Deutscher Wetterdienst
Wetter und Klima aus einer Hand

DWD

Visualisation of MTG data — Pytroll/Satpy

Gerrit Holl

Deutscher Wetterdienst (DWD)

26 June 2025

What is pytroll?

- Free and open source collection of Python packages
- Reading, processing, and visualising Earth observation satellite data
- With a strong focus on meteorological satellites

What is pytroll?

- Free and open source collection of Python packages
- Reading, processing, and visualising Earth observation satellite data
- With a strong focus on meteorological satellites

What is pytroll?

- Free and open source collection of Python packages
- Reading, processing, and visualising Earth observation satellite data
- With a strong focus on meteorological satellites

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

What is Satpy?

- High-level processing for satellite data
- 124 readers for satellites in LEO, GEO, HEO, and at Lagrange 1
- Resample to any PROJ.4 projection
- Generate all standard RGB composites (and many non-standard ones)
  - And easily define more
- Fully ready for FCI (level 1 and level 2) and LI (level 2).

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Pytroll or Satpy?

- Pytroll: 40 packages, many tasks
- Image manipulation (`trollimage`), filename parsing (`trollsift`), orbit calculations (`pyorbital`)
- Automated processing (`pytroll-watchers`, `pytroll-collectors`, `trollflow2`) configurable via YAML
- And many others
- Most users will only need a handful of packages

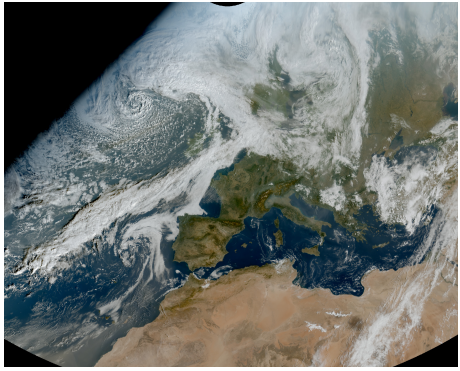
- Satpy: single package
- Used as a Python library via Python code
- Entry-point for most users

Satpy FCI example


```
import hdf5plugin
from satpy import Scene
from glob import glob
fci_files = glob("*FDHSI*BODY*20231001*C_0054_*.nc")
sc = Scene(filenames=fci_files, reader="fci_l1c_nc")
sc.load(["true_color"])
ls = sc.resample("eurol", resampler="gradient_search")
ls.save_datasets()
```

Satpy FCI example


```
import hdf5plugin
from satpy import Scene
from glob import glob
fci_files = glob("*FDHSI*BODY*20231001*C_0054_*.nc")
sc = Scene(filenames=fci_files, reader="fci_l1c_nc")
sc.load(["true_color"])
ls = sc.resample("eurol", resampler="gradient_search")
ls.save_datasets()
```



True Colour, 2023-10-01T08:50

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Reading MTG data with Satpy

- Satpy can read all public MTG-I data formats:
  - FCI L1c NetCDF (FDHSI, HRFI)
  - FCI L2 NetCDF (AMV, CLM, CT, CTTH, FIR, GII, OCA, OLR, CRM, ASR)
  - FCI L2 BUFR (AMV, ASR)
  - FCI L2 GRIB (CLM)
  - LI L2 NetCDF (LEF, LGR, LFL, AF, AFR, AFA)
- Help wanted:
  - IRS L1 (if needed)
  - IRS L2

Visualising MTG data with Satpy

- Visualise single-channel data
- Generate any predefined composite
- Satpy includes all the standard WMO/EUMeTrain composites and many others
- 69 composites directly included
- Easy to define own composite in YAML, no coding needed

Visualising MTG data with Satpy

- Visualise single-channel data
- Generate any predefined composite
- Satpy includes all the standard WMO/EUMeTrain composites and many others
- 69 composites directly included
- Easy to define own composite in YAML, no coding needed

Visualising MTG data with Satpy

- Visualise single-channel data
- Generate any predefined composite
- Satpy includes all the standard WMO/EUMeTrain composites and many others
- 69 composites directly included
- Easy to define own composite in YAML, no coding needed

Visualising MTG data with Satpy

- Visualise single-channel data
- Generate any predefined composite
- Satpy includes all the standard WMO/EUMeTrain composites and many others
- 69 composites directly included
- Easy to define own composite in YAML, no coding needed

Visualising MTG data with Satpy

- Visualise single-channel data
- Generate any predefined composite
- Satpy includes all the standard WMO/EUMeTrain composites and many others
- 69 composites directly included
- Easy to define own composite in YAML, no coding needed

Define your own day/night composite

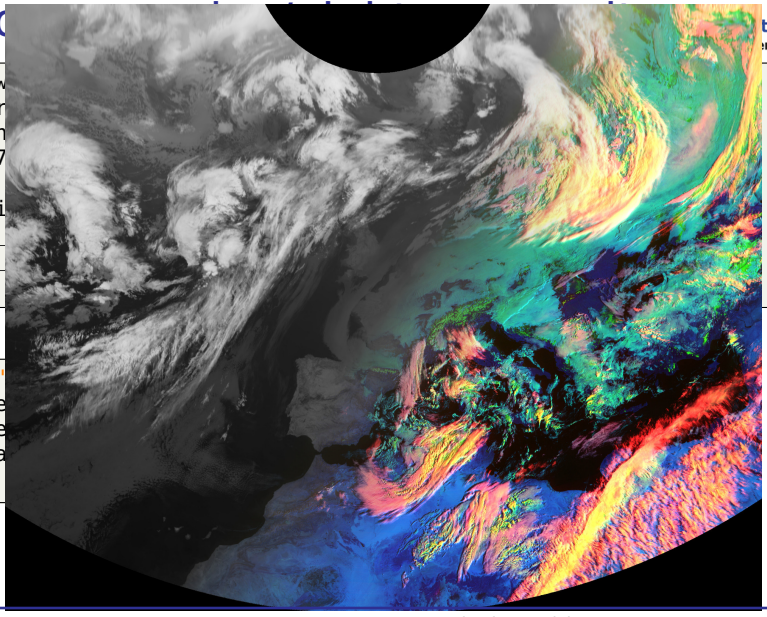

```
cloud_type_with_night_ir105:
  compositor: python/name:satpy.composites.DayNightCompositor
  standard_name: image_ready
  lim_low: 78
  lim_high: 88
  prerequisites:
    - cloud_type
    - night_ir105
```

```
sc.load(["cloud_type_with_night_ir105"])
ls = sc.resample("eurol",
  resampler="gradient_search")
ls.save_datasets()
```

Define yo


```
cloud_type_w
  compositor
  standard_n
  lim_low: 7
  lim_high:
  prerequisi
    - cloud_
    - night_
```

```
sc.load(["
ls = sc.re
  resample
ls.save_da
```



cloud type/night IR, 2025-01-06 08:00

Define own day/night composite

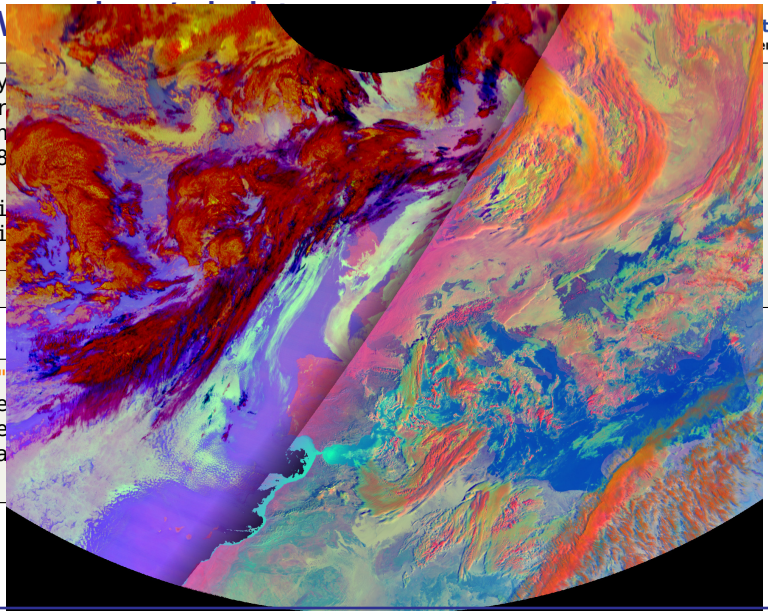

```
day_microphysics_with_night_microphysics:
  compositor: python/name:satpy.composites.DayNightCompositor
  standard_name: image_ready
  lim_low: 85
  lim_high: 90
  prerequisites:
    - day_microphysics
    - night_microphysics
```

```
sc.load(["day_microphysics_with_night_microphysics"])
ls = sc.resample("eurol",
  resampler="gradient_search")
ls.save_datasets()
```

Define ov


```
day_microphy
  compositor
  standard_n
  lim_low: 8
  lim_high:
  prerequisi
    - day_mi
    - night_
```

```
sc.load(["
ls = sc.re
  resample
ls.save_da
```



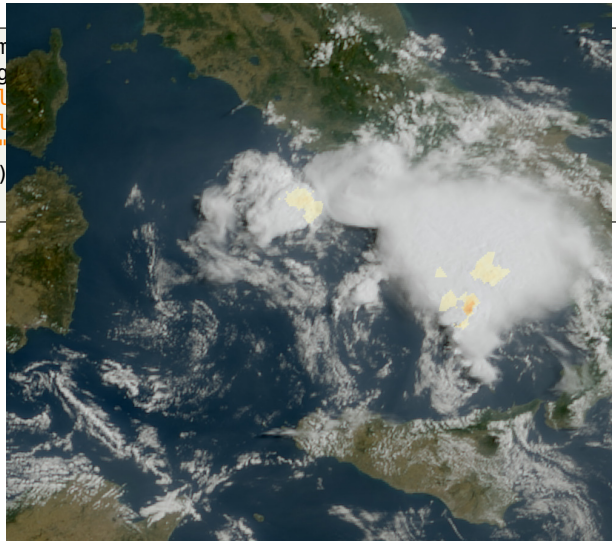
day/night microphysics, 2025-01-06 08:00

FCI + LI


```
sc = Scene(filenames=
  {"fci_l1c_nc": glob("*FCI*FDHSI*003?.nc"),
   "li_l2_nc": "glob("*LI*AFA*0015.nc"")})
sc.load(["true_color_with_night_ir105_acc_flash_area"])
ls = sc.resample("tyrrhenian", resampler="gradient_search")
ls.save_datasets()
```

FCI + LI


```
sc = Scene(filenam
  {"fci_l1c_nc": g
   "li_l2_nc": "gl
sc.load(["true_col
ls = sc.resample("
ls.save_datasets()
```



True color with AFA, 2025-06-18 07:30

Defining new FCI+LI composites


```
day_microphysics_with_night_microphysics_acc_flash_area:
  compositor: python/name:satpy.composites.BackgroundCompositor
  standard_name: image_ready
  prerequisites:
    - acc_flash_area_alpha
    - day_microphysics_with_night_microphysics
```

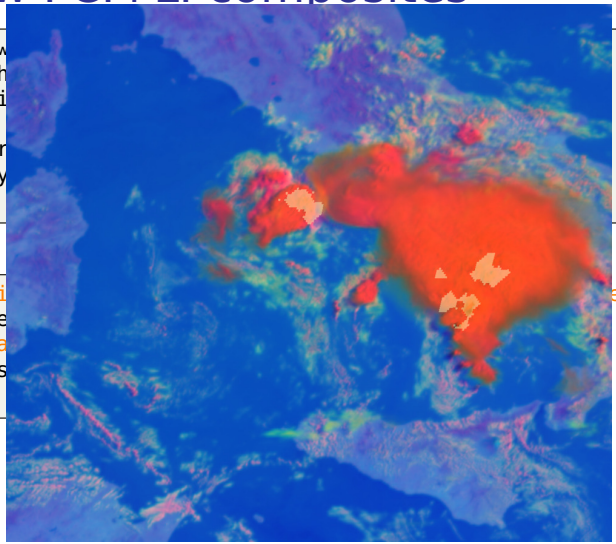
```
sc.load(["day_microphysics_with_night_microphysics_acc_flash_area"])
ls = sc.resample("tyrrhenian",
  resampler="gradient_search")
ls.save_datasets()
```

Defining new FCI+LI composites

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

```
day_microphysics_w
  compositor: pyth
  standard_name: i
  prerequisites:
    - acc_flash_ar
    - day_microphy
```

```
sc.load(["day_mi                                    ea"])
ls = sc.resample
  resampler="gra
ls.save_datasets
```



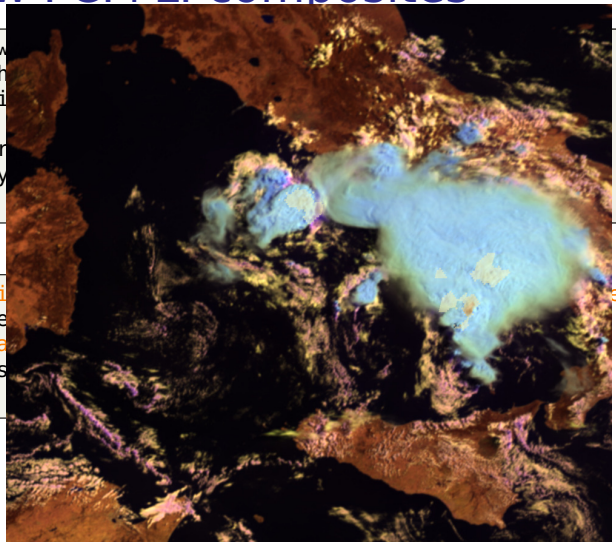
day microphysics with LI AFA, 2025-06-18 07:30

Defining new FCI+LI composites

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

```
day_microphysics_w
  compositor: pyth
  standard_name: i
  prerequisites:
    - acc_flash_ar
    - day_microphy
```

```
sc.load(["day_mi                                    ea"])
ls = sc.resample
  resampler="gra
ls.save_datasets
```



day cloud phase with LI AFA, 2025-06-18 07:30

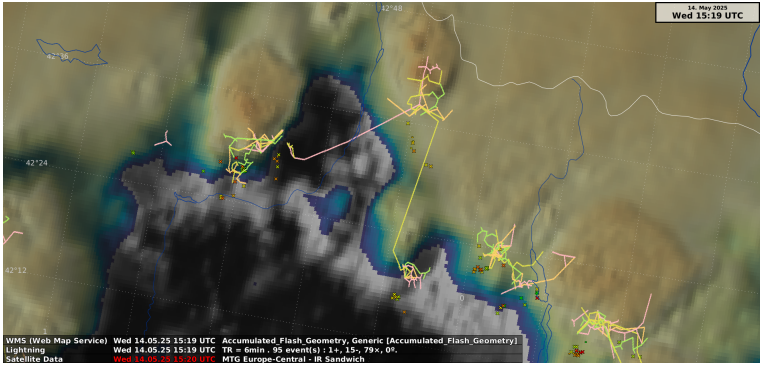
Teaser: Flash geometry

Coming soon to Satpy: Flash geometry based on ESSL algorithm

Teaser: Flash geometry

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Coming soon to Satpy: Flash geometry based on ESSL algorithm



flash geometry

Automated processing

- In an operational environment, we may need automated processing
- Pytroll offers packages for event-based triggering
- Composites and areas defined in YAML config

Automated processing

- In an operational environment, we may need automated processing
- Pytroll offers packages for event-based triggering
- Composites and areas defined in YAML config

Automated processing

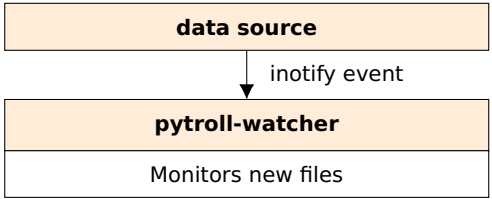
- In an operational environment, we may need automated processing
- Pytroll offers packages for event-based triggering
- Composites and areas defined in YAML config

FCI from EUMETCAST

data source

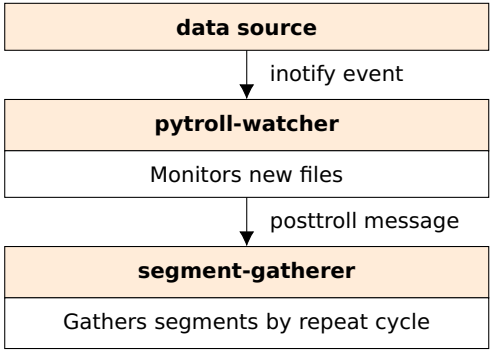
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



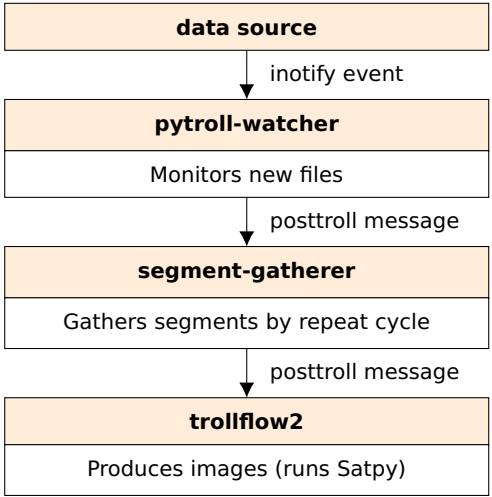
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



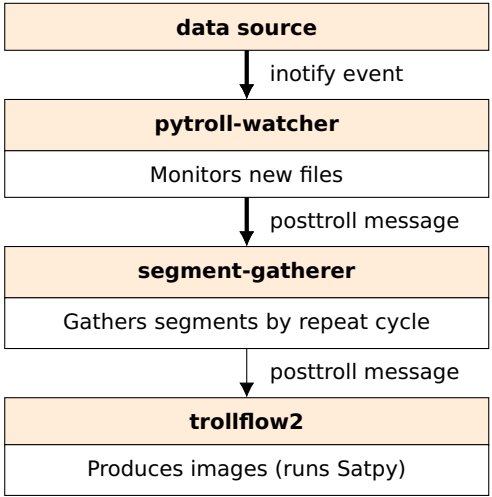
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



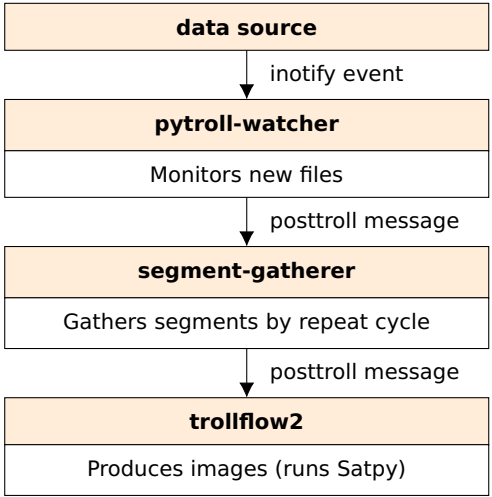
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



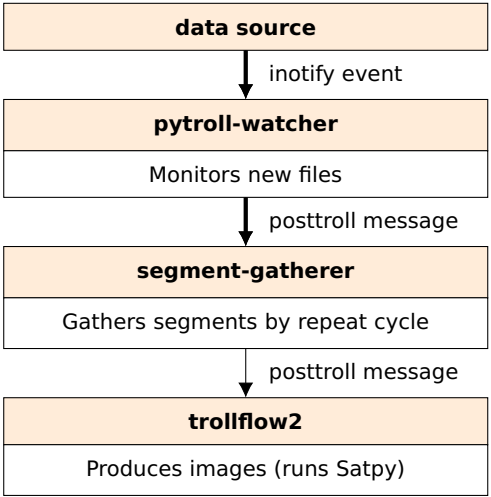
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



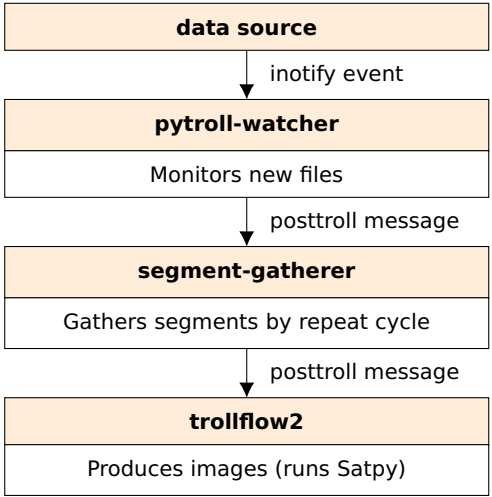
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



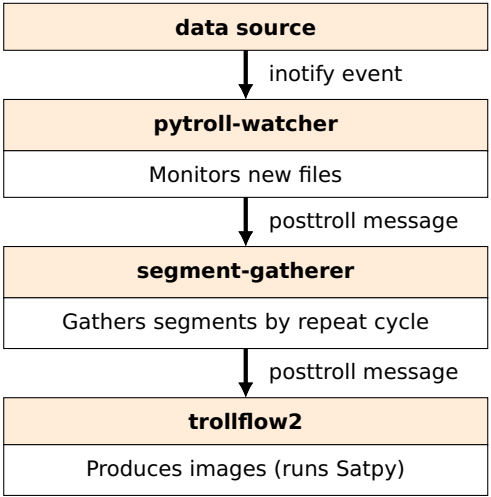
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



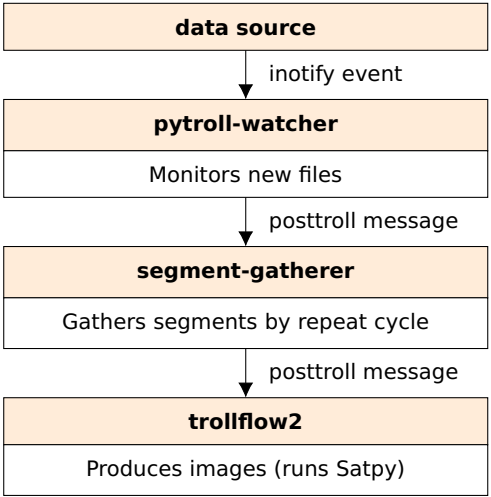
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



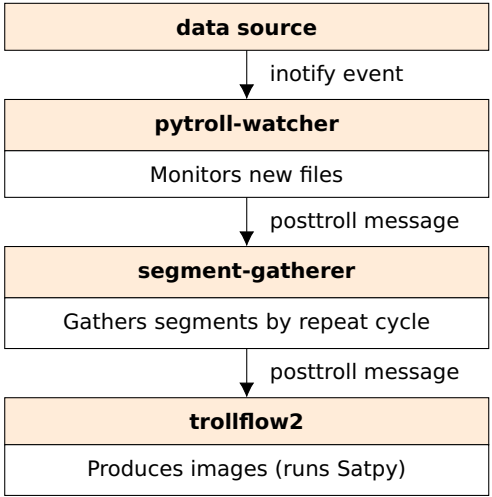
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



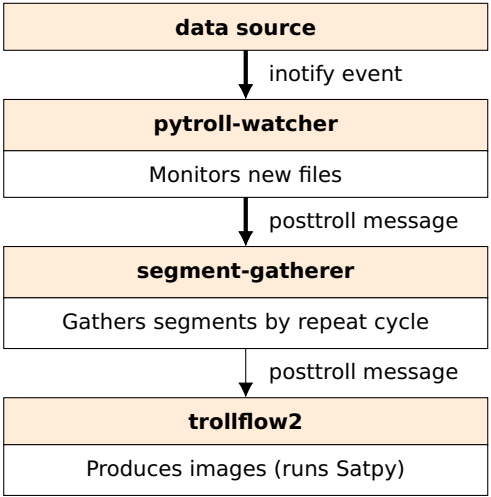
- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

FCI from EUMETCAST



- FCI-FDHSI-BODY-0050-0001.nc
- FCI-FDHSI-BODY-0050-0002.nc
- FCI-FDHSI-BODY-0050-0040.nc
- FCI-FDHSI-TRAIL-0050-0041.nc
- FCI-FDHSI-BODY-0051-0001.nc

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```yaml
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name:trollflow2.plugins.create_scene
  - fun: python/name:trollflow2.plugins.load_composites
  - fun: python/name:trollflow2.plugins.resample
  - fun: python/name:trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```yaml
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name:trollflow2.plugins.create_scene
  - fun: python/name:trollflow2.plugins.load_composites
  - fun: python/name:trollflow2.plugins.resample
  - fun: python/name:trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```yaml
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

What is trollflow2?

- Interfaces to Satpy.
- Define in configuration file:
  - What areas to cover.
  - What channels/composites to generate.
  - What file formats to generate.
  - And configuration parameters.
- Can publish a message when done.
- https://trollflow2.readthedocs.io/

```
product_list:
  subscribe_topics:
    - /set/fci
  reader:
    - fci_l1c_nc
  output_dir: /data/operational/out
  resampler: nearest
  fname_pattern: "{start_time}-{productname}-{areaname}.tif"
  areas:
    europe_500m:
      true_color:
        formats:
          - writer: geotiff
            compress: JPEG
    europe_1km:
      ir_105:
        formats:
          - writer: geotiff
            compress: LZW
workers:
  - fun: python/name: trollflow2.plugins.create_scene
  - fun: python/name: trollflow2.plugins.load_composites
  - fun: python/name: trollflow2.plugins.resample
  - fun: python/name: trollflow2.plugins.save_datasets
```

Installation

- Recommended installation with `miniforge` and `mamba`
- Install from `https://github.com/conda-forge/miniforge`
- `mamba install satpy`
- Warning: Anaconda is non-free for government or commercial use. miniforge and conda-forge are free.

Installation

- Recommended installation with `miniforge` and `mamba`
- Install from `https://github.com/conda-forge/miniforge`
- `mamba install satpy`
- Warning: Anaconda is non-free for government or commercial use. miniforge and conda-forge are free.

Installation

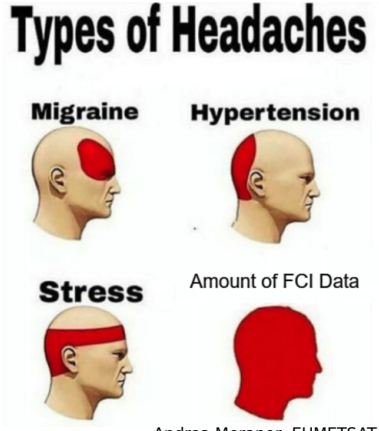
- Recommended installation with `miniforge` and `mamba`
- Install from `https://github.com/conda-forge/miniforge`
- `mamba install satpy`
- Warning: Anaconda is non-free for government or commercial use. miniforge and conda-forge are free.

Installation

- Recommended installation with `miniforge` and `mamba`
- Install from `https://github.com/conda-forge/miniforge`
- `mamba install satpy`
- Warning: Anaconda is non-free for government or commercial use. miniforge and conda-forge are free.

Satpy pitfalls

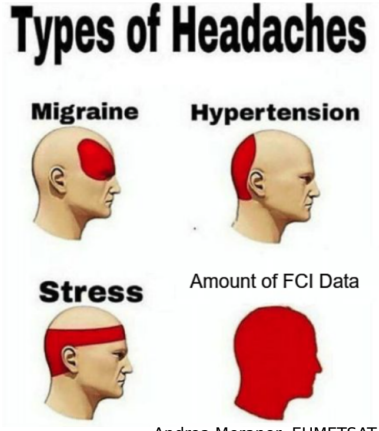
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

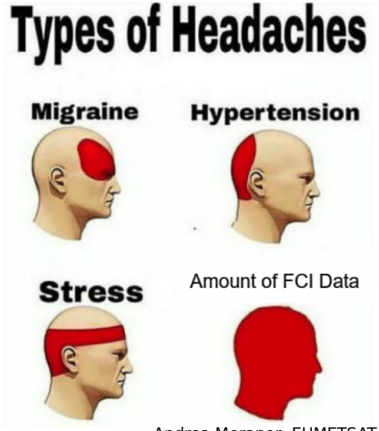
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

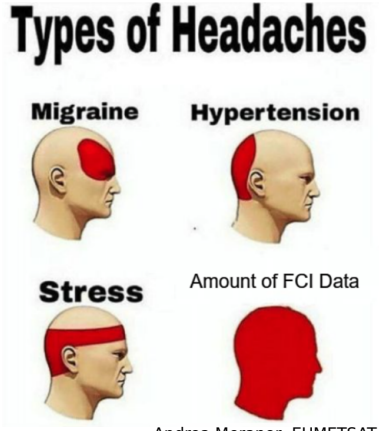
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

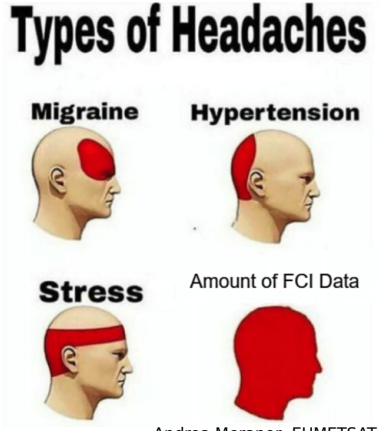
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

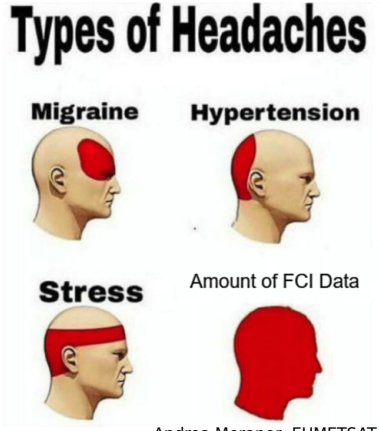
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

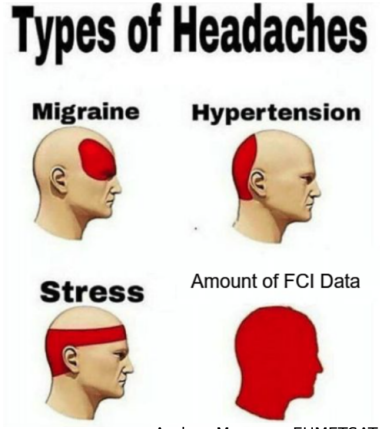
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

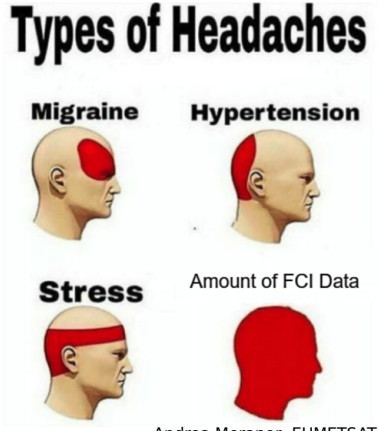
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

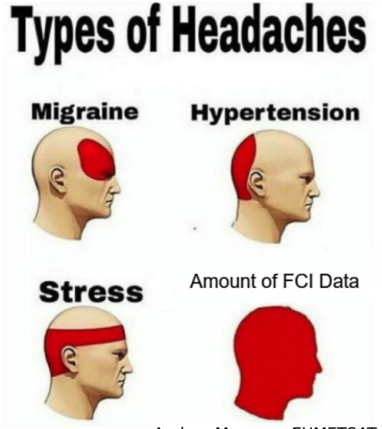
- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Satpy pitfalls

- Easy! What can possibly go wrong?
- `ArrayMemoryError`, `Segmentation Fault`, ...
  - Process less data
  - Upgrade machine
  - Change `DASK_NUM_WORKERS` (check Satpy FAQ)
- `undefined filter encountered`
  - `import hdf5plugin` BEFORE `import satpy`
- My composite looks completely wrong
  - `standard_name: image_ready` or read up on enhancements



Andrea Meraner, EUMETSAT

Getting help

- https://satpy.readthedocs.io
- Pytroll Slack via https://pytroll.github.io
- Mailing-list

Getting help

- https://satpy.readthedocs.io
- Pytroll Slack via https://pytroll.github.io
- Mailing-list

Getting help

- https://satpy.readthedocs.io
- Pytroll Slack via https://pytroll.github.io
- Mailing-list

Acknowledgements

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Pytroll and Satpy developed by Martin Raspaud, David Hoese, Panu Lahtinen, Gerrit Holl, Simon Proud, Stephan Finkensieper, Andrea Meraner, Adam Dybbroe, Johan Strandgren, Lu Liu, Joleen Feltz, Benjamin Rösner, Sauli Joro, Xin Zhang, Pierre de Buyl, Youva Aoun, Gionata Ghiggi, William Roberts, Lars Ørum Rasmussen, Clement Laplace, Olivier Samain, Mitch Herbertson, Yufei Zhu, Jorge Humberto Bravo Méndez, Sven-Erik Enno, Rohan Daruwala, Beke Kremmling, Trygve Aspenes, Talfan Barnie, Jonathan Beavers, Suyash Behera, Ray Bell, Jorge Bravo, Sebastian Brodehl, Andrew Brooks, Guido della Bruna, Pierre de Buyl, Eric Bruning, Manuel Carranza, Guido Cioni, Lorenzo Clementi, Colin Duff†, Ulrik Egede, Florian Fichtner, Andrea Grillini, Blanka Gvozdikova, Nina Håkansson, Ulrich Hamann, Marc Honnorat, Chung-Hsiang Horng, Mario Hros, Lloyd Hughes, Sara Hörnquist, Mikhail Itkin, Tommy Jasmin, Jactry Zeng, Zoé Jequier, Johannes Johansson, Przemyslaw Juda, Pouria Khalaj, Janne Kotro, Ralph Kuehn, Jussi Leinonen, Thomas Leppelt, Aronne Merrelli, Luca Merucci, Lucas Meyer, Zifeng Mo, David Navia, Ondřej Nedelcev, Oana Nicola. Esben S. Nielsen, Tom Parker, Christian Peters, Pepe Phillips, Ghislain Picard, Martin Radenz, Emmanuel Roche, Pascale Roquet, Kristian Rune Larsen, RutgerK, Bengt Rydberg, Marco Sassi, Stefan Scheiblauer, Ronald Scheirer, Michael Schmutz, Hauke Schulz, Jakub Seidl, Will Sharpe, Eysteinn Sigurðsson, Jean-Luc Shaw, Dario Stelitano, Matias Takala, Taiga Tsukada, Antonio Valentino, Christian Versloot, Helga Weber, hazbottles, oananicola, praerien, Sara Hörnquist, Clément, Xuanhan Lai, Camilla Hundahl Johnsen, Ruben Sala, Christian Kliche, Paulo V. C. Medeiros, and others