity at the photo-multiplier photocathode and the counted events, which can be described theoretically by means of photon statistics. Actual detector can be modelled as the paralyzable and the non-paralyzable model. Once information about the model to use for describing the counting system and the dead-time value is determined, based on standard operating procedures defined by the ACTRIS Center for Aerosol remote Sensing), these are provided to HiRELPP and the acquired counts are corrected for the dead-time effect.

Trigger-delay correction

In general, the data acquisition unit of a lidar system gets a trigger from the laser to start the signal recording. Due to the electronic circuits in the laser and in the data acquisition unit, there is always a delay between the outgoing laser pulse and the time at which the acquisition system actually starts to record the lidar profile. If this trigger delay is not properly taken into account, a systematic error is made in associating each lidar range bin with the corresponding atmospheric range. Once the correct measurement of the real trigger delay is done for each detection channel following the procedure indicated CARS, such information is inserted in the SCC configuration and HiRELPP correct acquired lidar signals for the trigger-delay.

Atmospheric and electronic background subtraction

The lidar signal has a constant background made of an atmospheric component and an electric component. This background can be determined either in the far range of the lidar signal, far enough that the expected contribution from atmospheric backscatter is negligible, or in the pre-trigger range before the laser pulse, where the signal must be free of electronic distortions. Each one of this option can be defined into HiRELPP. Additionally, it is possible to subtract so-called dark signals, which are measured, for example, with a fully obscured telescope so that no light from the atmosphere reaches the detectors and only eventual electronic distortions are left. This allows HiRELPP to remove potential distortions affecting analog lidar signals.

Low- and high-range automatic signal glueing

Lidar signals can cover a quite large dynamic range, because the intensity of the light backscattered from the aerosol-laden boundary layer in the near range (e.g. at 0.5 km altitude) is several orders of magnitudes higher than the intensity of the light backscattered from the rather clean troposphere (e.g. at 10 km altitude). As it is demanding to cover this large dynamic range with one data acquisition channel with linear response, several approaches are used to overcome this problem. One option is to split the signal output from a single photomultiplier into two signals and to record one signal using analog detection mode and the other with the photon-counting method. Another option is to split the lidar signal optically using a beam splitter and to detect the split components with two detectors and subsequent data acquisitions. A third option is to use two (or more) telescopes with separate detection electronics. Both SCC preprocessors (HiRELPLP and ELPP) glues the signals for the first 2 options, while gluing is implemented directly at optical property level in the third case (ELDA). Before gluing, the near-range and the far-range signals need to be screened for low-level clouds, corrected for instrumental effects like dead time, trigger delay, etc., and the backgrounds have to be subtracted as explained above. HiRELPP and ELPP contains a fully automatic algorithm for the gluing of analog and photon-counting signals as well as for the gluing of two photon-counting signals. The algorithm is implemented through three main steps: the procedure starts with the determination of a first guess of the gluing region, after that, the algorithm optimizes the gluing region performing statistical tests (implemented only in ELPP) and finally, the signals are glued in the optimal gluing region.

The typical HiRELPP products are netCDF pre-processed files containing pre-processed (un-calibrated) range corrected time series at instrumental vertical and time resolution. If the lidar instrument has polarization capabilities the volume linear depolarization ratio is provided as well.

9.3.3 CloudScreen

Lidar data contaminated by clouds has to be skipped because the retrieval algorithm implemented in the SCC are optimized for aerosol and may produce unreliable results when applied to clouds. The aim of the CloudScreen module is to detect clouds by ingesting as input un-calibrated high resolution pre-processed range corrected signal timeseries (HiRELPP products). The output of CloudScreen module is a netCDF file containing a 2-dimensional grid (x axis: time y axis: altitude) with the same resolution as the corresponding HiRELPP product, in which each pixel is flagged as cloud free or cloud contaminated. This information is then transferred to other SCC modules for the automatic removal of the cloud contribution within the aerosol optical property products.

9.3.4 ELPP

The ELPP module implements all the needed corrections and transformations to be applied to the raw data before they can be used to derive the optical products at low temporal/spatial resolution. As HiRELPP, ELPP implements correction of some instrumental effects (like for example, dead-time correction, trigger-delay correction, overlap correction, atmospheric and electronic background subtraction, low- and high-range automatic signal glueing) following the recommendations provided by the EARLINET quality assurance program. Additionally, to HiRELPP, time integration or vertical smoothing is performed by ELPP to meet the required condition on the products statistical error (defined in the SCC database for each data product type). ELPP makes also advantage of the CloudScreen output products so that signals affected by low clouds are automatically removed already at level of lidar pre-processor. Besides these corrections, ELPP is also responsible to generate the molecular signal needed to calculate the aerosol optical products. In both aerosol backscatter (Klett, 1981; Fernald, 1984; Di Girolamo et al., 1999; Ansmann et al., 1992a; Ferrare et al., 1998) and extinction (Ansmann et al., 1990, 1992b) retrievals the molecular contribution to the atmospheric extinction and transmissivity are required as input, which are calculated by ELPP at the emission and detection wavelengths in terms of vertical profiles at the same vertical resolution as the pre-processed lidar signals. The molecular number density profile is calculated by ELPP from vertical profiles of temperature T(z) and pressure P(z) using the ideal gas law and assuming as 1 the value of the air compressibility factor. Temperature and pressure profiles are either calculated from standard atmosphere model or taken from the measurements of a close-by radiosounding that can be provided to the SCC as a separate input file or provided by model data profiles. Once the molecular number density is obtained, the calculation of the molecular optical parameters, i.e., the backscatter and extinction coefficients, is done following the procedure reported in Bucholtz (1995) and Miles et al. (2001). More details about implemented algorithms in ELPP are reported in D'Amico et al., (2016). The typical ELPP products consist of netCDF pre-processed files containing low resolution pre-processed (un-calibrated) range corrected time series.

9.3.5 ELDA

ELDA applies the algorithms for the retrieval of aerosol optical parameters to the low resolution pre-processed signals, produced by ELPP module. The module provides aerosol optical products in a flexible way choosing from a set of possible pre-defined analysis procedures.

ELDA implements: - retrieval of aerosol extinction profile - retrieval of Raman aerosol backscatter profile - retrieval of elastic aerosol backscatter profile - particle/volume depolarization ratio profile

An automatic vertical-smoothing and time-averaging technique selects the optimal smoothing level as a function of altitude on the base of different thresholds on product uncertainties fixed in the SCC database for each product. Currently, ELDA delivers only optical products at a single wavelength (so for a multi-wavelength lidar, ELDA generates several independent optical products each referring to a single wavelength). Full description of implemented algorithms is reported in Mattis et al., (2016). For all products and retrieval algorithms, the user can choose whether the statistical uncertainties shall be calculated with the Monte Carlo method or by means of error propagation. The only exception are retrievals with the Klett-Fernald a lgorithm for which the estimation of uncertainties is implemented only with Monte Carlo method. Currently, the separated handling of statistical errors of the lidar signals, of systematic errors of the lidar signals, and of uncertainties of the retrieval algorithms is under research within the EARLINET community. ELDA allows for the automated vertical smoothing and temporal averaging of the derived products. The user has the option to adjust the degree of smoothing and averaging of each individual product by setting several parameters. In general, those parameters and constraints can be defined for two different altitude regions, below and above 2 km altitude. Two threshold values for the maximum allowable relative statistical error of the product below and above 2 km altitude (meaning high expected aerosol load and low aerosol load, respectively) can be defined. Beside these user-defined constraints, there are fixed limitations concerning the maximum allowable smoothing and averaging: it is not allowed to apply a smoothing that would result in effective vertical resolutions larger than 500m and 2km below and above 2km altitude, respectively. All methods of calculating profiles of particle backscatter coefficients include a certain calibration procedure. Usually a particle-free region in the free troposphere where the aerosol backscatter is assumed as null is used for calibration. A calibration window of user-defined width is shifted through the altitude region, where particle-free conditions typically occur (user-defined calibration interval). For each window position, the average and standard deviation of the signal or signal ratio is calculated. It is assumed that the window position where the signal or signal ratio has its minimum is closest to the assumed particle-free conditions. The average value within this calibration window and its standard deviation are used to estimate the calibration factor and its statistical uncertainty. If the user knows from ancillary data, e.g., from sun-photometer observations or from climatological data of the stratospheric particle load, that there is no particle-free altitude layer, it is possible to provide backscatter ratios different from 1 as calibration value. ELDA implements the derivative calculation into the aerosol extinction algorithm as derivative of the pre-processed signals by weighted or non-weighted linear fit method. Finally, concerning the assumptions needed in terms of Angstrom exponent (extinction calculation) and /or lidar ratio (elastic backscatter retrieval), it is possible to define in the SCC configuration the values to be used. In particular it is possible to include a lidar ratio (Angstrom) profile in order to improve the overall quality of the product. These values can be provided to the SCC together with the raw signals and are passed by ELPP to ELDA.

9.3.6 ELIC

The ELIC module calibrates both high- and low-resolution pre-processed products (HiRELPP end ELPP products respectively) using the same calibration constant computed by ELDA during the retrieval of low-resolution optical aerosol properties (elastic/Raman bacskcatter calibration). As already mentioned, both HiRELPP and ELPP deliver pre-processed range corrected signal timeseries. Pre-processed range corrected signals are not considered robust lidar products because even if they are proportional to the concentration of atmospheric backscatterers, they depend on specific lidar instrumental characteristics as well. In the retrieval of aerosol optical products (like for example aerosol backscatter), the range corrected signals are used as input and special calibration techniques are used to remove the instrumental dependence. The more is the signal to noise ratio the better is the result of these calibration techniques. Usually, a way to increase the signal to noise ratio is to degrade the time and/or space resolution of the input signals. In general, it is more demanding to get a reliable calibration when working with high resolution lidar data. This is the reason why in the SCC workflow, the calibration is done by ELDA which deals with un-calibrated low-resolution range corrected signals. Anyway, if we assume that the instrumental conditions are stable in the time interval in which the measurements take place, it is possible to use the calibration constants retrieved by ELDA calibrating low resolution signals also to calibrate the high resolution timeseries measured in the same time window. This is the main goal of the ELIC module which runs right after ELDA, gets the calibration constants retrieved by ELDA for all lidar channels and calibrates the corresponding high- and low-resolution range corrected signal timeseries. The ELIC products are netCDF files containing fully calibrated quantities like total attenuated backscatter and volume depolarization ratio.

9.3.7 ELDEC

All the participating stations operate lidar equipped with at least 2 channels detecting independent polarization states of backscattered light and, as consequence, can deliver atmospheric volume/particle depolarization ratio profiles. Anyway, to calculate the volume/particle depolarization ratio from the ratio of these polarization channels an accurate calibration is needed. ELDEC module provide this calibration parameter following the quality assurance procedures defined within ACTRIS CARS (Centre for Aerosol Remote Sensing). In particular, the depolarization calibration is made by submitting to the SCC special raw depolarization calibration datasets.

9.3.8 ELQUICK

The ELQUICK module generates standardized lidar quicklook for the whole ACTRIS/EARLINET network. Lidar quicklooks (png images) are useful representation of the high resolution timeseries of total attenuated backscatter and/or volume depolarization profiles contained in the ELIC products which can be considered as the two-dimensional pixel grid. The number of vertical pixels of this grid is the number of points of the total attenuated backscatter (or volume depolarization ratio) vertical profile while the number of horizontal pixels is to the number of total attenuated backscatter (or volume depolarization ratio) profiles included in the time series. The color corresponding to each individual pixel is, instead, connected to the value of the total attenuated backscatter (or volume depolarization ratio) at a given altitude and time. In this way, by observing such quicklook images it is easy to visualize aerosol layers and their evolution in both time and space.

9.3.9 File Format

All the SCC products are files in Network Common Data Form (NetCDF) which is a well known self-describing, machine-independent data format that support the creation, access, and sharing of array-oriented scientific data. For more information about NetCDF format: http://www.unidata.ucar.edu/software/netcdf/.

The NetCDF is a binary format that allows the definition of multi-dimensional variables of several types (integers, double, character, etc). For each variable it is possible to define one or more attributes where to specify variable properties like units, long name, description, etc.

It is possible to define global attributes which are not related to a specific variable but to the whole file.

A NetCDF file is composed by four different section:

dimensions this section contains all the dimensions used in the definition of all the variables included in the NetCDF file

variables this section contains all the variables stored in the NetCDF file. Each variable is defined as a multi-dimensional array of a specific type and with all the dimensions defined in the dimensions section

global attributes this section lists all the attributes referring to the whole file. As the variable the attributes (global or the one attached to a specific variable) can be of different type

data in this section the data contained in each variable defined in variable section is stored. Attribute values (both global or related to a specific variable) are not reported in data section but directly in variable or global attribute sections.

HiRELPP Products

This section describe the structure of HiRELPP products.

HiRELPP Products: dimensions

The following dimensions are defined in HiRELPP products:

- time
- level
- channel
- depolarization
- angle
- nv
- nc

The dimension *time* (mandatory) specifies the number of RCS (for each channel) composing all the pre-processed time-series reported in the products.

The dimension *level* (mandatory) represents the number of rangebins characterizing the pre-processed RCS.

The dimension *channel* (mandatory) indicates the number channels at which the RCS time-series included in the product refer to.

The dimension *depolarization* (optional) indicates how many volume depolarization time-series are included in the products.

The dimension *angle* (mandatory) takes into account how many zenith scan angles have been used to measure the RCS time-series reported into the products.

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

The dimension *nc* (optional) is ancillary and provides the number of channels involved in the calculation of RCS or volume depolarization time-series.

HiRELPP Products: variables

latitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *double latitude*

longitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *double longitude*

station_altitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *double station_altitude*

altitude

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(time, level)*

range

Type *double*

Dimensions *level*

Variable_Type Mandatory

Description lidar range scale

Units m

Definition *double range(level)*

laser_pointing_angle

Type *double*

Dimensions *angle*

Variable_Type Mandatory

Description laser pointing angle with respect to the zenith

Units degrees

Definition *double laser_pointing_angle(angle)*

laser_pointing_angle_of_profile

Type *int*

Dimensions *angle*

Variable_Type Mandatory

Description index of scan angle at which each single raw profile has been detected

Units –

Definition *int laser_pointing_angle_of_profile(angle)*

shots

Type *int*

Dimensions *time*

Variable_Type Mandatory

Description accumulated laser shots

Units –

Definition *int shots(time)*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

range_corrected_signal_channel_id

Type *int*

Dimensions *channel, nc*

Variable_Type Optional

Description SCC channel ID list involved in the calculation of the range corrected signal total time-series

Units –

Definition *int range_corrected_signal_channel_id(channel, nc)*

range_corrected_signal_channel_name

Type *string*

Dimensions *channel*

Variable_Type Mandatory

Description Channel name for the range corrected signal timeseries

Units –

Definition *string range_corrected_signal_channel_name(channel)*

range_corrected_signal_emission_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Emission wavelength used to measure the range corrected signal timeseries

Units *nm*

Definition *double range_corrected_signal_emission_wavelength(channel)*

range_corrected_signal_detection_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Detection wavelength used to measure the range corrected signal timeseries

Units *nm*

Definition *double range_corrected_signal_detection_wavelength(channel)*

range_corrected_signal_range

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal range type

Units –

Definition *byte range_corrected_signal_range(channel)*

range_corrected_signal_scatterers

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal scatterer type

Units –

Definition *byte range_corrected_signal_scatterers(channel)*

range_corrected_signal_detection_mode

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal detection mode

Units –

Definition *byte range_corrected_signal_detection_mode(channel)*

near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_minimum(channel, time)*

near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_maximum(channel, time)*

ultra_near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_minimum(channel, time)*

ultra_near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Maximum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_maximum(channel, time)*

range_corrected_signal

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Range corrected signal timeseries

Units –

Definition *double range_corrected_signal(channel, time, level)*

range_corrected_signal_statistical_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Statistical error (uncertainty) on range corrected signal timeseries

Units –

Definition *double range_corrected_signal_statistical_error(channel, time, level)*

range_corrected_signal_systematic_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Systematic error (uncertainty) on range corrected signal timeseries

Units –

Definition *double range_corrected_signal_statistical_error(channel, time, level)*

atmospheric_background

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Mean uncalibrated atmospheric background calculated from lidar signal

Units –

Definition *double atmospheric_background(channel, time)*

atmospheric_background_stdev

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Standard deviation of uncalibrated atmospheric background calculated from lidar signal

Units –

Definition *double atmospheric_background_stdev(channel, time)*

atmospheric_background_sterr

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Standard error of uncalibrated atmospheric background calculated from lidar signal

Units –

Definition *double atmospheric_background_sterr(channel, time)*

atmospheric_background_min

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Minimum uncalibrated atmospheric background calculated from lidar signal

Units –

Definition *double atmospheric_background_min(channel, time)*

atmospheric_background_max

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Maximum uncalibrated atmospheric background calculated from lidar signal

Units –

Definition *double atmospheric_background_max(channel, time)*

polarization_gain_factor_measurementid

Type *string*

Dimensions *depolarization*

Variable_Type Optional

Description Measurement ID corresponding to the polarization measurements used to calibrate polarization channels

Units –

Definition *string polarization_gain_factor_measurementid(depolarization)*

polarization_gain_factor

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain factor (eta*)

Units –

Definition *double polarization_gain_factor(depolarization)*

polarization_gain_factor_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_statistical_error(depolarization)*

polarization_gain_factor_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_systematic_error(depolarization)*

polarization_gain_factor_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_start_datetime(depolarization)*

polarization_gain_factor_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_stop_datetime(depolarization)*

polarization_gain_factor_correction

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain correction factor (K)

Units –

Definition *double polarization_gain_factor_correction(depolarization)*

polarization_gain_factor_correction_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_statistical_error(depolarization)*

polarization_gain_factor_correction_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_systematic_error(depolarization)*

polarization_gain_factor_correction_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain corrrection factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_start_datetime(depolarization)*

polarization_gain_factor_correction_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain correction factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_stop_datetime(depolarization)*

volume_linear_depolarization_ratio_channel_id

Type *int*

Dimensions *depolarization, nc*

Variable_Type Optional

Description SCC channel ID list involved in the calculation of the volume linear depolarization ratio timeseries

Units –

Definition *int volume_linear_depolarization_ratio_channel_id(depolarization, nc)*

volume_linear_depolarization_ratio_channel_name

Type *string*

Dimensions *depolarization*

Variable_Type Optional

Description Channel name for volume linear depolarization ration timeseries

Units –

Definition *string volume_linear_depolarization_ratio_channel_name(depolarization)*

volume_linear_depolarization_ratio_wavelength

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Wavelength at which the volume linear depolarization ratio is calculated

Units *nm*

Definition *double volume_linear_depolarization_ratio_wavelength(depolarization)*

volume_linear_depolarization_ratio_range

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Bitmask field indicating the volume linear depolarization range type

Units –

Definition *byte volume_linear_depolarization_ratio_range(depolarization)*

volume_linear_depolarization_ratio_scatterers

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Bitmask field indicating the volume linear depolarization scatteter type

Units –

Definition *byte volume_linear_depolarization_ratio_scatterers(depolarization)*

volume_linear_depolarization_ratio

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio(depolarization, time, level)*

volume_linear_depolarization_ratio_statistical_error

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Statistical error (uncertanity) on volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio_statistical_error(depolarization, time, level)*

volume_linear_depolarization_ratio_systematic_error

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Systematic error (uncertanity) on volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio_systematic_error(depolarization, time, level)*

cloud_mask_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the cloudmask type

Units –

Definition *byte cloud_mask_type*

cloud_mask

Type *byte*

Dimensions *time, level*

Variable_Type Optional

Description Bitmask describing the cloudmask

Units –

Definition *byte cloud_mask(time, level)*

HiRELPP Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

CloudScreen Products

This section describe the structure of CloudScreen products.

CloudScreen Products: dimensions

The following dimensions are defined in CloudScreen products:

- time
- level
- nv

The dimension *time* (mandatory) specifies the number of cloudmask vertical profiles reported in the products.

The dimension *level* (mandatory) represents the number of rangebins characterizing each cloudmask verical profile.

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

CloudScreen Products: variables

latitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *double latitude*

longitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *double longitude*

station_altitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *double station_altitude*

altitude

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(time, level)*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

automatic_cloud_mask

Type *byte*

Dimensions *time, level*

Variable_Type Mandatory

Description Automatic cloudmask

Units –

Definition *byte automatic_cloud_mask(time, level)*

manual_cloud_mask

Type *byte*

Dimensions *time, level*

Variable_Type Optional

Description Manual cloudmask

Units –

Definition *byte manual_cloud_mask(time, level)*

CloudScreen Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_algorithm

Type *string*

Global_Attribute_Type Mandatory

Description Name of the cloudmask algorithm

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

automatic_mask_channels

Type *string*

Global_Attribute_Type Mandatory

Description Channels used to generate the cloumask (SCC channel names)

ELPP Products

This section describe the structure of ELPP products.

ELPP Products: dimensions

The following dimensions are defined in ELPP products:

- time
- level
- channel
- depolarization
- angle
- nv
- nc

The dimension *time* (mandatory) specifies the number of RCS (for each channel) composing all the pre-processed time-series reported in the products.

The dimension *level* (mandatory) represents the number of rangebins characterizing the pre-processed RCS.

The dimension *channel* (mandatory) indicates the number channels at which the RCS time-series included in the product refer to.

The dimension *depolarization* (optional) indicates how many volume depolarization time-series are included in the products.

The dimension *angle* (mandatory) takes into account how many zenith scan angles have been used to measure the RCS time-series reported into the products.

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

The dimension *nc* (optional) is ancillary and provides the number of channels involved in the calculation of RCS or volume depolarization time-series.

ELPP Products: variables

latitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *double latitude*

longitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *double longitude*

station_altitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *double station_altitude*

altitude

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(time, level)*

range

Type *double*

Dimensions *level*

Variable_Type Mandatory

Description lidar range scale

Units m

Definition *double range(level)*

laser_pointing_angle

Type *double*

Dimensions *angle*

Variable_Type Mandatory

Description laser pointing angle with respect to the zenith

Units degrees

Definition *double laser_pointing_angle(angle)*

laser_pointing_angle_of_profile

Type *int*

Dimensions *angle*

Variable_Type Mandatory

Description index of scan angle at which each single raw profile has been detected

Units –

Definition *int laser_pointing_angle_of_profile(angle)*

shots

Type *int*

Dimensions *time*

Variable_Type Mandatory

Description accumulated laser shots

Units –

Definition *int shots(time)*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

cloud_mask_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the cloudmask type

Units –

Definition *byte cloud_mask_type*

cloud_mask

Type *byte*

Dimensions *time, level*

Variable_Type Optional

Description Bitmask describing the cloudmask

Units –

Definition *byte cloud_mask(time, level)*

cloud_fraction

Type *double*

Dimensions *time, level*

Variable_Type Optional

Description Fraction of cloud contamination

Units –

Definition *byte cloud_mask(time, altitude)*

temperature

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description Atmospheric temperature

Units K

Definition *double temperature(time, level)*

pressure

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description Atmospheric pressure

Units mbar

Definition *double pressure(time, level)*

molecular_calculation_source

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Source used to compute the atmospheric molecular density

Units –

Definition *byte molecular_calculation_source*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

range_corrected_signal_channel_id

Type *int*

Dimensions *channel, nc*

Variable_Type Optional

Description SCC channel ID list involved in the calculation of the range corrected signal total time-series

Units –

Definition *int range_corrected_signal_channel_id(channel, nc)*

range_corrected_signal_channel_name

Type *string*

Dimensions *channel*

Variable_Type Mandatory

Description Channel name for the range corrected signal timeseries

Units –

Definition *string range_corrected_signal_channel_name(channel)*

range_corrected_signal_emission_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Emission wavelength used to measure the range corrected signal timeseries

Units *nm*

Definition *double range_corrected_signal_emission_wavelength(channel)*

range_corrected_signal_detection_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Detection wavelength used to measure the range corrected signal timeseries

Units *nm*

Definition *double range_corrected_signal_detection_wavelength(channel)*

range_corrected_signal_range

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal range type

Units –

Definition *byte range_corrected_signal_range(channel)*

range_corrected_signal_scatterers

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal scatterer type

Units –

Definition *byte range_corrected_signal_scatterers(channel)*

range_corrected_signal_detection_mode

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify range corrected signal detection mode

Units –

Definition *byte range_corrected_signal_detection_mode(channel)*

polarization_crosstalk_parameter_g

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g(channel)*

polarization_crosstalk_parameter_g_statistical_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Statistical error on polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g_statistical_error(channel)*

polarization_crosstalk_parameter_g_systematic_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Systematic error on polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g_systematic_error(channel)*

polarization_crosstalk_parameter_h

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_h(channel)*

polarization_crosstalk_parameter_h_statistical_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Statistical error on polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_h_statistical_error(channel)*

polarization_crosstalk_parameter_h_systematic_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Systematic error on polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_H_systematic_error(channel)*

polarization_channel_geometry

Type *byte*

Dimensions *channel*

Variable_Type Optional

Description Polarization channel geometry

Units –

Definition *byte polarization_channel_geometry*

polarization_channel_configuration

Type *byte*

Dimensions *channel*

Variable_Type Optional

Description Polarization channel configuration

Units –

Definition *byte polarization_channel_configuration(channel)*

overlap_correction_function

Type *double*

Dimensions *channel,angle, level*

Variable_Type Mandatory

Description Overlap funciton used to correct the range corrected signal

Units –

Definition *double overlap_correction_function(channel, angle, level)*

assumed_particle_lidar_ratio

Type *double*

Dimensions *angle, level*

Variable_Type Optional

Description Assumed particle lidar ration for the elastic-only backscatter retrieval

Units *sr*

Definition *double assumed_particle_lidar_ratio(angle, level)*

assumed_particle_lidar_ratio_error

Type *double*

Dimensions *angle, level*

Variable_Type Optional

Description Total error (uncertainty) on assumed particle lidar ration for the elastic-only backscatter retrieval

Units *sr*

Definition *double assumed_particle_lidar_ratio(angle, level)*

molecular_extinction

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Calculated molecular extinction at emission wavelength

Units *m^{-1}*

Definition *double molecular_extinction(channel, time, level)*

molecular_transmissivity_at_emission_wavelength

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Calculate molecular transmissivity at emission wavelentgh

Units –

Definition *double molecular_transmissivity_at_emission_wavelength(channel, time, level)*

molecular_transmissivity_at_detection_wavelength

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Calculate molecular transmissivity at detection wavelentgh

Units –

Definition *double molecular_transmissivity_at_detection_wavelength(channel, time, level)*

molecular_lidar_ratio

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Calculated molecular lidar ratio at emission wavelength

Units *sr*

Definition *double molecular_lidar_ratio(channel)*

depolarization_calibration_index

Type *int*

Dimensions *channel*

Variable_Type Optional

Description Depolarization index to use to calibrate the channel

Units –

Definition *int depolarization_calibration_index(channel)*

polarization_calibration_type

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization calibration type

Units –

Definition *byte polarization_calibration_type(depolarization)*

molecular_depolarization_ratio

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Calculated molecular depolarization ratio

Units –

Definition *double molecular_depolarization_ratio(depolarization, time, level)*

near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_minimum(channel, time)*

near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_maximum(channel, time)*

ultra_near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_minimum(channel, time)*

ultra_near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Maximum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_maximum(channel, time)*

range_corrected_signal

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Range corrected signal timeseries

Units –

Definition *double range_corrected_signal(channel, time, level)*

range_corrected_signal_statistical_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Statistical error (uncertainty) on range corrected signal timeseries

Units –

Definition *double range_corrected_signal_statistical_error(channel, time, level)*

range_corrected_signal_systematic_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Systematic error (uncertainty) on range corrected signal timeseries

Units –

Definition *double range_corrected_signal_statistical_error(channel, time, level)*

polarization_gain_factor_measurementid

Type *string*

Dimensions *depolarization*

Variable_Type Optional

Description Measurement ID corresponding to the polarization measurements used to calibrate polarization channels

Units –

Definition *string polarization_gain_factor_measurementid(depolarization)*

polarization_gain_factor

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain factor (eta*)

Units –

Definition *double polarization_gain_factor(depolarization)*

polarization_gain_factor_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_statistical_error(depolarization)*

polarization_gain_factor_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_systematic_error(depolarization)*

polarization_gain_factor_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_start_datetime(depolarization)*

polarization_gain_factor_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_stop_datetime(depolarization)*

polarization_gain_factor_correction

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain correction factor (K)

Units –

Definition *double polarization_gain_factor_correction(depolarization)*

polarization_gain_factor_correction_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_statistical_error(depolarization)*

polarization_gain_factor_correction_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_systematic_error(depolarization)*

polarization_gain_factor_correction_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain corrrection factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_start_datetime(depolarization)*

polarization_gain_factor_correction_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain correction factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_stop_datetime(depolarization)*

ELPP Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

molecular_calculation_source_file

Type *string*

Global_Attribute_Type Optional

Description Name of the source file containing atmospheric PTU profile(s) used to compute molecular number density (if standard atmosphere is used this global attribute is missing)

ELDEC Products

This section describe the structure of ELDEC products.

ELDEC Products: dimensions

The following dimensions are defined in ELDEC products:

- time
- altitude
- ratio
- calibration
- nv

The dimension *time* (mandatory) specifies the number of calibration performed at different times.

The dimension *altitude* (mandatory) represents the number of rangebins characterizing the reflected to transmitted polarization calibration ratio.

The dimension *ratio* (mandatory) indicates the number reflected to transmitted ratios included in the product.

The dimension *calibration* indicates the number of calibrations provided by the product.

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

ELDEC Products: variables

latitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *double latitude*

longitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *double longitude*

station_altitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *double station_altitude*

altitude

Type *double*

Dimensions *altitude*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(altitude)*

range

Type *double*

Dimensions *altitude*

Variable_Type Mandatory

Description lidar range scale

Units m

Definition *double range(altitude)*

shots

Type *int*

Dimensions *time*

Variable_Type Mandatory

Description accumulated laser shots

Units –

Definition *int shots(time)*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

polarization_calibration_ratio_emission_wavelength

Type *double*

Dimensions *ratio*

Variable_Type Mandatory

Description Emission wavelength of the reflected to transmitted polarization calibration

Units *nm*

Definition *double polarization_calibration_ratio_emission_wavelength(ratio)*

polarization_calibration_ratio_detection_wavelength

Type *double*

Dimensions *ratio*

Variable_Type Mandatory

Description Detection wavelength of the reflected to transmitted polarization calibration

Units *nm*

Definition *double polarization_calibration_ratio_detection_wavelength(ratio)*

polarization_calibration_ratio_range

Type *byte*

Dimensions *ratio*

Variable_Type Mandatory

Description Bitmask to identify the reflected to transmitted polarization calibration ratio range type

Units –

Definition *byte polarization_calibration_ratio_range(ratio)*

polarization_calibration_ratio_configuration

Type *byte*

Dimensions *ratio*

Variable_Type Mandatory

Description Bitmask to identify the reflected to transmitted polarization calibration ratio configuration type

Units –

Definition *byte polarization_calibration_ratio_configuration(ratio)*

polarization_calibration_ratio

Type *double*

Dimensions *ratio, time, altitude*

Variable_Type Mandatory

Description Uncalibrated reflected to transmitted polarization calibration ratio

Units –

Definition *double polarization_calibration_ratio(ratio, time, altitude)*

polarization_calibration_ratio_statistical_error

Type *double*

Dimensions *ratio, time, altitude*

Variable_Type Mandatory

Description Statistical error (uncertainty) on uncalibrated reflected to transmitted polarization calibration ratio

Units –

Definition *double polarization_calibration_ratio_statistical_error(ratio, time, altitude)*

polarization_calibration_ratio_systematic_error

Type *double*

Dimensions *ratio, time, altitude*

Variable_Type Optional

Description Systematic error (uncertainty) on uncalibrated reflected to transmitted polarization calibration ratio

Units –

Definition *double polarization_calibration_ratio_statistical_error(ratio, time, altitude)*

polarization_calibration_minimum_range

Type *double*

Dimensions *ratio*

Variable_Type Mandatory

Description Minimum altitude to consider for polarization calibration

Units *m*

Definition *double polarization_calibration_minimum_range(ratio)*

polarization_calibration_maximum_range

Type *double*

Dimensions *ratio*

Variable_Type Mandatory

Description Maximum altitude to consider for polarization calibration

Units *m*

Definition *double polarization_calibration_maximum_range(ratio)*

polarization_calibration_ratio_average

Type *double*

Dimensions *ratio, time*

Variable_Type Mandatory

Description Average of the polarization calibration ratio within the calibration range

Units –

Definition *double polarization_calibration_ratio_average(ratio, time)*

polarization_calibration_ratio_average_statistical_error

Type *double*

Dimensions *ratio, time*

Variable_Type Mandatory

Description Statistical error (uncertainty) on the average of the polarization calibration ratio within the calibration range

Units –

Definition *double polarization_calibration_ratio_average_statistical_error(ratio, time)*

polarization_calibration_ratio_average_systematic_error

Type *double*

Dimensions *ratio, time*

Variable_Type Optional

Description Systematic error (uncertainty) on the average of the polarization calibration ratio within the calibration range

Units –

Definition *double polarization_calibration_ratio_average_systematic_error(ratio, time)*

polarization_gain_factor_wavelength

Type *double*

Dimensions *calibration*

Variable_Type Mandatory

Description Polarization gain factor wavelength

Units *nm*

Definition *double polarization_gain_factor_wavelength(calibration)*

polarization_gain_factor_range

Type *byte*

Dimensions *calibration*

Variable_Type Mandatory

Description Bitmask to describe the polarization gain factor range type

Units –

Definition *byte polarization_gain_factor_range(calibration)*

polarization_gain_factor

Type *double*

Dimensions *calibration, time*

Variable_Type Mandatory

Description Polarization gain factor (eta*)

Units –

Definition *double polarization_gain_factor(calibration, time)*

polarization_gain_factor_statistical_error

Type *double*

Dimensions *calibration, time*

Variable_Type Mandatory

Description Statistical error (uncertainty) on polarization gain factor (eta*)

Units –

Definition *double polarization_gain_factor_statistical_error(calibration, time)*

polarization_gain_factor_systematic_error

Type *double*

Dimensions *calibration, time*

Variable_Type Optional

Description Systematic error (uncertainty) on polarization gain factor (eta*)

Units –

Definition *double polarization_gain_factor_systematic_error(calibration, time)*

ELDEC Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

scc_product_ID

Type *string*

Global_Attribute_Type Optional

Description Numeric ID identifying the current product in SCC database

ELDA Products

This section describe the structure of ELDA products.

ELPP Products: dimensions

Aerosol optical products have the wavelength dimension (accommodating products at different wavelength in the same variable) and time dimension (accommodating different profiles in the same time series). The altitude dimension is defined for reporting the information as a function of the altitude. The dimension nv is used following the cf convention whenever a bound is needed for some evaluation. In our case nv is used for defining variables which have per definition some bounds as time_bounds reported below.

The following dimensions are defined in ELDA products:

- time
- altitude
- wavelength
- nv

The dimension *time* (mandatory) specifies the number of vertical profiles included in the product timeseries.

The dimension *altitude* (mandatory) represents the number of rangebins characterizing each vertical profile

The dimension *wavelength* (mandatory) indicates the number wavelengths for which the optical variables are provided

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

ELDA Products: variables

latitude

Type *float*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *float latitude*

longitude

Type *float*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *float longitude*

station_altitude

Type *float*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *float station_altitude*

altitude

Type *double*

Dimensions *altitude*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(altitude)*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

shots

Type *int*

Dimensions *time*

Variable_Type Mandatory

Description accumulated laser shots

Units –

Definition *int shots(time)*

cloud_mask_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bimask describing the cloudmask type

Units –

Definition *byte cloud_mask_type*

cloud_mask

Type *byte*

Dimensions *time, altitude*

Variable_Type Optional

Description Bimask describing the cloudmask

Units –

Definition *byte cloud_mask(time, altitude)*

vertical_resolution

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Mandatory

Description Effective vertical resolution according to Pappalardo et al., appl. opt. 2004

Units *m*

Definition *double vertical_resolution(wavelength, time, altitude)*

cirrus_contamination

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Do the profiles contain cirrus layers?

Units –

Definition *byte cirrus_contamination*

cirrus_contamination_source

Type *byte*

Dimensions –

Variable_Type Mandatory

Description How was cirrus_contamination obtained?

Units –

Definition *byte cirrus_contamination_source*

error_retrieval_method

Type *byte*

Dimensions *wavelength*

Variable_Type Mandatory

Description Method used for the retrieval of erros (uncertainties)

Units –

Definition *byte error_retrieval_method(wavelength)*

backscatter_evaluation_method

Type *byte*

Dimensions *wavelength*

Variable_Type Optional

Description Method used for the backscatter retrieval

Units –

Definition *byte backscatter_evaluation_method(wavelength)*

elastic_backscatter_algorithm

Type *byte*

Dimensions *wavelength*

Variable_Type Optional

Description Algorithm used for the retrieval of the elastic backscatter profile

Units –

Definition *byte elastic_backscatter_algorithm(wavelength)*

assumed_particle_lidar_ratio

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Assumed particle lidar ratio for the elastic-only backscatter retrieval

Units *sr*

Definition *double assumed_particle_lidar_ratio(wavelength, time, altitude)*

backscatter

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Particle backscatter coefficient

Units *1/(m*sr)*

Definition *double backscatter(wavelength, time, altitude)*

error_backscatter

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Absolute statistical error (unertanity) of particle backscatter coefficient

Units *1/(m*sr)*

Definition *double error_backscatter(wavelength, time, altitude)*

extinction

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Particle extinction coefficient

Units *1/m*

Definition *double extinction(wavelength, time, altitude)*

error_extinction

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Absolute statistical error (unertanity) of particle extinction coefficient

Units *1/m*

Definition *double error_extinction(wavelength, time, altitude)*

volumedepolarization

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Volume linear depolarization ratio

Units –

Definition *double volumedepolarization(wavelength, time, altitude)*

error_volumedepolarization

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Absolute statistical error (unertanity) of volume linear depolarization ratio

Units *1/m*

Definition *double error_volumedepolarization(wavelength, time, altitude)*

particledepolarization

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Particle linear depolarization ratio

Units –

Definition *double particledepolarization(wavelength, time, altitude)*

error_particledepolarization

Type *double*

Dimensions *wavelength, time, altitude*

Variable_Type Optional

Description Absolute statistical error (unertanity) of particle linear depolarization ratio

Units *1/m*

Definition *double error_particledepolarization(wavelength, time, altitude)*

user_defined_category

Type *int*

Dimensions –

Variable_Type Optional

Description User defined category of the measurement

Units –

Definition *int user_defined_category*

molecular_calculation_source

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Data source of the molecular calculations

Units –

Definition *byte molecular_calculation_source*

backscatter_calibration_value

Type *float*

Dimensions *wavelength*

Variable_Type Optional

Description Assumed backscatter-ratio value (unitless) in calibration range

Units –

Definition *float backscatter_calibration_value(wavelength)*

backscatter_calibration_search_range

Type *float*

Dimensions *wavelength, nv*

Variable_Type Optional

Description Height range wherein calibration range is searched

Units *m*

Definition *float backscatter_calibration_search_range(wavelength, nv)*

wavelength

Type *float*

Dimensions *wavelength*

Variable_Type Mandatory

Description Wavelength of the transmitted laser pulse

Units *nm*

Definition *float wavelength(wavelength)*

zenith_angle

Type *float*

Dimensions –

Variable_Type Mandatory

Description Laser pointing angle with respect to the zenith

Units *degrees*

Definition *float zenith_angle*

earlinet_product_type

Type *int*

Dimensions –

Variable_Type Mandatory

Description Earlinet product type

Units –

Definition *int earlinet_product_type*

backscatter_calibration_range_search_algorithm

Type *byte*

Dimensions *wavelength*

Variable_Type Optional

Description Algorithm used for the search of the calibration_range

Units –

Definition *byte backscatter_calibration_range_search_algorithm(wavelength)*

backscatter_calibration_range

Type *float*

Dimensions *wavelength, nv*

Variable_Type Optional

Description Height range where calibration was calculated

Units *m*

Definition *float backscatter_calibration_range(wavelength, nv)*

raman_backscatter_algorithm

Type *byte*

Dimensions *wavelength*

Variable_Type Optional

Description Algorithm used for the retrieval of the Raman backscatter profile

Units –

Definition *byte raman_backscatter_algorithm(wavelength)*

extinction_evaluation_algorithm

Type *byte*

Dimensions *wavelength*

Variable_Type Optional

Description Algorithm used for the extinction retrieval

Units –

Definition *byte extinction_evaluation_algorithm(wavelength)*

extinction_assumed_wavelength_dependence

Type *float*

Dimensions *wavelength*

Variable_Type Optional

Description Assumed wavelength dependence for particle extinction retrieval

Definition *float extinction_assumed_wavelength_dependence(wavelength)*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

ELDA Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

ELIC Products

This section describe the structure of ELIC products.

ELIC Products: dimensions

The following dimensions are defined in ELPP products:

- time
- level
- channel
- depolarization
- angle
- nv
- nc
- ncal

The dimension *time* (mandatory) specifies the number of RCS (for each channel) composing all the pre-processed time-series reported in the products.

The dimension *level* (mandatory) represents the number of rangebins characterizing the pre-processed RCS.

The dimension *channel* (mandatory) indicates the number channels at which the RCS time-series included in the product refer to.

The dimension *depolarization* (optional) indicates how many volume depolarization time-series are included in the products.

The dimension *angle* (mandatory) takes into account how many zenith scan angles have been used to measure the RCS time-series reported into the products.

The dimension *nv* (mandatory) is ancillary and it is always set to 2.

The dimension *nc* (optional) is ancillary and provides the number of channels involved in the calculation of RCS or volume depolarization time-series.

The dimension *ncal* (mandatory) is ancillary and provides the number of optical calibration used to calibrate the current product.

ELIC Products: variables

latitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site latitude

Units degrees_north

Definition *double latitude*

longitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description measurement site longitude

Units degrees_east

Definition *double longitude*

station_altitude

Type *double*

Dimensions –

Variable_Type Mandatory

Description station altitude above see level

Units m

Definition *double station_altitude*

altitude

Type *double*

Dimensions *time, level*

Variable_Type Mandatory

Description altitude above see level

Units m

Definition *double altitude(time, level)*

range

Type *double*

Dimensions *level*

Variable_Type Mandatory

Description lidar range scale

Units m

Definition *double range(level)*

laser_pointing_angle

Type *double*

Dimensions *angle*

Variable_Type Mandatory

Description laser pointing angle with respect to the zenith

Units degrees

Definition *double laser_pointing_angle(angle)*

laser_pointing_angle_of_profile

Type *int*

Dimensions *angle*

Variable_Type Mandatory

Description index of scan angle at which each single raw profile has been detected

Units –

Definition *int laser_pointing_angle_of_profile(angle)*

shots

Type *int*

Dimensions *time*

Variable_Type Mandatory

Description accumulated laser shots

Units –

Definition *int shots(time)*

time

Type *double*

Dimensions *time*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time(time)*

time_bounds

Type *double*

Dimensions *time, nv*

Variable_Type Mandatory

Description seconds since 1970-01-01T00:00:00Z

Units –

Definition *double time_bounds(time,nv)*

scc_product_type

Type *byte*

Dimensions –

Variable_Type Mandatory

Description Bitmask describing the SCC product type

Units –

Definition *byte scc_product_type*

attenuated_backscatter_channel_id

Type *int*

Dimensions *channel, nc*

Variable_Type Optional

Description SCC channel ID list involved in the calculation of the total attenuated backscatter time-series

Units –

Definition *int attenuated_backscatter_channel_id(channel, nc)*

attenuated_backscatter_channel_name

Type *string*

Dimensions *channel*

Variable_Type Mandatory

Description Channel name for the total attenuated backscatter timeseries

Units –

Definition *string attenuated_backscatter_channel_name(channel)*

attenuated_backscatter_emission_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Emission wavelength used to measure the total attenuated backscatter timeseries

Units *nm*

Definition *double attenuated_backscatter_emission_wavelength(channel)*

attenuated_backscatter_detection_wavelength

Type *double*

Dimensions *channel*

Variable_Type Mandatory

Description Detection wavelength used to measure the total attenuated backscatter timeseries

Units *nm*

Definition *double attenuated_backscatter_detection_wavelength(channel)*

attenuated_backscatter_range

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify total attenuated backscatter range type

Units –

Definition *byte attenuated_backscatter_range(channel)*

attenuated_backscatter_scatterers

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify total attenuated backscatter scatterer type

Units –

Definition *byte attenuated_backscatter_scatterers(channel)*

attenuated_backscatter_detection_mode

Type *byte*

Dimensions *channel*

Variable_Type Mandatory

Description Bitmask to identify total attenuated backscatter detection mode

Units –

Definition *byte attenuated_backscatter_detection_mode(channel)*

near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_minimum(channel, time)*

near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the near range and far range signals

Units *m*

Definition *double near_range_glueing_region_maximum(channel, time)*

ultra_near_range_glueing_region_minimum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mininum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_minimum(channel, time)*

ultra_near_range_glueing_region_maximum

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Maximum altitude of glueing region for the ultra near range and near range signals

Units *m*

Definition *double ultra_near_range_glueing_region_maximum(channel, time)*

attenuated_backscatter

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Total attenuated baskcatter timeseries

Units *1/(m*sr)*

Definition *double attenuated_backscatter(channel, time, level)*

attenuated_backscatter_statistical_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Mandatory

Description Statistical error (uncertainty) on total attenuated backscatter timeseries

Units *1/(m*sr)*

Definition *double attenuated_backscatter_statistical_error(channel, time, level)*

attenuated_backscatter_systematic_error

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Systematic error (uncertainty) on total attenuated backscatter timeseries

Units –

Definition *double attenuated_backscatter_systematic_error(channel, time, level)*

polarization_gain_factor_measurementid

Type *string*

Dimensions *depolarization*

Variable_Type Optional

Description Measurement ID corresponding to the polarization measurements used to calibrate polarization channels

Units –

Definition *string polarization_gain_factor_measurementid(depolarization)*

polarization_gain_factor

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain factor (eta*)

Units –

Definition *double polarization_gain_factor(depolarization)*

polarization_gain_factor_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_statistical_error(depolarization)*

polarization_gain_factor_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain factor

Units –

Definition *double polarization_gain_factor_systematic_error(depolarization)*

polarization_gain_factor_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_start_datetime(depolarization)*

polarization_gain_factor_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_stop_datetime(depolarization)*

polarization_gain_factor_correction

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Reflected to transmitted polarization channel gain correction factor (K)

Units –

Definition *double polarization_gain_factor_correction(depolarization)*

polarization_gain_factor_correction_statistical_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Statistical error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_statistical_error(depolarization)*

polarization_gain_factor_correction_systematic_error

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Systamatic error (uncertainty) on reflected to transmitted polarization channel gain correction factor

Units –

Definition *double polarization_gain_factor_correction_systematic_error(depolarization)*

polarization_gain_factor_correction_start_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain corrrection factor measurement start datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_start_datetime(depolarization)*

polarization_gain_factor_correction_stop_datetime

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization gain correction factor measurement stop datetime

Units seconds since 1970-01-01T00:00:00Z

Definition *double polarization_gain_factor_correction_stop_datetime(depolarization)*

cloud_mask_type

Type *byte*

Dimensions –

Variable_Type Optional

Description Bimask describing the cloudmask type

Units –

Definition *byte cloud_mask_type*

cloud_mask

Type *byte*

Dimensions *time, level*

Variable_Type Optional

Description Bimask describing the cloudmask

Units –

Definition *byte cloud_mask(time, level)*

cloud_fraction

Type *double*

Dimensions *time, level*

Variable_Type Optional

Description Fraction of cloud contamination

Units –

Definition *byte cloud_mask(time, altitude)*

temperature

Type *double*

Dimensions *time, level*

Variable_Type Optional

Description Atmospheric temperature

Units K

Definition *double temperature(time, level)*

pressure

Type *double*

Dimensions *time, level*

Variable_Type Optional

Description Atmospheric pressure

Units mbar

Definition *double pressure(time, level)*

molecular_calculation_source

Type *byte*

Dimensions –

Variable_Type Optional

Description Source used to compute the atmospheric molecular density

Units –

Definition *byte molecular_calculation_source*

polarization_crosstalk_parameter_g

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g(channel)*

polarization_crosstalk_parameter_g_statistical_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Statistical error on polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g_statistical_error(channel)*

polarization_crosstalk_parameter_g_systematic_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Systematic error on polarization crosstalk parameter (G)

Units –

Definition *double polarization_crosstalk_parameter_g_systematic_error(channel)*

polarization_crosstalk_parameter_h

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_h(channel)*

polarization_crosstalk_parameter_h_statistical_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Statistical error on polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_h_statistical_error(channel)*

polarization_crosstalk_parameter_h_systematic_error

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Systematic error on polarization crosstalk parameter (H)

Units –

Definition *double polarization_crosstalk_parameter_H_systematic_error(channel)*

polarization_channel_geometry

Type *byte*

Dimensions *channel*

Variable_Type Optional

Description Polarization channel geometry

Units –

Definition *byte polarization_channel_geometry*

polarization_channel_configuration

Type *byte*

Dimensions *channel*

Variable_Type Optional

Description Polarization channel configuration

Units –

Definition *byte polarization_channel_configuration(channel)*

overlap_correction_function

Type *double*

Dimensions *channel,angle, level*

Variable_Type Optional

Description Overlap funciton used to correct the range corrected signal

Units –

Definition *double overlap_correction_function(channel, angle, level)*

assumed_particle_lidar_ratio

Type *double*

Dimensions *angle, level*

Variable_Type Optional

Description Assumed particle lidar ration for the elastic-only backscatter retrieval

Units *sr*

Definition *double assumed_particle_lidar_ratio(angle, level)*

assumed_particle_lidar_ratio_error

Type *double*

Dimensions *angle, level*

Variable_Type Optional

Description Total error (uncertainty) on assumed particle lidar ration for the elastic-only backscatter retrieval

Units *sr*

Definition *double assumed_particle_lidar_ratio(angle, level)*

molecular_extinction

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Calculated molecular extinction at emission wavelength

Units *m^{-1}*

Definition *double molecular_extinction(channel, time, level)*

molecular_transmissivity_at_emission_wavelength

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Calculate molecular transmissivity at emission wavelentgh

Units –

Definition *double molecular_transmissivity_at_emission_wavelength(channel, time, level)*

molecular_transmissivity_at_detection_wavelength

Type *double*

Dimensions *channel, time, level*

Variable_Type Optional

Description Calculate molecular transmissivity at detection wavelentgh

Units –

Definition *double molecular_transmissivity_at_detection_wavelength(channel, time, level)*

molecular_lidar_ratio

Type *double*

Dimensions *channel*

Variable_Type Optional

Description Calculated molecular lidar ratio at emission wavelength

Units *sr*

Definition *double molecular_lidar_ratio(channel)*

depolarization_calibration_index

Type *int*

Dimensions *channel*

Variable_Type Optional

Description Depolarization index to use to calibrate the channel

Units –

Definition *int depolarization_calibration_index(channel)*

polarization_calibration_type

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Polarization calibration type

Units –

Definition *byte polarization_calibration_type(depolarization)*

molecular_depolarization_ratio

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Calculated molecular depolarization ratio

Units –

Definition *double molecular_depolarization_ratio(depolarization, time, level)*

atmospheric_background

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Mean atmospheric background calculated from lidar signal

Units *1/(m^3*sr)*

Definition *double atmospheric_background(channel, time)*

atmospheric_background_stdev

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Standard deviation of atmospheric background calculated from lidar signal

Units *1/(m^3*sr)*

Definition *double atmospheric_background_stdev(channel, time)*

atmospheric_background_sterr

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Standard error of atmospheric background calculated from lidar signal

Units *1/(m^3*sr)*

Definition *double atmospheric_background_sterr(channel, time)*

atmospheric_background_min

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Minimum atmospheric background calculated from lidar signal

Units *1/(m^3*sr)*

Definition *double atmospheric_background_min(channel, time)*

atmospheric_background_max

Type *double*

Dimensions *channel, time*

Variable_Type Optional

Description Maximum atmospheric background calculated from lidar signal

Units *1/(m^3*sr)*

Definition *double atmospheric_background_max(channel, time)*

volume_linear_depolarization_ratio_channel_id

Type *int*

Dimensions *depolarization, nc*

Variable_Type Optional

Description SCC channel ID list involved in the calculation of the volume linear depolarization ratio timeseries

Units –

Definition *int volume_linear_depolarization_ratio_channel_id(depolarization, nc)*

volume_linear_depolarization_ratio_channel_name

Type *string*

Dimensions *depolarization*

Variable_Type Optional

Description Channel name for volume linear depolarization ration timeseries

Units –

Definition *string volume_linear_depolarization_ratio_channel_name(depolarization)*

volume_linear_depolarization_ratio_wavelength

Type *double*

Dimensions *depolarization*

Variable_Type Optional

Description Wavelength at which the volume linear depolarization ratio is calculated

Units *nm*

Definition *double volume_linear_depolarization_ratio_wavelength(depolarization)*

volume_linear_depolarization_ratio_range

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Bitmask field indicating the volume linear depolarization range type

Units –

Definition *byte volume_linear_depolarization_ratio_range(depolarization)*

volume_linear_depolarization_ratio_scatterers

Type *byte*

Dimensions *depolarization*

Variable_Type Optional

Description Bitmask field indicating the volume linear depolarization scatteter type

Units –

Definition *byte volume_linear_depolarization_ratio_scatterers(depolarization)*

volume_linear_depolarization_ratio

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio(depolarization, time, level)*

volume_linear_depolarization_ratio_statistical_error

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Statistical error (uncertanity) on volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio_statistical_error(depolarization, time, level)*

volume_linear_depolarization_ratio_systematic_error

Type *double*

Dimensions *depolarization, time, level*

Variable_Type Optional

Description Systematic error (uncertanity) on volume linear depolarization ratio timeseries

Units –

Definition *double volume_linear_depolarization_ratio_systematic_error(depolarization, time, level)*

attenuated_backscatter_calibration

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Constant used to calibrate the total attenuated backscatter

Units –

Definition *double attenuated_backscatter_calibration(channel, time)*

attenuated_backscatter_calibration_statistical_error

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Statistical error (uncertainty) of the attenuated backscatter calibration

Units –

Definition *double attenuated_backscatter_calibration_statistical_error(channel, time)*

attenuated_backscatter_calibration_systematic_error

Type *double*

Dimensions *channel, time*

Variable_Type Mandatory

Description Systematic error (uncertainty) of the attenuated backscatter calibration

Units –

Definition *double attenuated_backscatter_calibration_sytematic_error(channel, time)*

attenuated_backscatter_calibration_start_datetime

Type *double*

Dimensions *channel, ncal*

Variable_Type Mandatory

Description Attenuated backscatter calibration measurement start datetime

Units *seconds since 1970-01-01T00:00:00Z*

Definition *double attenuated_backscatter_calibration_start_datetime(channel, ncal)*

attenuated_backscatter_calibration_stop_datetime

Type *double*

Dimensions *channel, ncal*

Variable_Type Mandatory

Description Attenuated backscatter calibration measurement stop datetime

Units *seconds since 1970-01-01T00:00:00Z*

Definition *double attenuated_backscatter_calibration_stop_datetime(channel, ncal)*

attenuated_backscatter_calibration_measurementid

Type *string*

Dimensions *channel, ncal*

Variable_Type Mandatory

Description Attenuated backscatter calibration measurementID

Units –

Definition *string attenuated_backscatter_calibration_measurementid(channel, ncal)*

attenuated_backscatter_calibration_id

Type *int*

Dimensions *channel, ncal*

Variable_Type Mandatory

Description Attenuated backscatter calibration ID

Units –

Definition *int attenuated_backscatter_calibration_id(channel, ncal)*

ELIC Products: global attributes

Conventions

Type *string*

Global_Attribute_Type Mandatory

Description List of the conventions the product is compliant to

title

Type *string*

Global_Attribute_Type Mandatory

Description Product title

source

Type *string*

Global_Attribute_Type Mandatory

Description String describing the source used to generate the product

references

Type *string*

Global_Attribute_Type Mandatory

Description Reference(s) for data processing

location

Type *string*

Global_Attribute_Type Mandatory

Description Measurement site location

station_ID

Type *string*

Global_Attribute_Type Mandatory

Description Unique 3 digit code identifying SCC station

PI

Type *string*

Global_Attribute_Type Mandatory

Description PI full name

PI_affiliation

Type *string*

Global_Attribute_Type Mandatory :Description: PI complete affiliation

PI_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description PI affiliation acronym

PI_address

Type *string*

Global_Attribute_Type Optional

Description PI address

PI_phone

Type *string*

Global_Attribute_Type Optional

Description PI phone number

PI_email

Type *string*

Global_Attribute_Type Mandatory

Description PI e-mail

Data_Originator

Type *string*

Global_Attribute_Type Mandatory

Description Data originator full name

Data_Originator_affiliation

Type *string*

Global_Attribute_Type Mandatory

Description Data originator complete affiliation

Data_Originator_affiliation_acronym

Type *string*

Global_Attribute_Type Mandatory

Description Data originator affiliation acronym

Data_Originator_address

Type *string*

Global_Attribute_Type Optional

Description Data originator complete address

Data_Originator_phone

Type *string*

Global_Attribute_Type Optional

Description Data originator phone number

Data_Originator_email

Type *string*

Global_Attribute_Type Mandatory

Description Data originator e-mail address

institution

Type *string*

Global_Attribute_Type Mandatory

Description Istitution the product belong to

system

Type *string*

Global_Attribute_Type Mandatory

Description Name of the measuring instrument (lidar)

hoi_system_ID

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

hoi_configuration_ID;

Type *int*

Global_Attribute_Type Mandatory

Description Numeric ID identifying the configuration of the measuring system in the ACTRIS/EARLINET Handbook Of Instrument (HOI)

measurement_ID;

Type *string*

Global_Attribute_Type Mandatory

Description Alphanumeric identifier for the raw data measurements use to generate the product

measurement_start_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measurement start datetime YYYY-mm-ddTHH:MM:SSZ

measurement_stop_datetime

Type *string*

Global_Attribute_Type Mandatory

Description Measruement stop datetime YYYY-mm-ddTHH:MM:SSZ

comment

Type *string*

Global_Attribute_Type Optional

Description Any product comments

scc_version_description

Type *string*

Global_Attribute_Type Mandatory

Description Extended SCC version description

scc_version

Type *string*

Global_Attribute_Type Mandatory

Description SCC version number

processor_name

Type *string*

Global_Attribute_Type Mandatory

Description Name of the processor used to generate the product

processor_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the processor used to generate the product

history

Type *string*

Global_Attribute_Type Mandatory

Description Field describing the history of the product

__file_format_version

Type *string*

Global_Attribute_Type Mandatory

Description Version of the product file format

data_processing_institution

Type *string*

Global_Attribute_Type Mandatory

Description Instituion responsible for data processing

input_file

Type *string*

Global_Attribute_Type Mandatory

Description Filename of the input data product used to generate the current product

molecular_calculation_source_file

Type *string*

Global_Attribute_Type Optional

Description Name of the source file containing atmospheric PTU profile(s) used to compute molecular number density (if standard atmosphere is used this global attribute is missing)

SCC Bitmask/flag description

In this section a description of all the SCC bitmask or flag fields is provided.

Bitmask fields

Table A1: SCC product type bitmask description

Bit	Description
1st	experimental
2nd	operational

Table A2: Range type bitmask description

Bit	Description
1st	ultra_near_range
2nd	near_range
3rd	far_range

Table A3: Scatteters type bitmask description

Bit	Description
1st	particle
2nd	nitrogen
3rd	oxygen
4th	water_vapour

Table A4: Detection mode bitmask description

Bit	Description
1st	analog
2nd	photoncountin

Table A5: Cloudmask bitmask description

Bit	Description
1st	unknown cloud
2nd	cirrus
3rd	mixed-phase cloud

Table A6: Cloudmask type bitmask description

Bit	Description
1st	manual cloudmask
2nd	automatic cloudmask

Table A.7: Polarization channel geometry bitmask description

Bit	Description
1st	transmitted signal
2nd	reflected signal

Table A.8: Polarization channel configuration bitmask description

Bit	Description
1st	0 degrees
2nd	90 degrees
3rd	+45 degree
4th	-45 degrees

Table A.9: Polarization calibration type bitmask description

Bit	Description
1st	automatic
2nd	manual

Table A.10: User defind category bitmask description

Bit	Description
1st	cirrus
2nd	climatol
3rd	dicucles
4th	volcanic
5th	forfires
6th	photosmog
7th	rurban
8th	sahadust
9th	stratos
10th | satallite_overpasses	

Flag fields

Table B.1: Molecular calculation source flag description

Value	Description
0	US standard atmosphere
1	Radiosounding
2	ECMWF forecast
3	ICON-IGLO-12-13 forecast
4	ICON-IGLO-24-25 forecast
5	ICON-IGLO-36-47 forecast
6	GDAS forecast
7	ERA5-1-12
8	ERA5-7-18

Table B.2: Cirrus contamination flag description

Value	Description
0	not available
1	no cirrus
2	cirrus detecte

Table B.3: Cirrus contamination source flag description

Value	Description
0	not available
1	user provid
2	automatic calculate

Table B.4: Error retrieval method flag description

Value	Description
0	Monte Carlo
1	Standard error propagation
2	automatic calculate

Table B.5: Backscatter evaluation method flag description

Value	Description
0	Raman
1	Elastic

Table B.6: Elastic backscatter algorithm flag description

Value	Description
0	Klett-Fernald
1	Iterative

Table B.7: Earlinet product type flag description

Value	Description
1	e0355
2	b0355
3	e0351
4	b0351
5	e0532
6	b0532
7	e1064
8	b1064
9	b0253
10	b0313
11	b0335
12	b0511
13	b0694
14	b0817

Table B.8: Backscatter calibration range search algorithm flag description

Value	Description
0	minimum of signal ratio
1	minimum of elastic signal
2	Rayleigh fit

Table B.9: Raman backscatter algorithm flag description

Value	Description
0	Ansmann method
1	via backscatter ratio

Table B.10: Extinction evaluation algorithm flag description

Value	Description
0	weighted linear fit
1	non weighted linear fi

CHAPTER 10

User management

10.1 Acount types

10.2 Requesting a new account

10.3 User account security

CHAPTER 11

FAQ

11.1 Frequently asked questions

11.1.1 Using an ancillary file

Q: I use the "Quick upload" to submit a measurement file together with an ancillary file. Will this ancillary file be used in the processing of my measurement.

A: No. Which ancillary file is used when processing a measurement is defined in the measurement netcdf file. If your new ancillary file is not mentioned in the measurement file, it will not be used.

11.1.2 Reusing an ancillary file

Q: I want to use one ancillary file (ex. an overlap file) in the processing of multiple measurements. Do I need to submit it multiple times?

A: No. You just need to define the file name of the file to use in the measurements netcdf file.

11.1.3 Deleting an ancillary file

Q: I want to delete an uploaded ancillary file but in the "Admin" interface I can't find a "Delete" button.

A: Probably the ancillary file you are trying to delete is needed by some uploaded measurement, and for this reason you are not allowed to delete it. You will need to delete the corresponding measurements first, before deleting the ancillary file.

11.1.4 Clouds in the data

Q: Is it necessary to provide only measurement periods with cloud free conditions and homogeneous atmosphere or is this part of ELDA or the pre-processing of the data? For example in cases with scattered low cumulus clouds.

A: At moment you should provide cloud free data because the automatic cloud screening is not yet implemented in the SCC. We are working on this issue and hopefully the new module will be implemented at the end of ACTRIS project.

11.1.5 Minimum number of analog files to submit

Q: Is it possible to submit a single analog profile for processing?

A: It depepends on the type of product you ar trying to calculate.

- If the product is a linear polarization calibration, it is possible to submit a timeseries containing one single analog profile. This is allowed because some stations are performing depolarization calibration measurements using single profiles. If the errors are present they will be taken into account for the calculation of the error on calibration constant. If they are not the error on calibration constant is calculate as the standard deviation withing the calibration range.
- If the product to calculate is NOT a linear polarization calibration, if the raw analog error have not been provided we need at least 3 raw profile in order to compute the statistical error

11.1.6 High/low range channels

Q: What is the definition of low range, ultra near range and high range channels? Are there any threshold values?

A: There are no threshold values defined for the different range types. This information is used only in the gluing procedures just to identify which channel should be taken as low range and which one as far range. If no gluing is applied by the SCC the range id flags are not taken into account.

11.1.7 Extra netcdf parameters

Q: Is it possible for documentation purposes to put own parameters in the SCC-NetCDF file? For example who created the file. . . . Are there any reasons against this?

A: Technically as long as you use not standard SCC variables for your own parameters there are no problems for the SCC. It will just ignore these not standard variables.

11.1.8 Netcdf version

Q: Which NetCDF version is to use? NetCDF3, NetCDF3 Classic, NetCDF4, NetCDF4 Classic?

A: The NetCDF libraries 4.1.3 are used in all the SCC modules. So all the NetCDF formats you have indicated should be compatible with the SCC (we have tested NetCDF3 and NetCDF4).

11.1.9 Lidar ratio

Q: What are the values for the lidar ratio used in the SCC_DB?

A: The values of (fixed) lidar ratio used by the SCC in the elastic retrieval can be set by the user using the SCC web interface. In particular you can define a lidar ratio value for each elastic backscatter product: in the product page there is the section "Elastic Backscatter options" in which there is the field "Fixed lr". In case you want to use a lidar ratio profile you should set LR_Input accordingly and provide an external LR profile NetCDF file (see documentation on SCC file format).

11.1.10 Calculation of Raman and elastic backscatter

Q: In cases of measurements where Raman channels are available, the SCC will calculate the Raman backscatter profile. If I want to retrieve Klett-retrievals for this channel, too (e.g 532nm) is it sufficient to set the value in LR_input(channels) to 1 or 0 plus a LR-profile to get both retrievals?

A: No. In general, for each lidar configuration you can define a set of optical products to be calculated for that configuration using the SCC web interface. So suppose you have a system with 532nm and 607nm channels. In this case you have 2 options:

1. Raman backscatter and extinction in the e532 file and Raman backscatter (full resolution) in the b532 file. In this case you should associate to the configuration a product of type "lidar ratio and will produce the e532 file and a product of type "Raman backscatter" which will produce the b532 file
2. Raman backscatter and extinction in the e532 file and elastic backscatter in the b532 file. In this case you should associate to the configuration a product of type "lidar ratio and extinction" which will produce the e532 file and a product of type "elastic backscatter" which will produce the b532 file

Note: you cannot calculate a b532 file containing the Raman and elastic backscatters at the same time. The reason is that the 2 products will produce an output file with the same name (according to the EARLINET rules). Moreover in general, it makes no sense to calculate the elastic backscatter when you can calculate the Raman one which usually is better.

11.1.11 Filename conventions

Q: What are the conventions for the filenames for the various files that need to be uploaded?

A: The following definitions apply:

SCC raw lidar data file

In the current version of the SCC there is not limit in the name of the raw data file. We suggest, however, that this file is named <measurement_id>.nc. For example, if your measurement had a measurement ID of 20130101cc00 the corresponding NetCDF file should be named 20130101cc00.nc

Sounding file

The file should be named as rs_measID.nc. Considering the above example the sounding file should be named rs_20130101cc00.nc

In this case you should also set the global attribute Sounding_File_Name in the raw lidar data file as:

```
Sounding_File_Name=rs_20130101cc00.nc
```

Lidar ratio file

The file should be named as lr_measID.nc. Considering the above example the sounding file should be named lr_20130101cc00.nc

In this case you should also set the global attribute LR_File_Name in the raw lidar data file as:

```
LR_File_Name=lr_20130101cc00.nc
```

Overlap file

The file should be named as ov_measID.nc. Considering the above example the sounding file should be named ov_20130101cc00.nc

In this case you should also set the global attribute Overlap_File_Name in the raw lidar data file as:

```
Overlap_File_Name=ov_20130101cc00.nc
```

11.1.12 Photocounting values should be integers

Q: In one of my measurements I get an error concerning the photoncounting values:

```
Pre processing (177): Found no integer values in photoncounting signal
```

The Raw_Lidar_Data variable in the NetCDF-file is defined as double. So is it necessary for Photoncounting signals to only provide integer values?

A: Two important considerations:

1. The Raw_Lidar_Data array should contain your *real* raw data. This means that *no corrections/operations* should be made on your signals before filling the Raw_Lidar_Data array. This is particularly important because *all* the operations and corrections should be applied by the SCC and not by the user before the submission. In this way we can keep track of all the operations made on the signals (for QA purposes) and moreover we are sure that all the corrections are applied in a correct order (this is particularly important for non linear operations, think for example to the dead time correction).
2. The analog signals should be expressed in mV and the photoncounting signals in raw counts

So if your photoncounting values are not integers they are not expressed in raw counts (which of course should be integers). So the point here is not how to convert them in integers but to submit the right quantity in the right units.

So please check carefully your converter and be sure to really submit raw counts for photoncounting channels and raw mV for the analog ones.

11.1.13 Preprocessing failed but no Exit code is provided

Q: The preprocessing of one of my measurements failed (I get a status -127). However, when I check the Exit codes to see the description of the problem, I get an empty value (-). What does this mean?

A: This means that the preprocessor crushed unexpectedly! Sorry for that! Report the problem in the forum and it will be fixed soon.

11.1.14 Force cloud-free conditions

Q: The SCC misclassifies dense aerosol layers contents as cloud. What can I do if I'm sure the corresponding measurement is not affected by clouds.

A: It is possible to set as cloud-free any measurement submitted to the SCC. If this is done the automatic cloud mask will be always produced and stored in the corresponding CloudScreen product but it will be in the further SCC data processing. In this case, the corresponding SCC products requiring as input the cloud mask will be labelled as manual cloud-screened. There are three equivalent ways to set as cloud-free an already submitted (and processed) measurements:

1. From "Data processing", click on the measurementID corresponding to the measurement you want to set as cloud-free, then clock on "Edit in Admin" and finally check "Manually Assumed Cloud Free"

2. From “Admin”, click on “Measurements”, click on the measurementID corresponding to the measurement you want to set as cloud-free an finally check “Manually Assumed Cloud Free”
3. From “Admin”, click on “Measurements”, select one or more measurments you want to set as cloud-free, select “Manually Assumed Cloud Free/Not Cloud Free and Reprocess selected measurements” in the action list located at the page bottom and finally click on “Go”.

In all the cases the processing on the corresponding measurementID(s) will be restarted automatically.

CHAPTER 12

Indices and tables

- genindex
- search