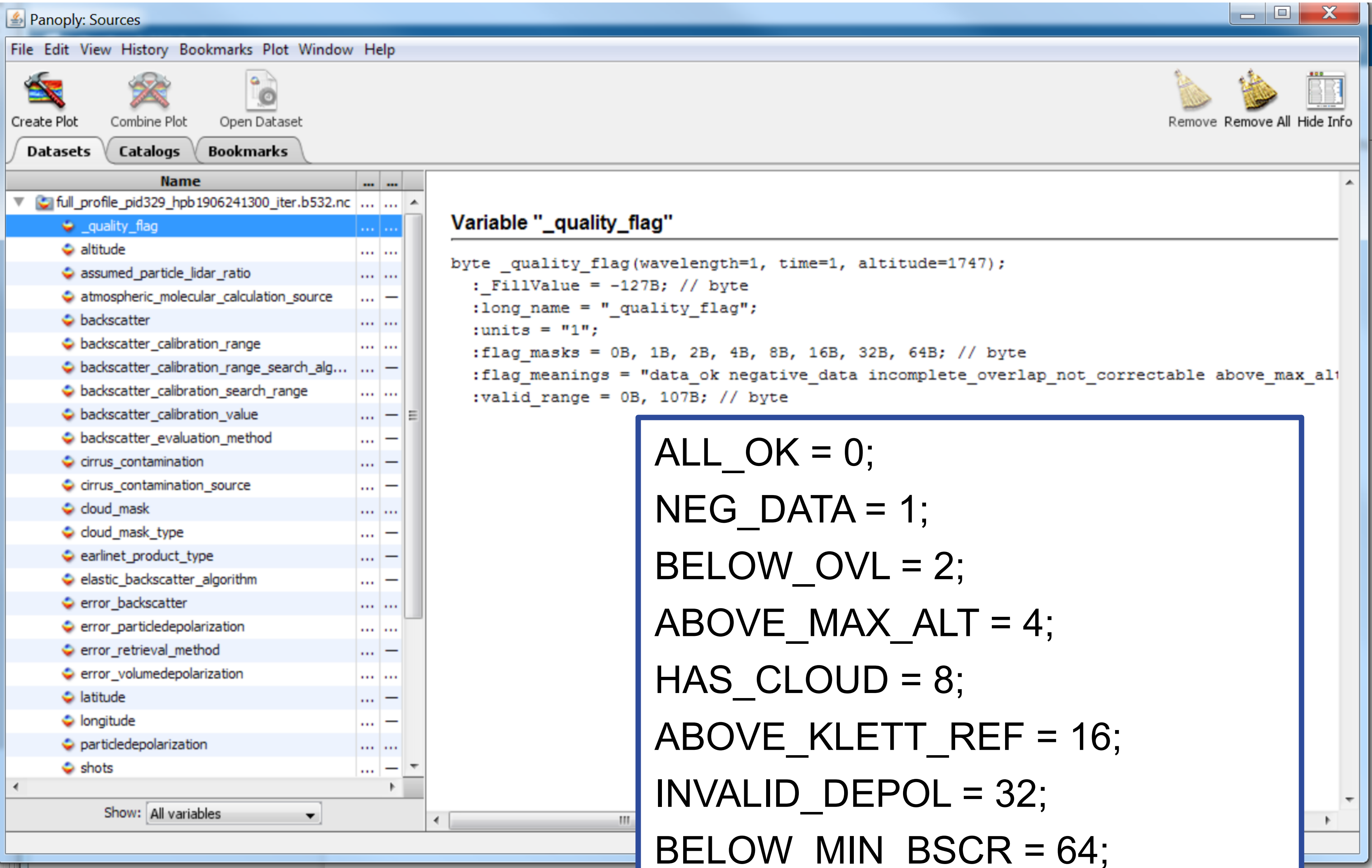
status ELDA: last activities

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

- ➔ added quality flags
  - qf handling throughout the code
  - new, extended output file (in addition to the normal one)
  - tested on hpb data
  - feature requested by Doina, but not yet support for testing or response
- ➔ document on class structure (preliminary)
- ➔ more documentation within the code

ELDA quality flags
(why data points might be missing)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand



Panoply: Sources
File Edit View History Bookmarks Plot Window Help
Create Plot
Combine Plot
Open Dataset
Remove
Remove All
Hide Info
Datasets
Catalogs
Bookmarks
Name
full_profile_pid329_hpb1906241300_iter.b532.nc
_quality_flag
altitude
assumed_particle_lidar_ratio
atmospheric_molecular_calculation_source
backscatter
backscatter_calibration_range
backscatter_calibration_range_search_alg...
backscatter_calibration_search_range
backscatter_calibration_value
backscatter_evaluation_method
cirrus_contamination
cirrus_contamination_source
cloud_mask
cloud_mask_type
earlinet_product_type
elastic_backscatter_algorithm
error_backscatter
error_particledepolarization
error_retrieval_method
error_volumedepolarization
latitude
longitude
particledepolarization
shots
Show: All variables

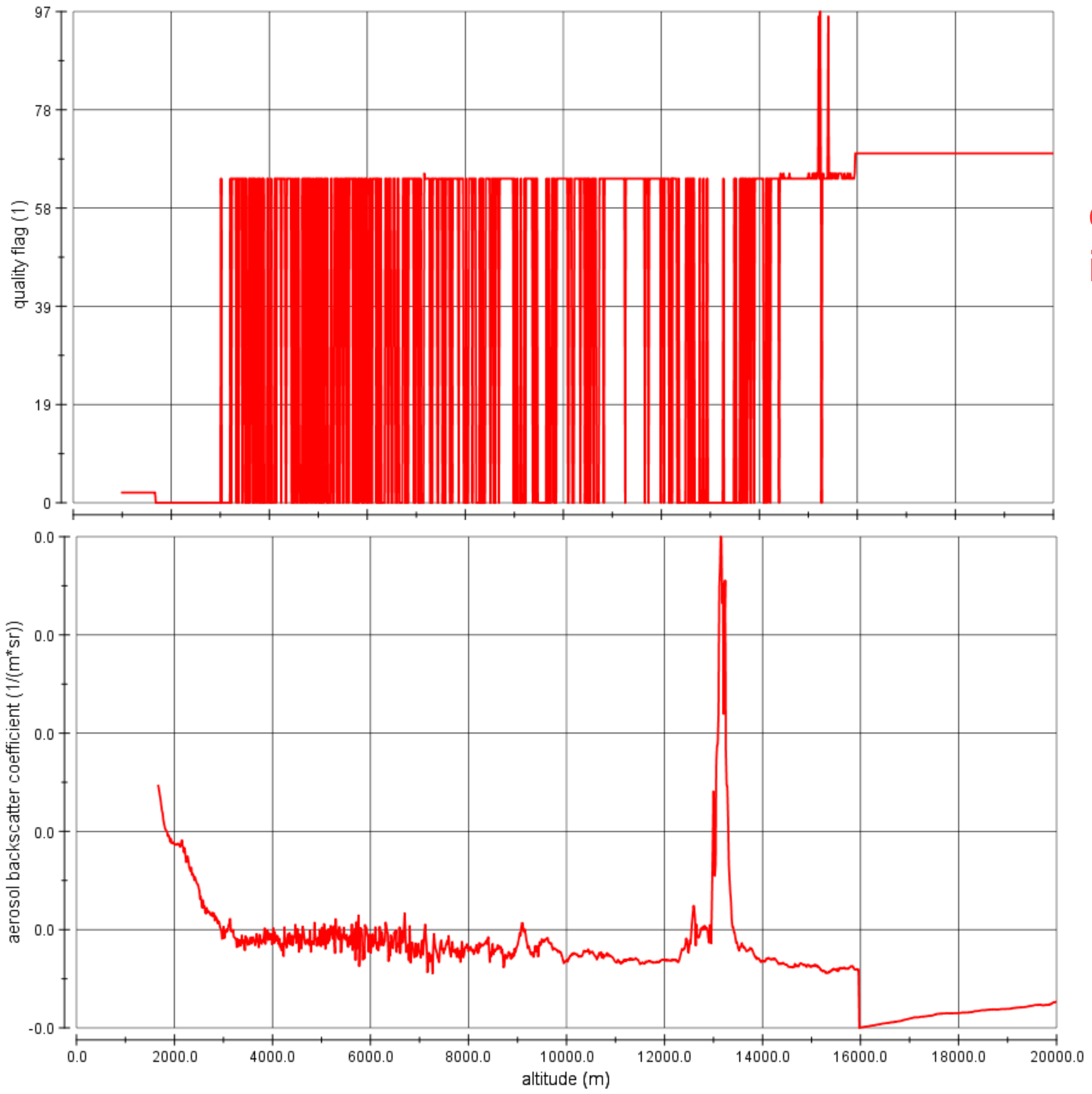
Variable "_quality_flag"

```
byte _quality_flag(wavelength=1, time=1, altitude=1747);
  :_FillValue = -127B; // byte
  :long_name = "_quality_flag";
  :units = "1";
  :flag_masks = 0B, 1B, 2B, 4B, 8B, 16B, 32B, 64B; // byte
  :flag_meanings = "data_ok negative_data incomplete_overlap_not_correctable above_max_alt
  :valid_range = 0B, 107B; // byte
```

ALL_OK = 0;
NEG_DATA = 1;
BELOW_OVL = 2;
ABOVE_MAX_ALT = 4;
HAS_CLOUD = 8;
ABOVE_KLETT_REF = 16;
INVALID_DEPOL = 32;
BELOW_MIN_BSCR = 64;

ELDA quality flags
(why data points might be missing)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

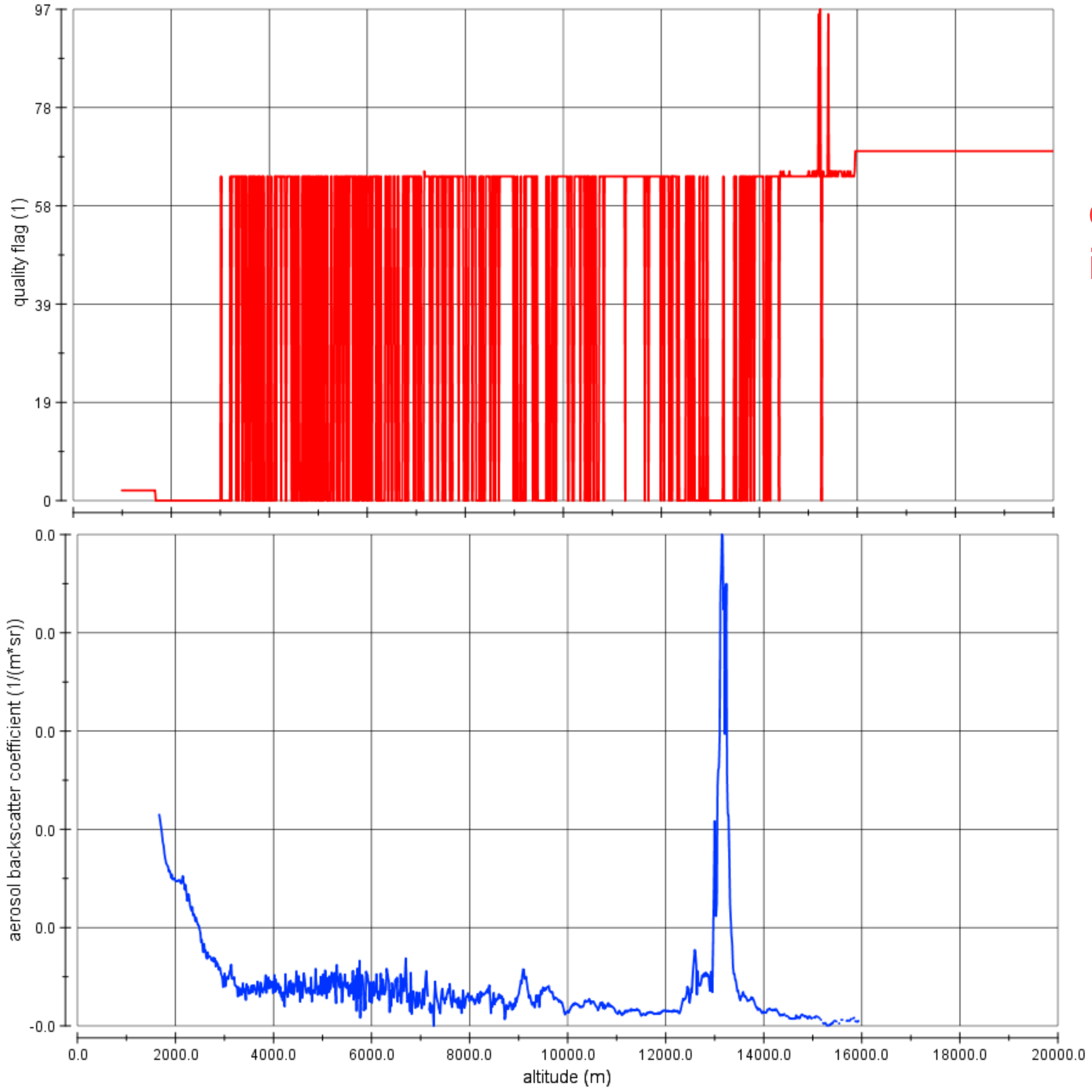


qf profile
in extended file

full profile
(including 'bad' data points)
in extended file

ELDA quality flags
(why data points might be missing)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand



qf profile
in extended file

full profile
(including 'bad' data points)
in extended file

screened profile
(only 'good' data points)
in standard file

status ELDA: next tasks

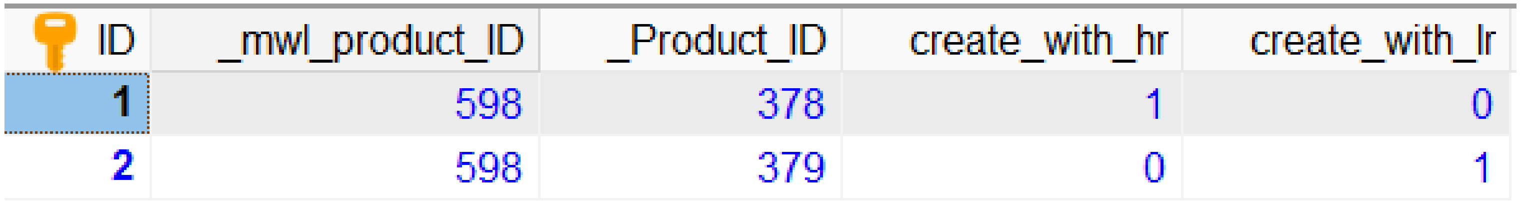
- ➔ test quality flag output (bu!)
- ➔ finish document on class structure
- ➔ prepare procedure for implementation of new Rayleigh calibration code

status ELDAmwl

- repository moved to Potenza
- implemented functionality for
  - check code formatting (pep8 & pycodestyle)
  - testing (good coverage for all db related code)
  - source code documentation (sphinx documentation generator)
- general design for modularity is implemented (see examples)
- current work:
  - filling the skeleton with ‘real’ code -> design adjustments necessary
  - code documentation whenever a class or procedure is changed
- goal:
  - retrieval of first ext, bsc, lr products before Easter
  - with this milestone, the general structure should be ready for other developers to contribute

New / modified db tables
mwlproduct_product

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

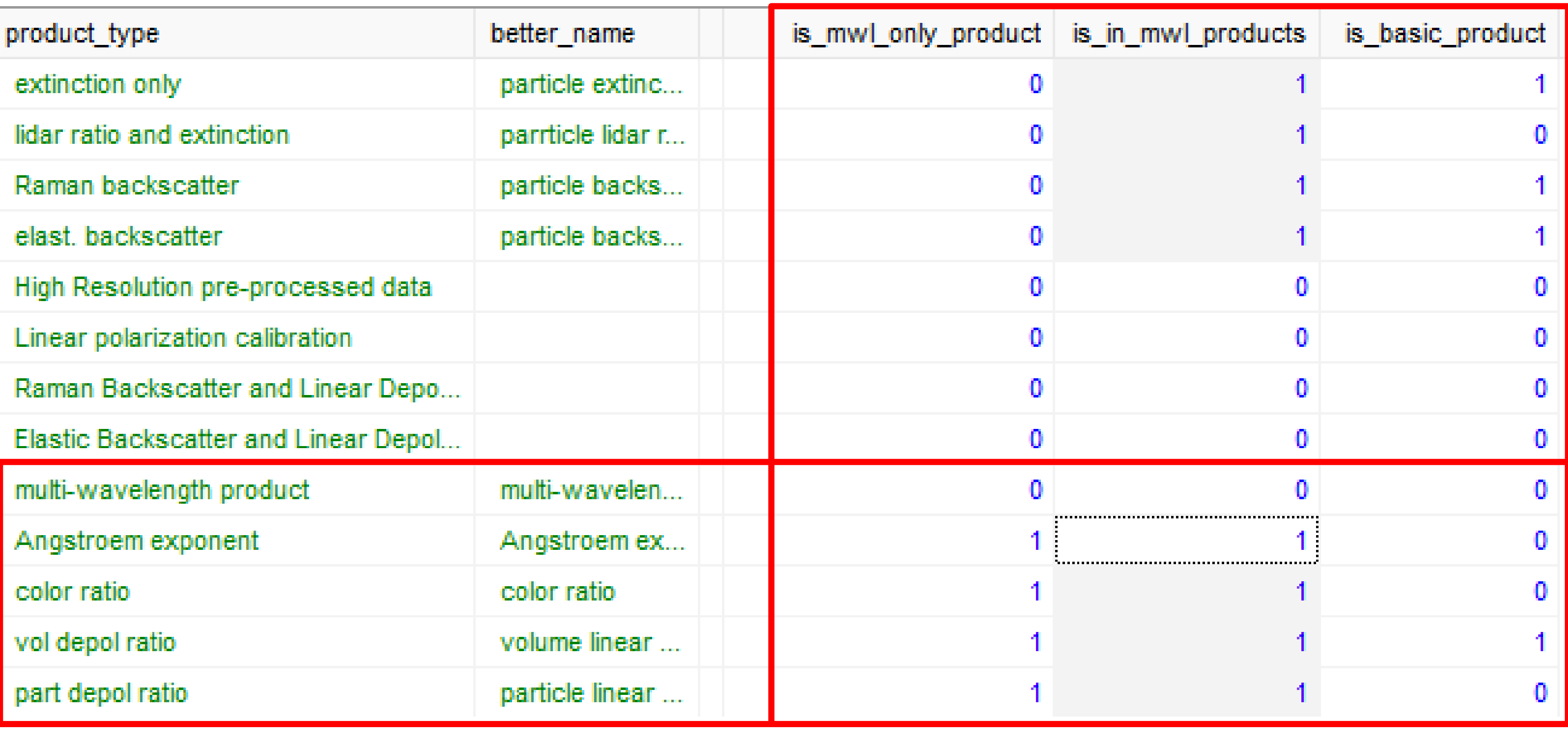


ID	_mwl_product_ID	_Product_ID	create_with_hr	create_with_lr
1	598	378	1	0
2	598	379	0	1

→ **requires new view in interface?**

New / modified db tables
_product_types

Deutscher Wetterdienst
Wetter und Klima aus einer Hand



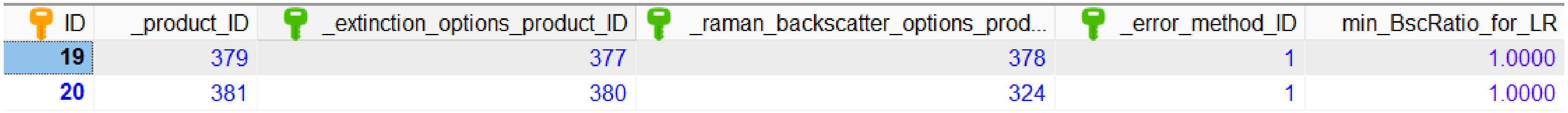
product_type	better_name		is_mwl_only_product	is_in_mwl_products	is_basic_product
extinction only	particle extinc...		0	1	1
lidar ratio and extinction	parrticle lidar r...		0	1	0
Raman backscatter	particle backs...		0	1	1
elast. backscatter	particle backs...		0	1	1
High Resolution pre-processed data			0	0	0
Linear polarization calibration			0	0	0
Raman Backscatter and Linear Depo...			0	0	0
Elastic Backscatter and Linear Depol...			0	0	0
multi-wavelength product	multi-wavelen...		0	0	0
Angstroem exponent	Angstroem ex...		1	1	0
color ratio	color ratio		1	1	0
vol depol ratio	volume linear ...		1	1	1
part depol ratio	particle linear ...		1	1	0

→ requires no changes in other modules

New / modified db tables
Options of new products

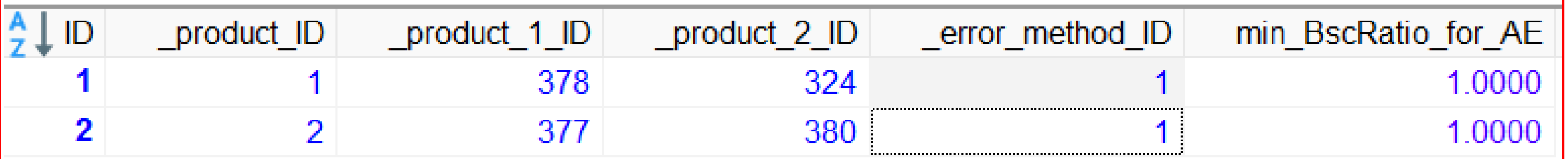
Deutscher Wetterdienst
Wetter und Klima aus einer Hand

ext_bsc_options



ID	_product_ID	_extinction_options_product_ID	_raman_backscatter_options_prod...	_error_method_ID	min_BscRatio_for_LR
19	379	377	378	1	1.0000
20	381	380	324	1	1.0000

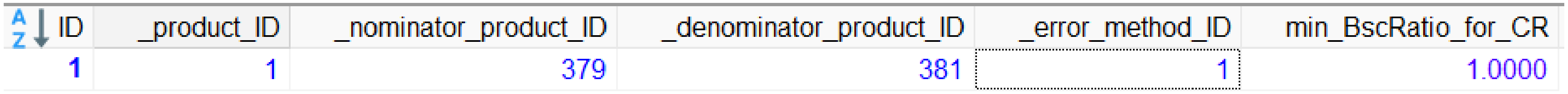
angstroem_exp_options (e.g. RBsc355 & RBsc532, Ext355 & Ext 532)



ID	_product_ID	_product_1_ID	_product_2_ID	_error_method_ID	min_BscRatio_for_AE
1	1	378	324	1	1.0000
2	2	377	380	1	1.0000

→ requires new view in interface ?

color_ratio_options (e.g. LR 355 & LR 532)

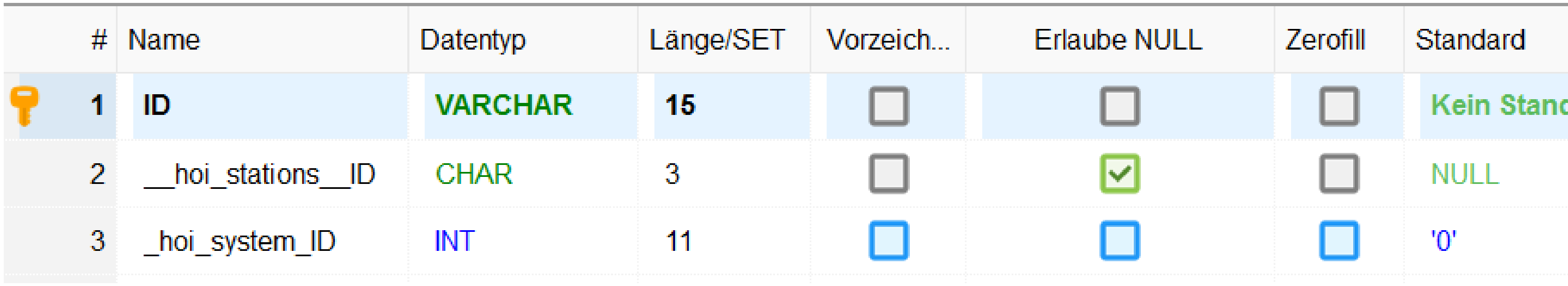


ID	_product_ID	_nominator_product_ID	_denominator_product_ID	_error_method_ID	min_BscRatio_for_CR
1	1	379	381	1	1.0000

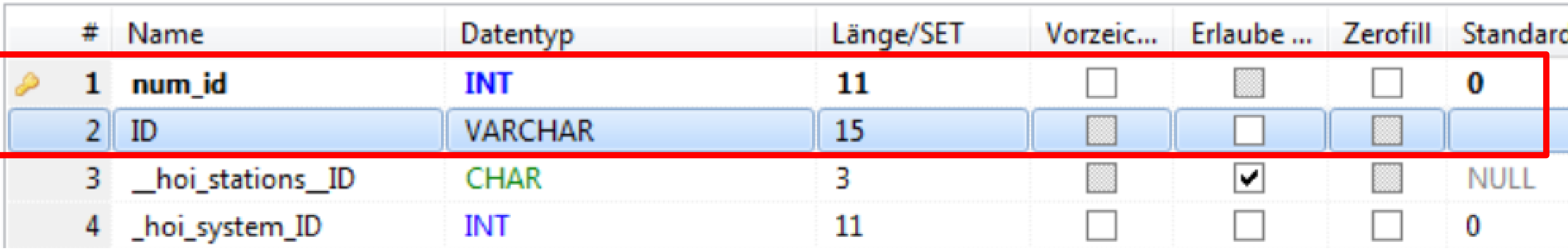
→ requires new view in interface ?

New / modified db tables
measurements

Deutscher Wetterdienst
Wetter und Klima aus einer Hand



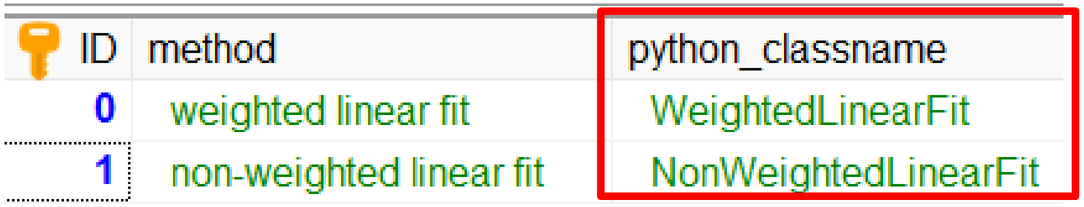
#	Name	Datentyp	Länge/SET	Vorzeich...	Erlaube NULL	Zerofill	Standard
1	ID	VARCHAR	15	☐	☐	☐	Kein Stan
2	__hoi_stations__ID	CHAR	3	☐	✓	☐	NULL
3	_hoi_system_ID	INT	11	☐	☐	☐	'0'



#	Name	Datentyp	Länge/SET	Vorzeic...	Erlaube ...	Zerofill	Standar
1	num_id	INT	11	☐	▨	☐	0
2	ID	VARCHAR	15	▨	☐	▨	
3	_hoi_stations_ID	CHAR	3	▨	✓	▨	NULL
4	_hoi_system_ID	INT	11	☐	☐	☐	0

→ should not require changes in other modules, but needs to be tested !!!
(but we already plan to clean up the db structure)

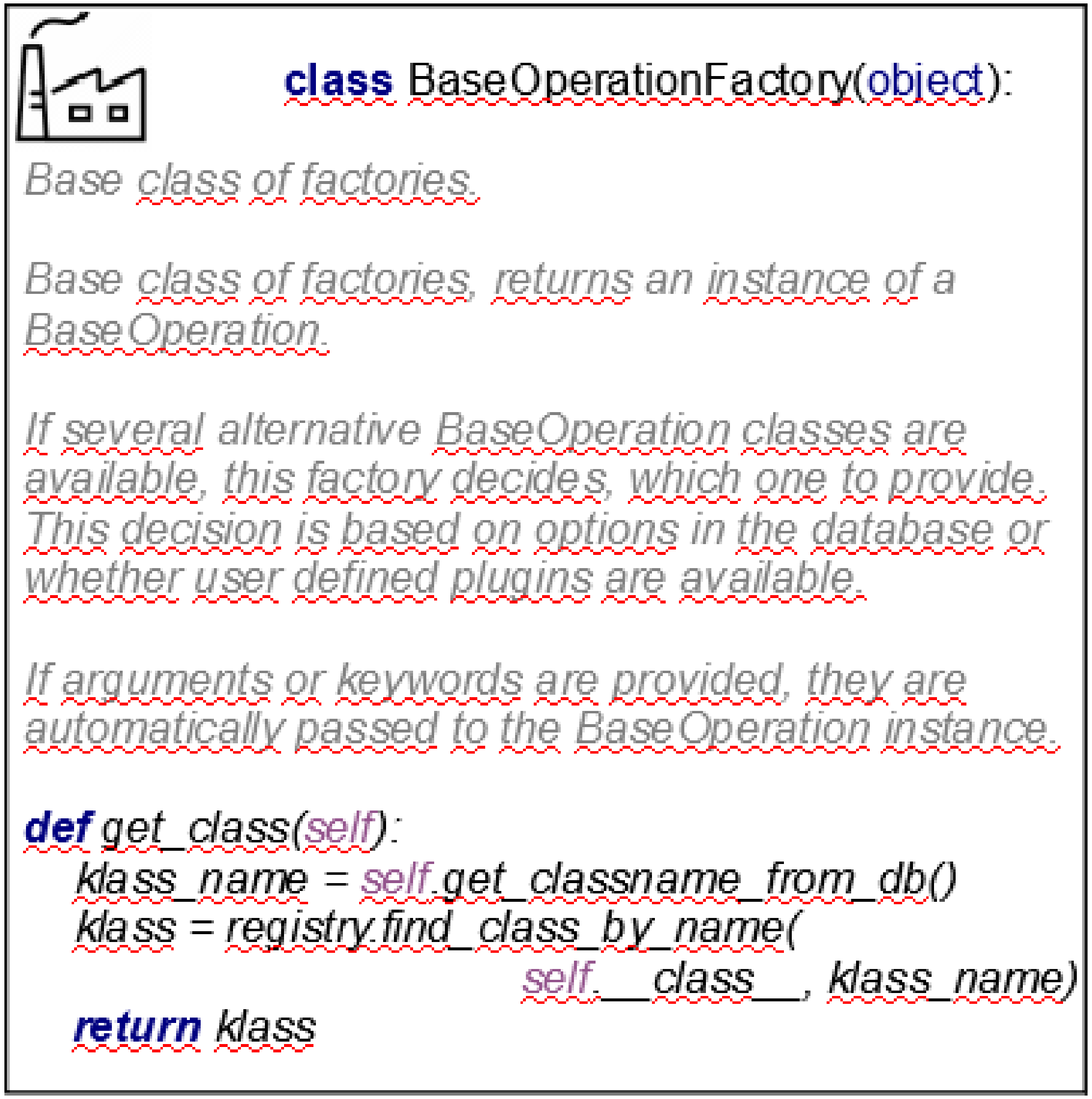
New / modified db tables
tables with method definitions



ID	method	python_classname
0	weighted linear fit	WeightedLinearFit
1	non-weighted linear fit	NonWeightedLinearFit

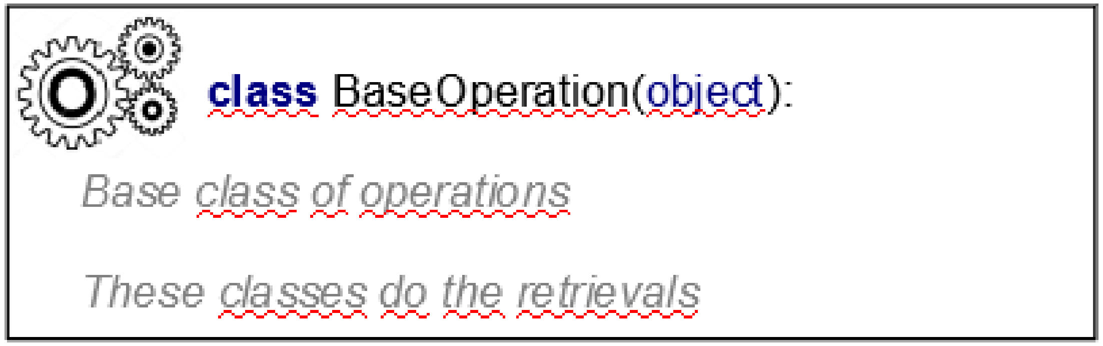
→ requires no changes in other modules

Modular design
base classes



```
class BaseOperationFactory(object):
Base class of factories.

Base class of factories, returns an instance of a
BaseOperation.

If several alternative BaseOperation classes are
available, this factory decides, which one to provide.
This decision is based on options in the database or
whether user defined plugins are available.

If arguments or keywords are provided, they are
automatically passed to the BaseOperation instance.

def get_class(self):
    klass_name = self.get_classname_from_db()
    klass = registry.find_class_by_name(
                         self.__class__, klass_name)
    return klass
```

Modular design
base classes



```
class BaseOperation(object):

    Base class of operations

    These classes do the retrievals
```

Modular design
Example: if there is only 1 method

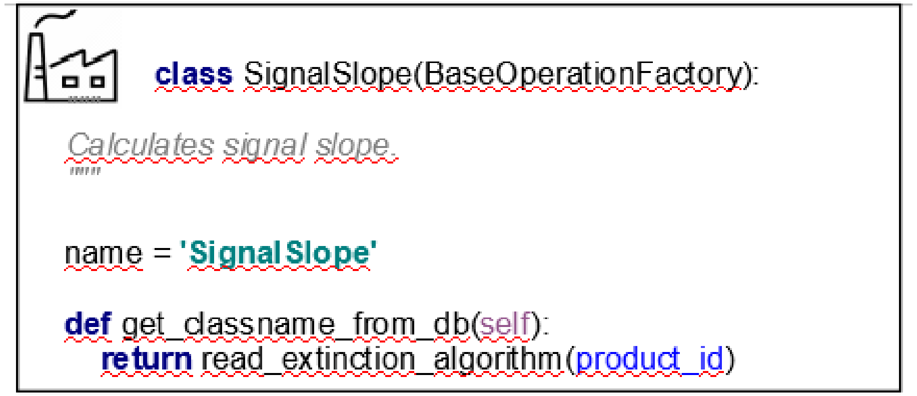
Deutscher Wetterdienst
Wetter und Klima aus einer Hand

```
class SlopeToExtinction(BaseOperationFactory):
    Calculates particle extinction coefficient from signal slope.

    name = 'SlopeToExtinction'

    def get_classname_from_db(self):
        """
        return: always 'getSlopeToExtinction' .
        """
        return 'getSlopeToExtinction'
```

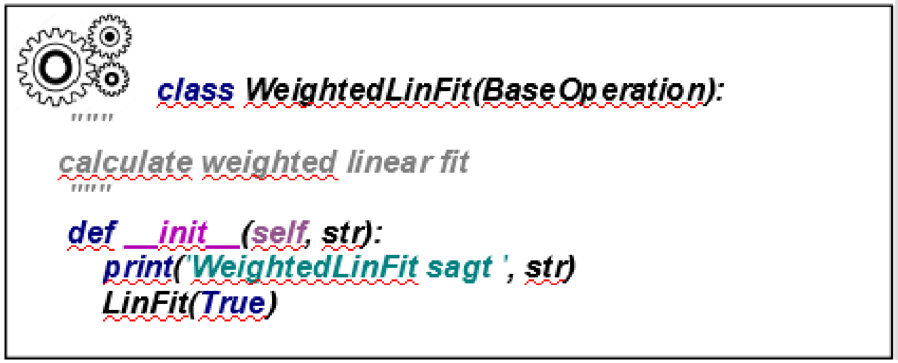
```
class getSlopeToExtinction(BaseOperation):
    Calculates particle extinction coefficient from signal slope.
    """
        WFA = 1.0 + power((DetectionWL / EmissionWL)…
        WFAinv = 1/WFA

        for bin in range (firstBin, lastBin):
            if valid[bin] :
                Data[bin] = Data[bin] * WFAinv
                Err[bin] = Err[bin] * WFAinv
```

Modular design
Example: add a new method (Extinction)

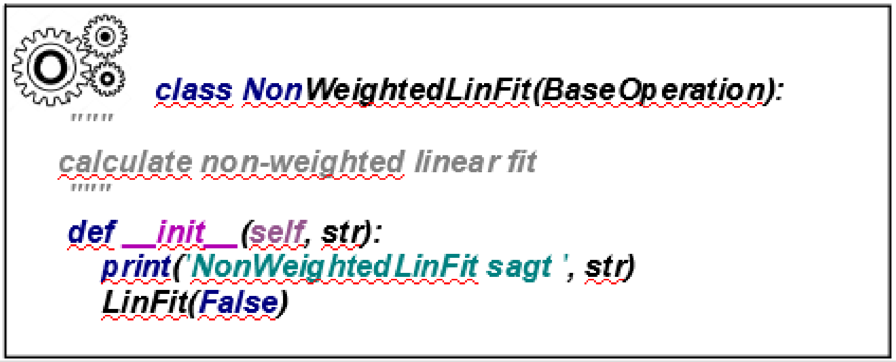
Deutscher Wetterdienst
Wetter und Klima aus einer Hand



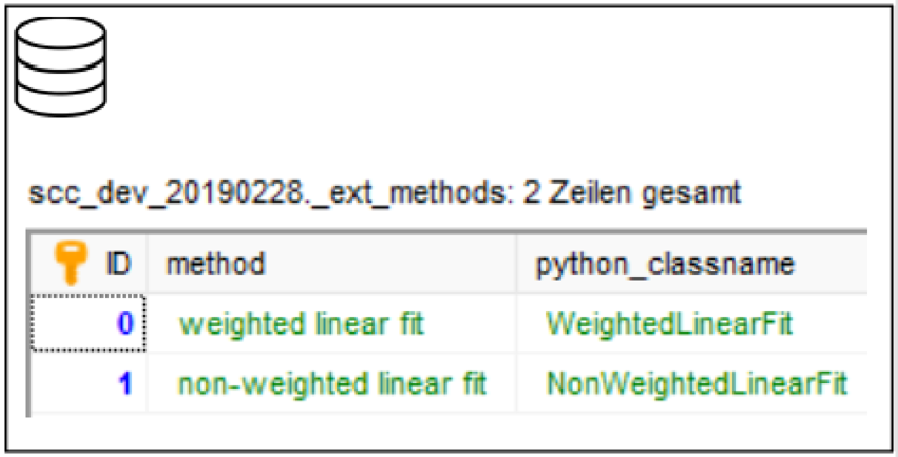
```
class SignalSlope(BaseOperationFactory):
    """
    Calculates signal slope.
    """

    name = 'SignalSlope'

    def get_classname_from_db(self):
        return read_extinction_algorithm(product_id)
```



```
class WeightedLinFit(BaseOperation):
    """
    calculate weighted linear fit
    """
    def __init__(self, str):
        print('WeightedLinFit sagt ', str)
        LinFit(True)
```



```
class NonWeightedLinFit(BaseOperation):
    """
    calculate non-weighted linear fit
    """
    def __init__(self, str):
        print('NonWeightedLinFit sagt ', str)
        LinFit(False)
```



scc_dev_20190228._ext_methods: 2 Zeilen gesamt

ID	method	python_classname
0	weighted linear fit	WeightedLinearFit
1	non-weighted linear fit	NonWeightedLinearFit

Modular design
Example: add a new method (Extinction)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

```
class SignalSlope(BaseOperationFactory):
    """
    Calculates signal slope.
    """

    name = 'SignalSlope'

    def get_classname_from_db(self):
        return read_extinction_algorithm(product_id)
```

```
class WeightedLinFit(BaseOperation):
    """
```

```
class NonWeightedLinFit(BaseOperation):
    """
    calculate non-weighted linear fit
    """

    def __init__(self, str):
        print('NonWeightedLinFit sagt ', str)
        LinFit(False)
```

scc_dev_20190228._ext_methods: 3 Zeilen gesamt

ID	method	python_classname
0	weighted linear fit	WeightedLinearFit
1	non-weighted linear fit	NonWeightedLinearFit
2	Savitzky-Golay fit	SavGolaySlope

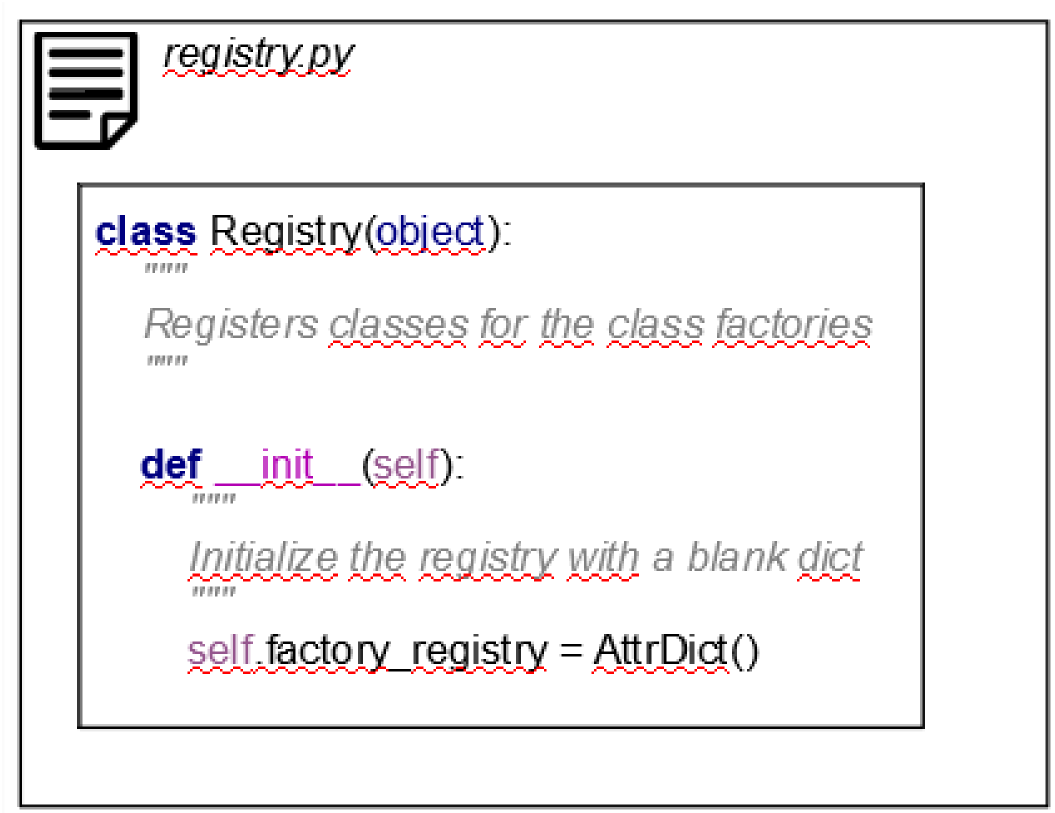
savitzky_golay_slope.py

```
class SavGolaySlope(object):

    def __init__(self, str):
        print('SavGolaySlope ', str)
```

```
registry.register_class(SignalSlope,
                        'SavitzkyGolay',
                        SavGolaySlope)
```

Modular design

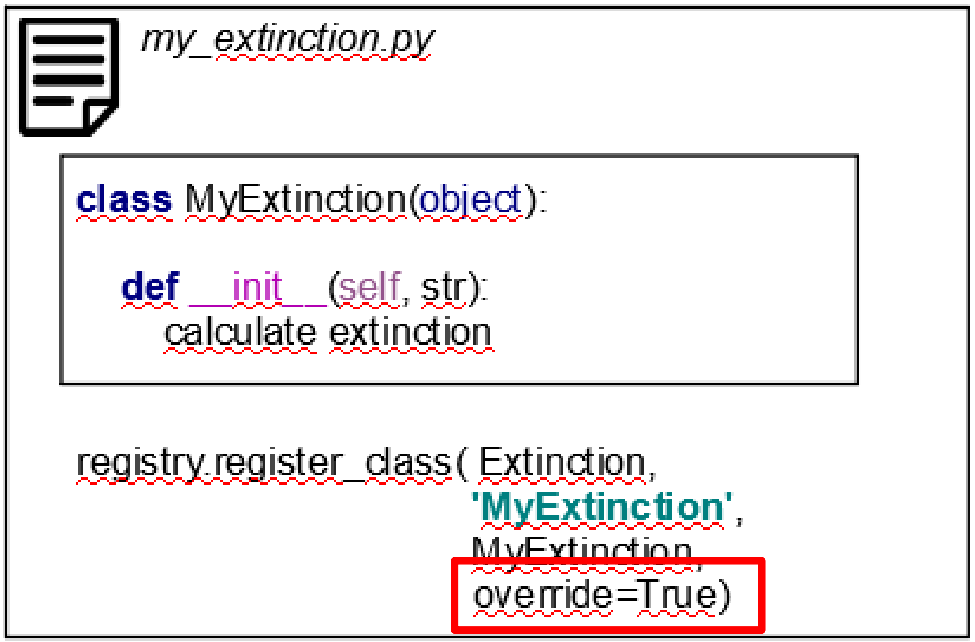
registry



registry.py

```
class Registry(object):
    """
    Registers classes for the class factories
    """

    def __init__(self):
        """
        Initialize the registry with a blank dict
        """
        self.factory_registry = AttrDict()
```

Modular design
Example: replace a method

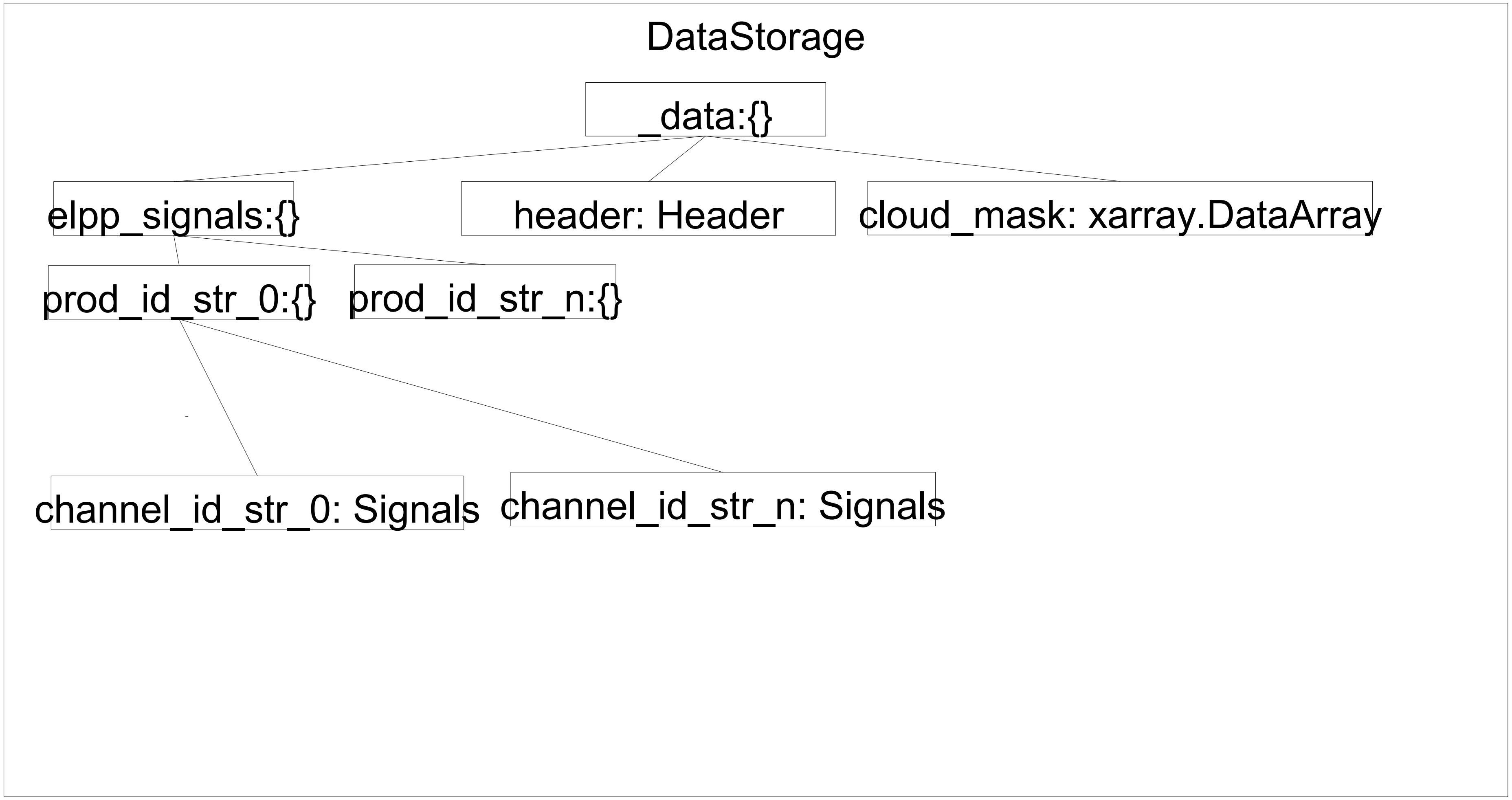


my_extinction.py

```
class MyExtinction(object):

    def __init__(self, str):
        calculate extinction


registry.register_class( Extinction,
                         'MyExtinction',
                         MyExtinction,
                         override=True)
```

global data storage class (under development)

Deutscher Wetterdienst
Wetter und Klima aus einer Hand



Interaction with other modules

- Some new views in user interface
- Add new column id in db table measurements
  - Clean-up db structure ?
- ELPP provides pre-processed signals as before
- ELDAmwl shall provide NetCDF files as before
  - Plus mwl file (hierarchic NC4 file) as discussed in spring 2019