

SATPY

an open-source Python library for Earth Observation
satellite data

Pierre de Buyl, PyTroll developers

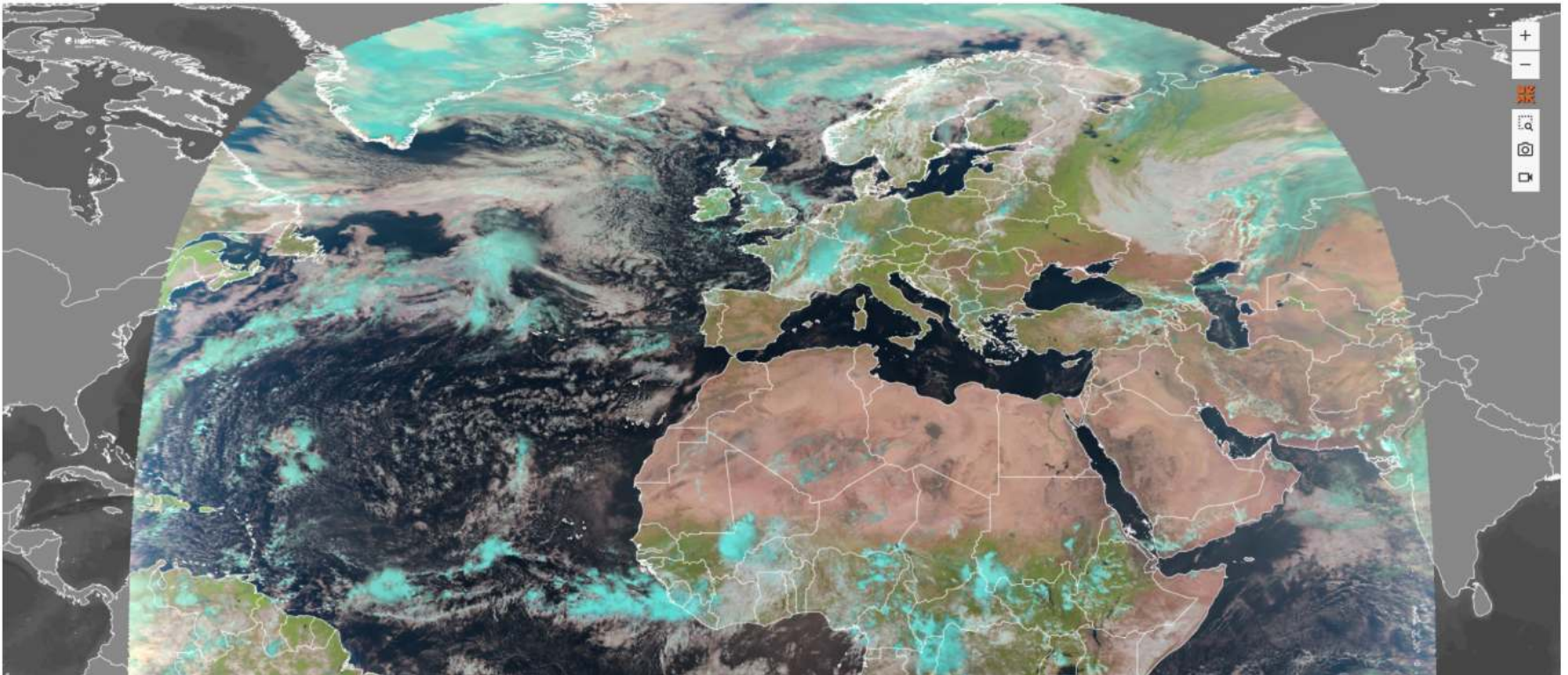
EUMETSAT 2025 - Lyon

OVERVIEW

SATPY @ VIEW.EUMETSAT.INT

PyTroll/satpy-based image generation for MSG 0 degree service

As announced in June, the gradual transition of the EUMETView image generation of MSG layers to a new, PyTroll/satpy-based system is ongoing. After the transition of MSG IODC layers, MSG 0 degree layers are also migrated to the new system on 26 August, with the MSG RSS Service migration to follow soon, which will complete the transition.



WHAT IS SATPY?

- Started in 2015.
- Satpy is an open-source software library for reading and processing data from Earth observing satellite instruments.
- Satpy is part of the PyTroll project, a collection of open-source software.
- Satpy can probably read "your" satellite (or you can add the reader)

WHO IS SATPY

- PyTroll is a community of developers started in 2009.
- PyTroll is supported by a Memorandum of Understanding, a formal collaboration agreement between SMHI, FMI, KNMI, DWD, MeteoSwiss, and MET Norway.
- More than 100 persons have contributed to the satpy repository.

Pytroll

[Code of Conduct](#) [Community](#) [Gallery](#) [Coding Guidelines](#) [History of Pytroll](#) [Pytroll Project Logos](#)

[Contributor Weeks](#) [Pytroll Manifesto](#) [Packages Summary](#) [Software](#) [Pytroll User Days](#)



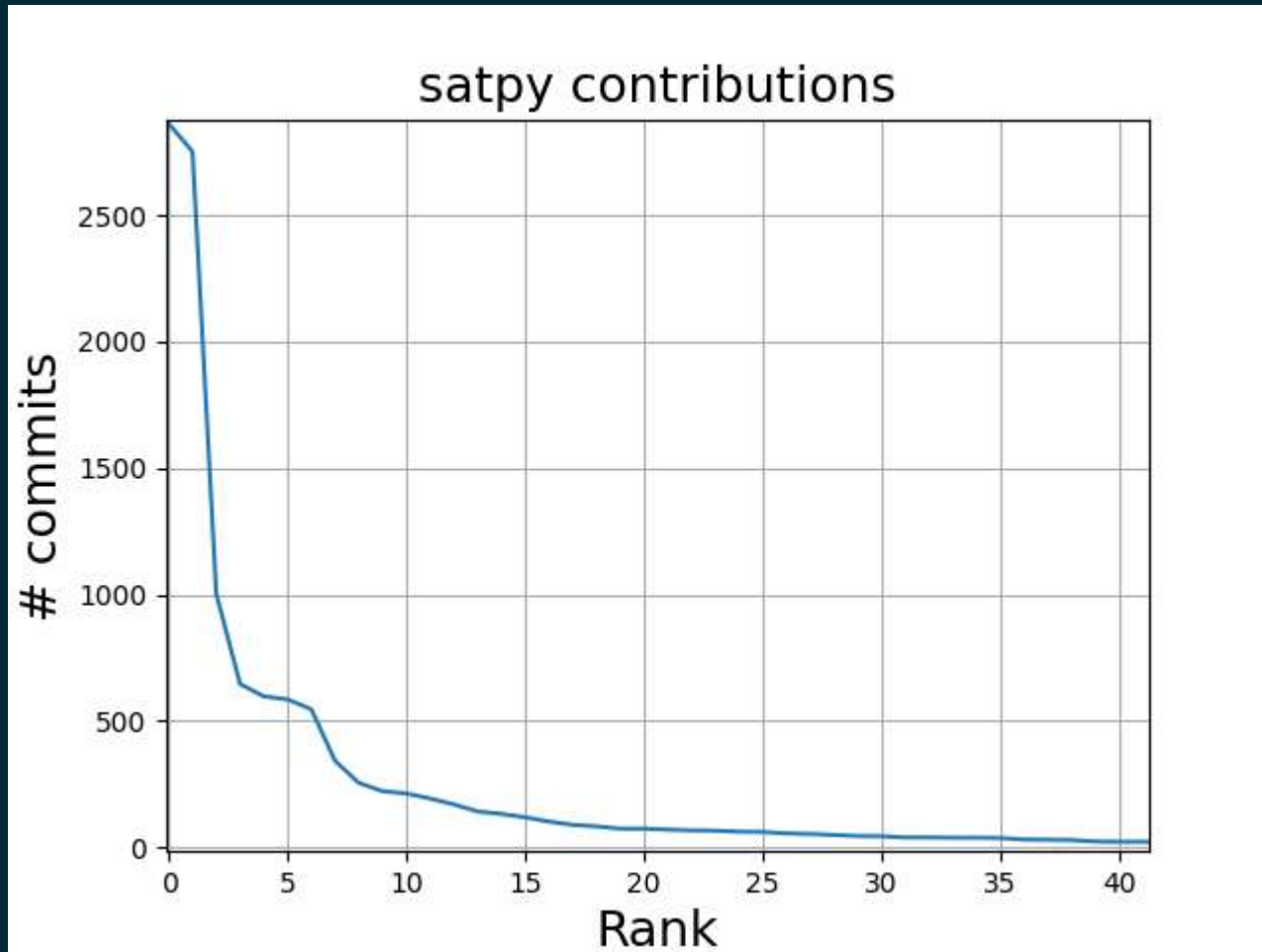
Pytroll is an easy to use, modular, free and open source python framework for the processing of earth observation satellite data. The provided python packages are designed to be used both in R&D environments and in 24/7 operational production.

The focus is on atmospheric applications and imaging sensors, but as seen from the list of supported satellite sensors below the data that can be handled by Pytroll allows the usage in a wide range of earth sciences.

PYTROLL WORKSHOPS



SATPY CONTRIBUTOR GRAPH



SATPY DISCUSSIONS

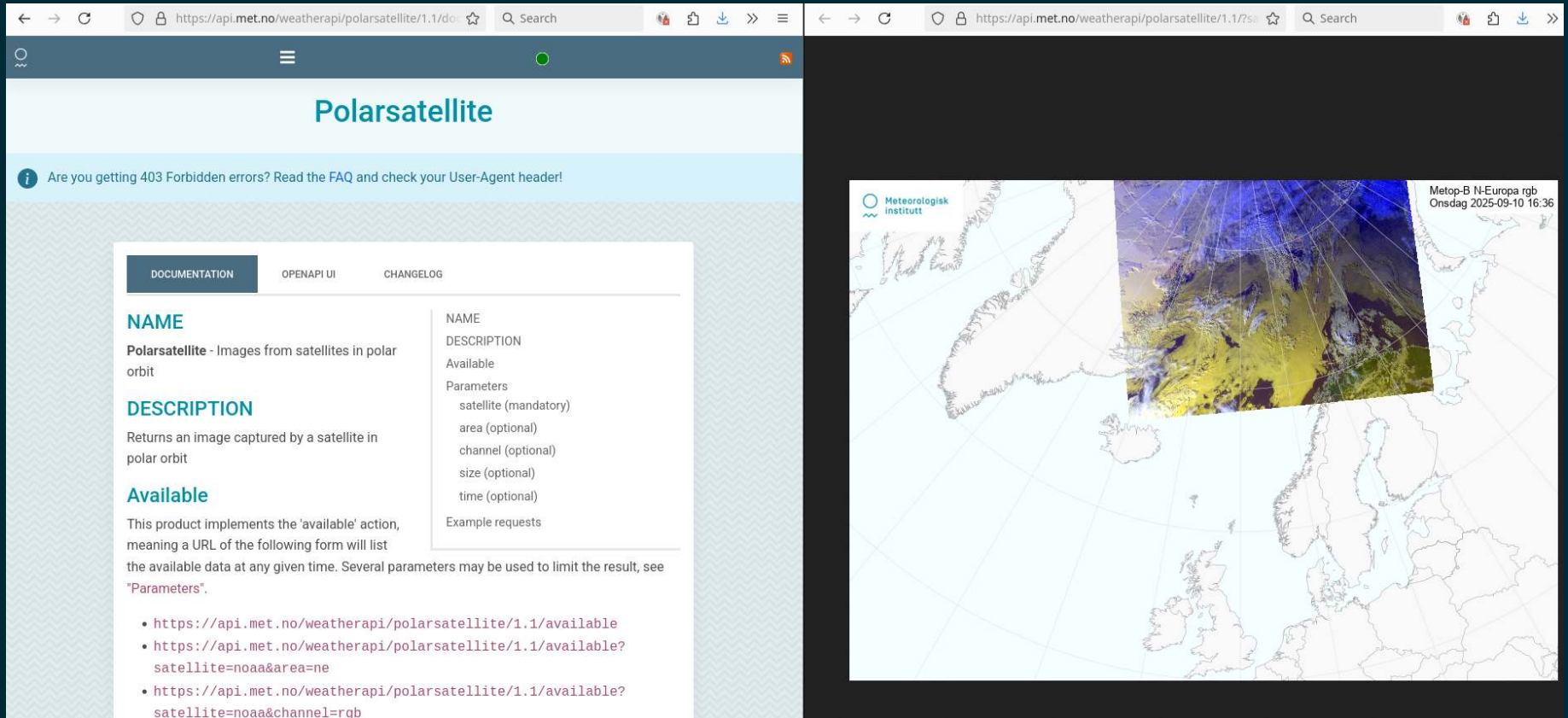
- Via GitHub issues or slack (#satpy, #pyresample, #composites, etc.)
- Tips for reading and displaying data
- Bug fixing
- New instruments

WHO USES SATPY?

CSPP GEOSPHERE (SSEC - UNIVERSITY OF WISCONSIN – MADISON)



MET.NO API SERVER



A map of Europe and its surrounding regions, including parts of North Africa, the Middle East, and Iceland. Countries are color-coded: dark blue for the United Kingdom, Ireland, Iceland, Norway, Sweden, Finland, Denmark, Germany, Poland, Czech Republic, Slovakia, Austria, Hungary, Switzerland, Italy, Greece, Turkey, and Cyprus; light blue for France, Spain, Portugal, and Malta; and grey for Russia, Ukraine, Belarus, Lithuania, Latvia, Estonia, Latvia, Georgia, Armenia, Azerbaijan, and the Balkans (Bulgaria, Romania, Serbia, Montenegro, Albania, Kosovo, Bosnia and Herzegovina, Macedonia, and Albania).

EUROPEAN MET OFFICES

EUMETSAT - FCI PRODUCT GUIDE



EUM/MTG/USR/13/719113
v1J e-signed, 5 June 2020

MTG FCI L1 Product User Guide [FCIL1PUG]MTG-831410

geophysical parameters from various file formats to the common *Xarray* [DataArray](#) and [Dataset](#) classes for easier interoperability with other scientific python libraries. *Satpy* also provides interfaces for creating RGB (Red/Green/Blue) images and other composite types by combining data from multiple instrument bands or products. Various atmospheric corrections and visual enhancements are provided for improving the usefulness and quality of output images. The [Pyresample](#) package is used to resample data to different uniform areas or grids. The documentation is available at <http://satpy.readthedocs.org/>.

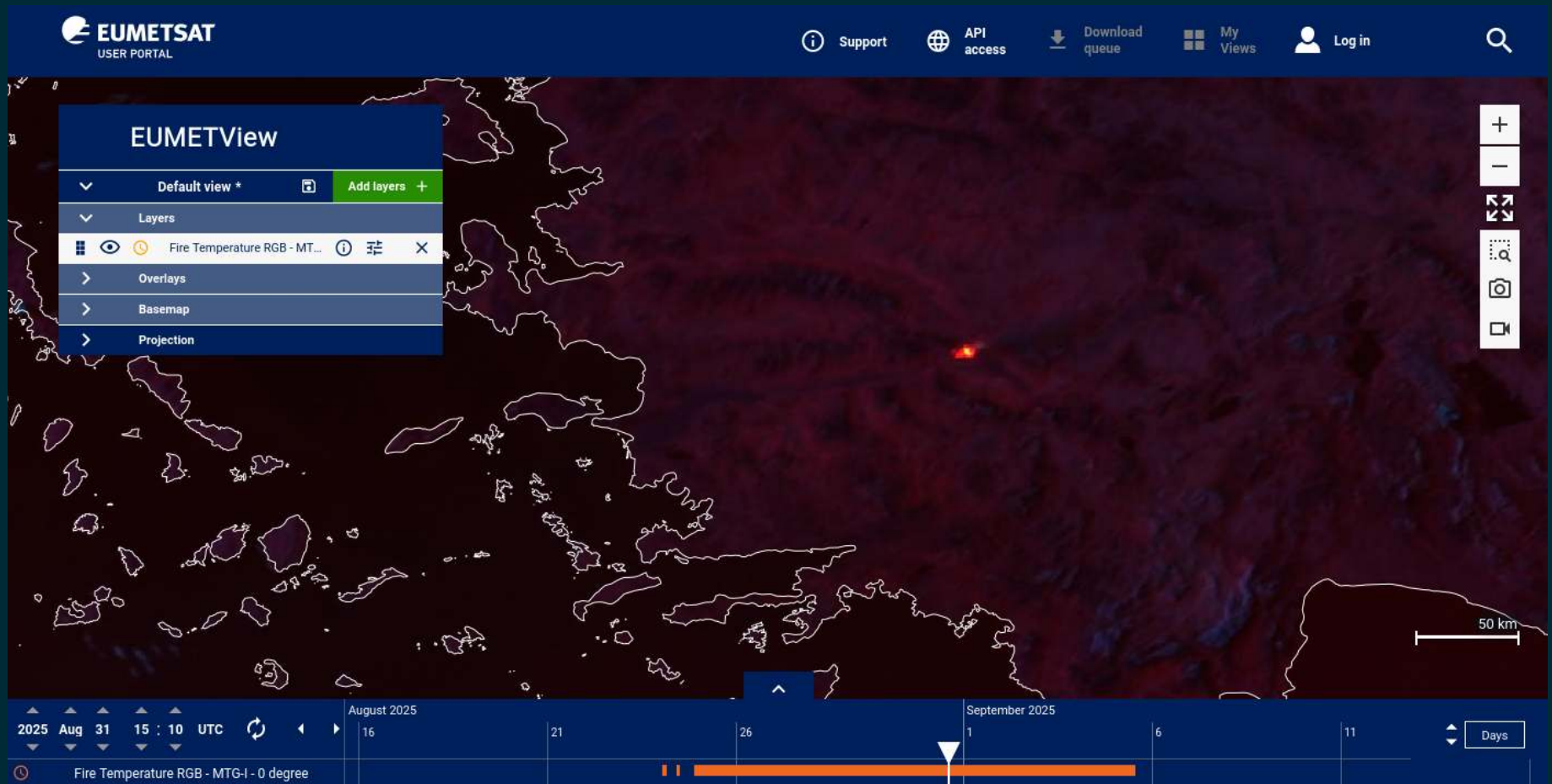
Satpy also includes a reader of the FCI Level 1c data. The following Python code snippet shows an example on how to use *Satpy* to generate a `natural_color` RGB composite over the European area. A more detailed tutorial is available as part of the Satpy documentation at https://satpy.readthedocs.io/en/latest/examples/fci_l1c_natural_color.html.

```
from satpy.scene import Scene
from satpy import find_files_and_readers
```

EUMETSAT - FCI TRUE COLOR



EUMETSAT - FCI FIRE PRODUCT



TECHNICAL POINT OF VIEW

WHAT'S IN THE BOX (EXCERPT)

- Base dependencies: `dask`, `NumPy`, `pillow`, `pykdtree`, `pyproj`, `pyresample`, `trollimage`, `trollsift`, `xarray`, `zarr`
- Readers dependencies: `pyhdf`, `netCDF4`, `rasterio`, `rioxarray`, `pyorbital`, `h5netcdf`, `pyPublicDecompWT`, `pygrib`, `fsspec`, `numba`

CAPABILITIES

```
>>> print(satpy.available_readers())  
['abi_l1b', 'abi_l1b_scmi', 'abi_l2_nc', 'acspo',  
'agri_fy4a_l1', 'agri_fy4b_l1', 'ahi_hrit', 'ahi_hsd',  
'ahi_l1b_gridded_bin', 'ahi_l2_nc', 'ami_l1b', 'amsr2_l1b',  
'amsr2_l2', 'amsr2_l2_gaasp', 'amsub_l1c_aapp',  
'ascats_12_soilmoisture_buf'r', 'atms_l1b_nc', 'atms_sdr hdf5',  
'avhrr_l1b_aapp', 'avhrr_l1b_eps', 'avhrr_l1b_hrpt',  
'avhrr_l1c_eum_gac_fdr_nc', 'aws1_mwr_l1b_nc',  
'aws1_mwr_l1c_nc', 'camel_l3_nc', 'clavrx', 'cmsaf-  
claas2_l2_nc', 'electrol_hrit', 'epic_l1b_h5',  
'eps_sterna_mwr_l1b_nc', 'fci_l1c_nc', 'fci_l2_buf'r',  
'fci_l2_grib', 'fci_l2_nc', 'fy3a_mersi1_l1b',  
'fy3b_mersi1_l1b', 'fy3c_mersi1_l1b', 'generic_image', 'geocat',  
'gerb_l2_hr_h5', 'ghi_l1', 'ghrsst_l2', 'gld360_uarf2',  
'glm_l2', 'gms5-vissr_l1b', 'goci2_l2_nc', 'goes-imager_hrit',
```

- Calibration, Resampling, Composite generation

SATPY SCENE OBJECT

```
fn = glob.glob('/mnt/data/MTI1_202503101400/  
W*RAD*FDHSI*BODY*.nc')  
scn = satpy.Scene(filenamees=fn, reader='fci_l1c_nc')
```

SATPY SCENE - DATASET

```
scn.load(['ir_133'])  
scn['ir_133']
```

```
<xarray.DataArray (y: 5568, x: 5568)> Size: 124MB  
dask.array<concatenate, shape=(5568, 5568), dtype=float32,  
chunksize=(150, 5568), chunktype=numpy.ndarray>  
Coordinates:  
    crs          object 8B  
PROJCRS["unknown", BASEGEOGCRS["unknown", DATUM["unknown...  
    * y          (y) float64 45kB -5.567e+06 -5.565e+06 ...  
5.565e+06 5.567e+06  
    * x          (x) float64 45kB -5.567e+06 -5.565e+06 ...  
5.565e+06 5.567e+06  
Attributes: (12/21)  
    orbital_parameters: {'satellite_actual_longitude':  
-0.2617217555642128,...  
    sensor:             fci  
    standard_name:       toa_brightness_temperature  
    ..  
    ..
```

SATPY SCENE - COMPOSITES

```
scn.available_composite_names()
```

```
[ '24h_microphysics', 'airmass', 'ash', 'cimss_cloud_type',
'cimss_cloud_type_raw', 'cloud_phase',
'cloud_phase_distinction', 'cloud_phase_distinction_raw',
'cloud_phase_raw', 'cloud_phase_with_night_ir105', 'cloud_type',
'cloud_type_with_night_ir105', 'cloudtop',
'colorized_ir_clouds', 'convection',
'day_essl_colorized_low_level_moisture',
'day_essl_low_level_moisture', 'day_microphysics',
'day_severe_storms', 'day_severe_storms_tropical', 'dust',
'essl_colorized_low_level_moisture', 'essl_low_level_moisture',
'fci_fire_channels_sum', 'fire_temperature',
'fire_temperature_38refl', 'fire_temperature_rad',
'flames_masked', 'fog', 'geo_color',
'geo_color_background_with_low_clouds', 'geo_color_high_clouds',
'geo_color_low_clouds', 'geo_color_night', 'green_snow',
'ice_area_fraction_100_200', 'ice_area_fraction_200_300', 'ice_area_fraction_300_400', 'ice_area_fraction_400_500', 'ice_area_fraction_500_600', 'ice_area_fraction_600_700', 'ice_area_fraction_700_800', 'ice_area_fraction_800_900', 'ice_area_fraction_900_1000', 'ice_area_fraction_1000_1100', 'ice_area_fraction_1100_1200', 'ice_area_fraction_1200_1300', 'ice_area_fraction_1300_1400', 'ice_area_fraction_1400_1500', 'ice_area_fraction_1500_1600', 'ice_area_fraction_1600_1700', 'ice_area_fraction_1700_1800', 'ice_area_fraction_1800_1900', 'ice_area_fraction_1900_2000', 'ice_area_fraction_2000_2100', 'ice_area_fraction_2100_2200', 'ice_area_fraction_2200_2300', 'ice_area_fraction_2300_2400', 'ice_area_fraction_2400_2500', 'ice_area_fraction_2500_2600', 'ice_area_fraction_2600_2700', 'ice_area_fraction_2700_2800', 'ice_area_fraction_2800_2900', 'ice_area_fraction_2900_3000', 'ice_area_fraction_3000_3100', 'ice_area_fraction_3100_3200', 'ice_area_fraction_3200_3300', 'ice_area_fraction_3300_3400', 'ice_area_fraction_3400_3500', 'ice_area_fraction_3500_3600', 'ice_area_fraction_3600_3700', 'ice_area_fraction_3700_3800', 'ice_area_fraction_3800_3900', 'ice_area_fraction_3900_4000', 'ice_area_fraction_4000_4100', 'ice_area_fraction_4100_4200', 'ice_area_fraction_4200_4300', 'ice_area_fraction_4300_4400', 'ice_area_fraction_4400_4500', 'ice_area_fraction_4500_4600', 'ice_area_fraction_4600_4700', 'ice_area_fraction_4700_4800', 'ice_area_fraction_4800_4900', 'ice_area_fraction_4900_5000', 'ice_area_fraction_5000_5100', 'ice_area_fraction_5100_5200', 'ice_area_fraction_5200_5300', 'ice_area_fraction_5300_5400', 'ice_area_fraction_5400_5500', 'ice_area_fraction_5500_5600', 'ice_area_fraction_5600_5700', 'ice_area_fraction_5700_5800', 'ice_area_fraction_5800_5900', 'ice_area_fraction_5900_6000', 'ice_area_fraction_6000_6100', 'ice_area_fraction_6100_6200', 'ice_area_fraction_6200_6300', 'ice_area_fraction_6300_6400', 'ice_area_fraction_6400_6500', 'ice_area_fraction_6500_6600', 'ice_area_fraction_6600_6700', 'ice_area_fraction_6700_6800', 'ice_area_fraction_6800_6900', 'ice_area_fraction_6900_7000', 'ice_area_fraction_7000_7100', 'ice_area_fraction_7100_7200', 'ice_area_fraction_7200_7300', 'ice_area_fraction_7300_7400', 'ice_area_fraction_7400_7500', 'ice_area_fraction_7500_7600', 'ice_area_fraction_7600_7700', 'ice_area_fraction_7700_7800', 'ice_area_fraction_7800_7900', 'ice_area_fraction_7900_8000', 'ice_area_fraction_8000_8100', 'ice_area_fraction_8100_8200', 'ice_area_fraction_8200_8300', 'ice_area_fraction_8300_8400', 'ice_area_fraction_8400_8500', 'ice_area_fraction_8500_8600', 'ice_area_fraction_8600_8700', 'ice_area_fraction_8700_8800', 'ice_area_fraction_8800_8900', 'ice_area_fraction_8900_9000', 'ice_area_fraction_9000_9100', 'ice_area_fraction_9100_9200', 'ice_area_fraction_9200_9300', 'ice_area_fraction_9300_9400', 'ice_area_fraction_9400_9500', 'ice_area_fraction_9500_9600', 'ice_area_fraction_9600_9700', 'ice_area_fraction_9700_9800', 'ice_area_fraction_9800_9900', 'ice_area_fraction_9900_10000', 'ice_area_fraction_10000_10100', 'ice_area_fraction_10100_10200', 'ice_area_fraction_10200_10300', 'ice_area_fraction_10300_10400', 'ice_area_fraction_10400_10500', 'ice_area_fraction_10500_10600', 'ice_area_fraction_10600_10700', 'ice_area_fraction_10700_10800', 'ice_area_fraction_10800_10900', 'ice_area_fraction_10900_11000', 'ice_area_fraction_11000_11100', 'ice_area_fraction_11100_11200', 'ice_area_fraction_11200_11300', 'ice_area_fraction_11300_11400', 'ice_area_fraction_11400_11500', 'ice_area_fraction_11500_11600', 'ice_area_fraction_11600_11700', 'ice_area_fraction_11700_11800', 'ice_area_fraction_11800_11900', 'ice_area_fraction_11900_12000', 'ice_area_fraction_12000_12100', 'ice_area_fraction_12100_12200', 'ice_area_fraction_12200_12300', 'ice_area_fraction_12300_12400', 'ice_area_fraction_12400_12500', 'ice_area_fraction_12500_12600', 'ice_area_fraction_12600_12700', 'ice_area_fraction_12700_12800', 'ice_area_fraction_12800_12900', 'ice_area_fraction_12900_13000', 'ice_area_fraction_13000_13100', 'ice_area_fraction_13100_13200', 'ice_area_fraction_13200_13300', 'ice_area_fraction_13300_13400', 'ice_area_fraction_13400_13500', 'ice_area_fraction_13500_13600', 'ice_area_fraction_13600_13700', 'ice_area_fraction_13700_13800', 'ice_area_fraction_13800_13900', 'ice_area_fraction_13900_14000', 'ice_area_fraction_14000_14100', 'ice_area_fraction_14100_14200', 'ice_area_fraction_14200_14300', 'ice_area_fraction_14300_14400', 'ice_area_fraction_14400_14500', 'ice_area_fraction_14500_14600', 'ice_area_fraction_14600_14700', 'ice_area_fraction_14700_14800', 'ice_area_fraction_14800_14900', 'ice_area_fraction_14900_15000', 'ice_area_fraction_15000_15100', 'ice_area_fraction_15100_15200', 'ice_area_fraction_15200_15300', 'ice_area_fraction_15300_15400', 'ice_area_fraction_15400_15500', 'ice_area_fraction_15500_15600', 'ice_area_fraction_15600_15700', 'ice_area_fraction_15700_15800', 'ice_area_fraction_15800_15900', 'ice_area_fraction_15900_16000', 'ice_area_fraction_16000_16100', 'ice_area_fraction_16100_16200', 'ice_area_fraction_16200_16300', 'ice_area_fraction_16300_16400', 'ice_area_fraction_16400_16500', 'ice_area_fraction_16500_16600', 'ice_area_fraction_16600_16700', 'ice_area_fraction_16700_16800', 'ice_area_fraction_16800_16900', 'ice_area_fraction_16900_17000', 'ice_area_fraction_17000_17100', 'ice_area_fraction_17100_17200', 'ice_area_fraction_17200_17300', 'ice_area_fraction_17300_17400', 'ice_area_fraction_17400_17500', 'ice_area_fraction_17500_17600', 'ice_area_fraction_17600_17700', 'ice_area_fraction_17700_17800', 'ice_area_fraction_17800_17900', 'ice_area_fraction_17900_18000', 'ice_area_fraction_18000_18100', 'ice_area_fraction_18100_18200', 'ice_area_fraction_18200_18300', 'ice_area_fraction_18300_18400', 'ice_area_fraction_18400_18500', 'ice_area_fraction_18500_18600', 'ice_area_fraction_18600_18700', 'ice_area_fraction_18700_18800', 'ice_area_fraction_18800_18900', 'ice_area_fraction_18900_19000', 'ice_area_fraction_19000_19100', 'ice_area_fraction_19100_19200', 'ice_area_fraction_19200_19300', 'ice_area_fraction_19300_19400', 'ice_area_fraction_19400_19500', 'ice_area_fraction_19500_19600', 'ice_area_fraction_19600_19700', 'ice_area_fraction_19700_19800', 'ice_area_fraction_19800_19900', 'ice_area_fraction_19900_20000', 'ice_area_fraction_20000_20100', 'ice_area_fraction_20100_20200', 'ice_area_fraction_20200_20300', 'ice_area_fraction_20300_20400', 'ice_area_fraction_20400_20500', 'ice_area_fraction_20500_20600', 'ice_area_fraction_20600_20700', 'ice_area_fraction_20700_20800', 'ice_area_fraction_20800_20900', 'ice_area_fraction_20900_21000', 'ice_area_fraction_21000_21100', 'ice_area_fraction_21100_21200', 'ice_area_fraction_21200_21300', 'ice_area_fraction_21300_21400', 'ice_area_fraction_21400_21500', 'ice_area_fraction_21500_21600', 'ice_area_fraction_21600_21700', 'ice_area_fraction_21700_21800', 'ice_area_fraction_21800_21900', 'ice_area_fraction_21900_22000', 'ice_area_fraction_22000_22100', 'ice_area_fraction_22100_22200', 'ice_area_fraction_22200_22300', 'ice_area_fraction_22300_22400', 'ice_area_fraction_22400_22500', 'ice_area_fraction_22500_22600', 'ice_area_fraction_22600_22700', 'ice_area_fraction_22700_22800', 'ice_area_fraction_22800_22900', 'ice_area_fraction_22900_23000', 'ice_area_fraction_23000_23100', 'ice_area_fraction_23100_23200', 'ice_area_fraction_23200_23300', 'ice_area_fraction_23300_23400', 'ice_area_fraction_23400_23500', 'ice_area_fraction_23500_23600', 'ice_area_fraction_23600_23700', 'ice_area_fraction_23700_23800', 'ice_area_fraction_23800_23900', 'ice_area_fraction_23900_240
```

INSTALLATION USING MINIFORGE

Step 1: Install and activate miniforge <https://github.com/conda-forge/miniforge>

Step 2: Install satpy

```
mamba install satpy
```

Step 3: Install reader dependencies

```
mamba install matplotlib ipynb rioxarray bottleneck cartopy  
libgdal-jp2openjpeg s3fs
```

INSTALLATION ON THE EWC

- Use miniforge or pip
- All examples in this presentation generated using jupyterhub on the EWC
- Access to data using eumdac or EUMETCast Terrestrial
- Used by Pytroll members with SEVIRI and FCI workflows

EXAMPLES

SENTINEL 2 (I)

Lyon, 2025-08-17

```
import satpy
filenames = satpy.find_files_and_readers(base_dir='/mnt/data/
      S2', reader='msi_safe')
scn = satpy.Scene(filenames=filenames)
scn.load(['false_color', 'true_color'])
scn.save_datasets()
```

An aerial photograph showing a winding river through a landscape of agricultural fields. The fields are colored in shades of red and brown, indicating different vegetation indices. The river is dark blue. The text "SENTINEL 2 (II)" is overlaid in the center in a bold, white, sans-serif font.

SENTINEL 2 (II)

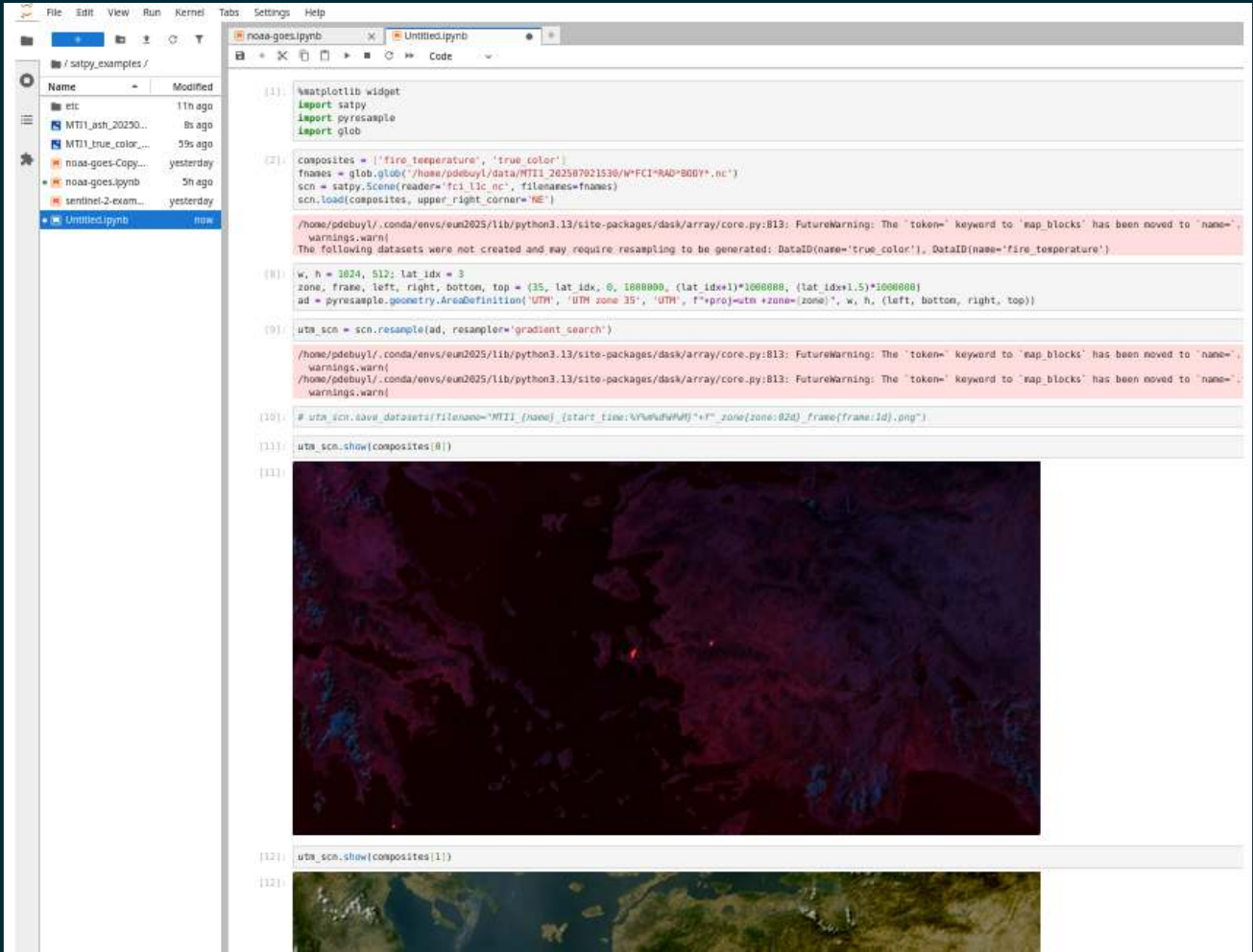
A satellite image showing a wide river valley. The river flows from the top center towards the bottom. The valley floor is a mix of urban development, appearing as a dense grid of grey and brown, and agricultural fields, which are in various shades of green and brown. The surrounding hillsides are covered in dense green vegetation. The text "SENTINEL 2 (III)" is overlaid in the center of the image.

SENTINEL 2 (III)

MTG FCI

- FCI reader in satpy used for EUMETSAT public imagery
- Access to the FCI decompressor via `hdf5plugin`
- Local area resampling
- Color composition
- Merging with Lightning Imager L2 data

MTG FCI NOTEBOOK



2025/07/02 IMAGE OF THE WEEK

```
[1]: %matplotlib widget
import satpy
import pyresample
import glob

[ ]: composites = ['fire_temperature', 'true_color']
fnames = glob.glob('/home/pdebuyl/data/MTI1_202507021530/W*FCI*RAD*BODY*.nc')
scn = satpy.Scene(reader='fci_llc_nc', filenames=fnames)
scn.load(composites, upper_right_corner='NE')

[8]: w, h = 1024, 512; lat_idx = 3
zone, frame, left, right, bottom, top = (35, lat_idx, 0, 1000000, (lat_idx+1)*1000000, (lat_idx+1.5)*1000000)
ad = pyresample.geometry.AreaDefinition('UTM', 'UTM zone 35', 'UTM', f"+proj=utm +zone={zone}", w, h, (left, bottom, right, top))

[ ]: utm_scn = scn.resample(ad, resampler='gradient_search')

[13]: utm_scn.save_datasets(filename="MTI1_{name}_{start_time:%Y%m%d%H%M}" + f"_zone{zone:02d}_frame{frame:1d}.png")

[11]: utm_scn.show(composites[0])

[11]:
```





LIGHTNING IMAGER

- Complex case: merging instruments and type of data
- The "LI Accumulated Flash Area" product stores the data per 30s chunks
- Reader `li_12_nc` in `satpy`
- "MultiScene" with `true_color_with_night_ir105_acc_flash_area`, and `true_color_with_night_ir105_acc_flash_area`

A satellite image of Earth showing a large storm system over the Atlantic Ocean. The storm is characterized by a dense, swirling cloud mass with a bright white center. Numerous small, bright yellow and red dots are scattered across the storm, representing lightning strikes. The surrounding ocean is dark, and the landmasses of North and South America are visible in the upper right. The text "LIGHTNING IMAGER" is overlaid in the center of the image.

LIGHTNING IMAGER

GOES 19: READING REMOTE DATA USING FSSPEC

```
import os ; os.environ['SATPY_CONFIG_PATH'] = 'etc'
import satpy
reader_kwargs = {'storage_options': {'s3': {'anon': True},
    'simplecache': {'cache_storage': '/tmp/s3_cache'} } }
yyyy = '2025' ; jday = 228 ; hour = '13' # 16 august 2025 at
    13:00
filenames =[f'simplecache::s3://noaa-goes19/ABI-L1b-RadF/{yyyy}/
    {jday:03d}/{hour}/OR_ABI-L1b-RadF-M6C*_s{yyyy}{jday:03d}
    {hour}0*.nc']
composite = 'night_ir_with_color'
scn = satpy.Scene(reader='abi_l1b', filenames=filenames,
    reader_kwargs=reader_kwargs)
scn.load([composite])
```

File Edit View Run Kernel Tabs Settings Help

Launcher x noaa-goes.ipynb x +

+ < > < > < > < > Code <

Notebook

•[1]:

```
import os, itertools
os.environ['SATPY_CONFIG_PATH'] = 'etc'
import satpy
import s3fs
import pyresample
reader_kwargs = {'storage_options': {'s3': {'anon': True}, 'simplecache': {'cache_storage': '/tmp/s3_cache'}}}
```

•[2]:

```
yyyy = '2025' ; jday = 228 # 16 august
hour = '13'
filenames = [f'simplecache::s3://noaa-goes19/ABI-L1b-RadF/{yyyy}/{jday:03d}/{hour}/OR_ABI-L1b-RadF-M6C*_s{yyyy}{jday:03d}{hour}0*.nc']
noaa_bucket = s3fs.S3FileSystem(anon=True)
composite = 'night_ir_with_color'
if len(noaa_bucket.glob(filenames[0].split(':')[1])) == 0:
    print("No file found")
scn = satpy.Scene(reader='abi_l1b', filenames=filenames, reader_kwargs=reader_kwargs)
scn.load([composite])
```

The following datasets were not created and may require resampling to be generated: DataID(name='night_ir_with_color')

•[4]:

```
# Caribbean area coordinates N 31 W -84 S 11 E -54
projection = {'proj': 'latlong', 'datum': 'WGS84'}
ad = pyresample.AreaDefinition("eqc", "lat/lon", "eqc", width=900, height=600,
                               projection=projection, area_extents=(-84, 11, -54, 31))

sub_scn = scn.resample(ad, resampler='gradient_search')
sub_scn.show(composite)
```

/home/pdebuyl/.conda/envs/eum2025/lib/python3.13/site-packages/dask/_task_spec.py:759: RuntimeWarning: invalid value encountered in log
return self.func(*new_argspec)

[4]:



GOES 19

- Files are saved to disk and animated with ffmpeg

```
ffmpeg -f image2 -pattern_type glob -framerate 6\  
-i 'night_ir_with_color_202508*.png' -pix_fmt yuv420p\  
-strict -2 G19_CARIBBEAN_20250816.mp4
```

GOES 19

REFERENCES & ACKNOWLEDGMENTS

- Many thanks to the PyTroll community for their help
- Satpy <https://satpy.readthedocs.io/>
- Satpy <https://github.com/pytroll/satpy>
- PyTroll <https://pytroll.github.io/>
- <https://www.eumetsat.int/image-week-wildfires-eastern-mediterranean>
- [https://en.wikipedia.org/wiki/Hurricane_Erin_\(2025\)](https://en.wikipedia.org/wiki/Hurricane_Erin_(2025))