

Welcome

WORKSHOP ON

30-31 OCTOBER 2018

Developing Python
frameworks for earth system
sciences



One year on ... where are we?

What we learned then and done since

Stephan Siemen
Development Section, ECMWF

```
jupyter Vertical profile Last Checkpoint: 3 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
+ < > Run C Code

In [11]: import mpy.metview as mpy
         from ipywidgets import interact, interactive, FloatSlider, IntSlider

In [2]: t_fc24 = mpy.read('t_fc24.grib')

In [9]: def plot_vprof_at_point(lat, lon):
         vprofview = mpy.mvertprofview(point = [lat,lon])
         g = mpy.mgraph(graph_line_thickness = 4, graph_line_style = 'dash')
         print('Vertical profile at point: lat=', lat, ' lon=', lon)
         im = mpy.plot(vprofview, t_fc24, g)
         display(im)
         return im

In [10]: interactive_plot = interactive(plot_vprof_at_point, lon=FloatSlider(
output = interactive_plot.children[-1]
output.layout.height = '450px'
output.layout.width = '450px')
```

The screenshot displays the OpenMeteo web application interface. At the top, there are logos for the European Union, Copernicus, ECMWF, and Climate Change Service. The user is identified as Sylvie Lamy-Thepaut. The main navigation bar includes Home, Search, Datasets, Applications, Your requests, Toolbox, and Help & support. The interface is divided into three main sections:

- Toolbox editor:** On the left, there is a sidebar with 'Applications', 'Data', and 'Documentation' tabs. Under 'Applications', there is a search bar and a list of tools categorized into 'your workspace' (01 Retrieve Data, 02 Plot Map) and 'examples' (00 Hello World, 01 Retrieve data, 02 Plot map, 03 Extract time series and plot graph, 11 Calculate time mean and standard deviation, 12 Calculate climatologies, 21 Calculate regional mean and anomalies, 31 Calculate trends, 41 Calculate GDD, 42 Use cdo functions, 51 Calculate zonal means, 52 Format maps to allow visual comparison).
- Code editor:** The central area shows a code editor with Python code for the '02 Plot Map' application. The code imports 'cdstoolbox as ct' and defines a 'plot_map' function that retrieves data from the 'reanalysis-era5-single-levels' dataset for a specific variable, year, and month.
- Plot Map:** On the right, there is a 'Plot Map' section. It features a world map showing 'Mean Near-Surface Air Temperature in 2008-01'. A color scale on the right indicates temperatures from -40 to 40 degrees Celsius. Below the map, there are interactive dropdown menus for 'variable' (set to '2m_temperature'), 'year' (set to '2008'), and 'month' (set to '1').

At the bottom right of the interface, the version is noted as 'Version: 3.2.3 - build e42057a'.

What we learned from last year's workshop

- There are already many good efforts and solutions out there
 - Many good “wheels” which do not need to be reinvented
- Confirmation of our direction for developments
 - Handle fields through *xarray*
- Building a community is more than just releasing software under Open Source
 - ‘Open Source’ versus ‘Open Development’



CDS (toolbox) release earlier this year



- New portal to find / download and work with Copernicus Climate data

- Try it out yourself:

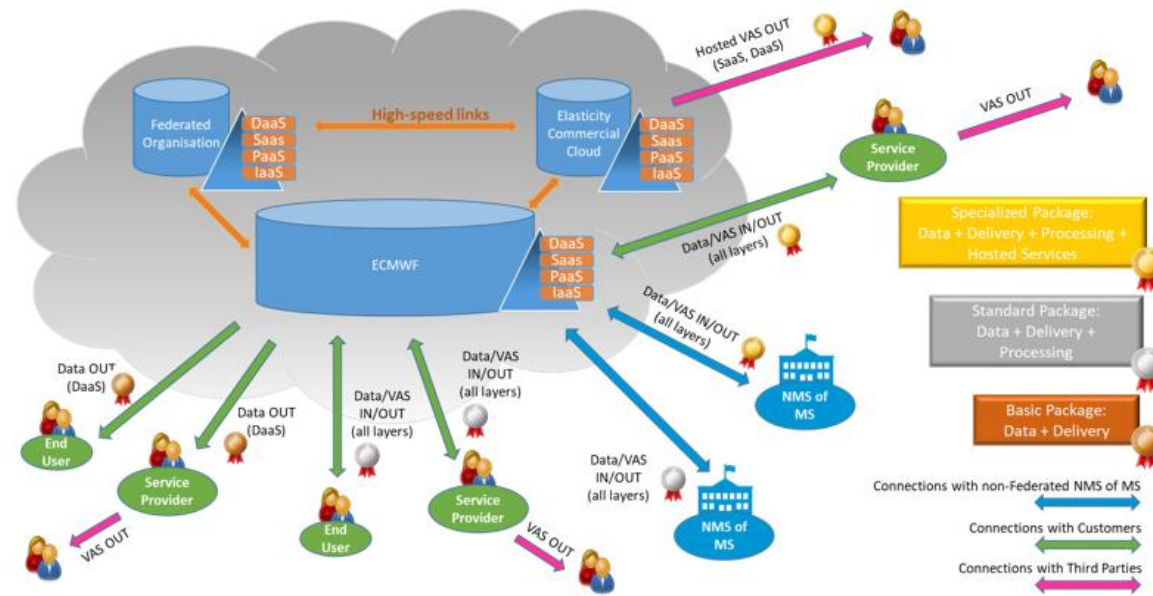
<https://cds.climate.copernicus.eu>

```
1 import cdstoolbox as ct
2
3
4 @ct.application(title='Calculate GDD')
5 @ct.input.dropdown('year', values=range(2008, 2018))
6 @ct.input.text('tas_min_celsius', type = float, default=10.)
7 @ct.output.figure()
8 def gdd_app(
9     year,
10    tas_min_celsius,
11 ):
12     ---
13     Application main steps:
14
15     - retrieve a variable over a defined time range
16     - compute the daily mean
17     - compute the "Growing Degree Days" index
18     - compute its mean over time dimension (over all timesteps)
19     - show the result on a map
20     ---
21
22
23 data = ct.catalogue.retrieve(
24     'reanalysis-era5-single-levels',
25     {
26         'variable': '2m_temperature',
27         'grid': ['3', '3']
28     },
29     'product_type': 'reanalysis',
30     'year': str(year),
31     'month': [
32         '01', '02', '03', '04', '05', '06',
33         '07', '08', '09', '10', '11', '12'
34     ],
35     'day': [
36         '01', '02', '03', '04', '05', '06',
```



New opportunities through cloud services

- ECMWF looks together with its partners on providing private clouds close to data
 - *European Weather Cloud* with EUMETSAT
 - Copernicus *WEkEO* DIAS in co-operation with EUMETSAT & Mercator Ocean
- Making it easier for users to work with ECMWF forecast & Copernicus data
 - And Python will play an important role here ...



Make software easily available within existing frameworks

- Source code & examples on GitHub (CaaS)
- Packages need be on PyPi and Conda
- ECMWF Python software on DockerHub
- CDS Toolbox - SaaS

The collage consists of four overlapping screenshots:

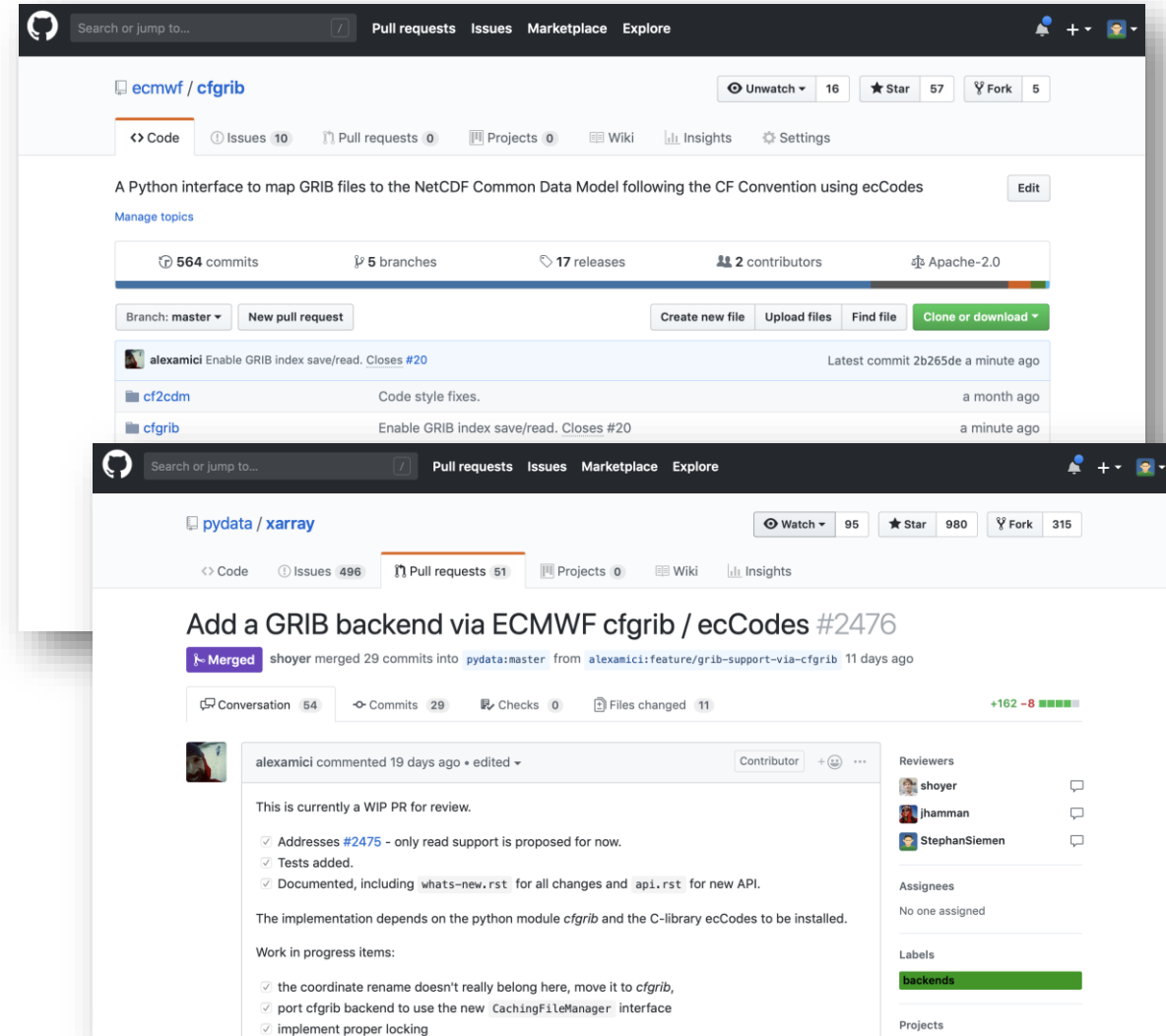
- Top Left:** A GitHub repository page for 'cfrib 0.9.2' by the European Centre for Medium-Range Weather Forecasts. It shows the repository name, version, and a 'pip install cfrib' command.
- Top Right:** A DockerHub repository page for 'ecmwf/jupyter-notebook'. It shows the repository name, version, and a 'docker pull ecmwf/jupyter-notebook' command.
- Bottom Left:** A code editor showing Python code for calculating climatology. The code uses the CDS toolbox to retrieve data and calculate the mean and standard deviation of near surface air temperature over a 12-month period.
- Bottom Right:** A web interface for 'Calculate climatologies'. It shows a line chart of 'Near Surface Air Temperature climatological mean (°C)' over 12 months. The chart shows a seasonal cycle with a minimum in winter and a maximum in summer. The interface includes input fields for 'variable', 'frequency', 'month', and 'start'.

We need to improve on the first generation Python interfaces

- SWIG is very limited
 - Support Python3 with all our packages by end of 2018
 - Need to support multiple Python version at once
 - Very close to Fortran/C interfaces
 - Not taken into account any Pythonic ideas
- Consequence: split Python interfaces from software packages
- Package in PyPi

cfgrib – linking xarray and ecCodes

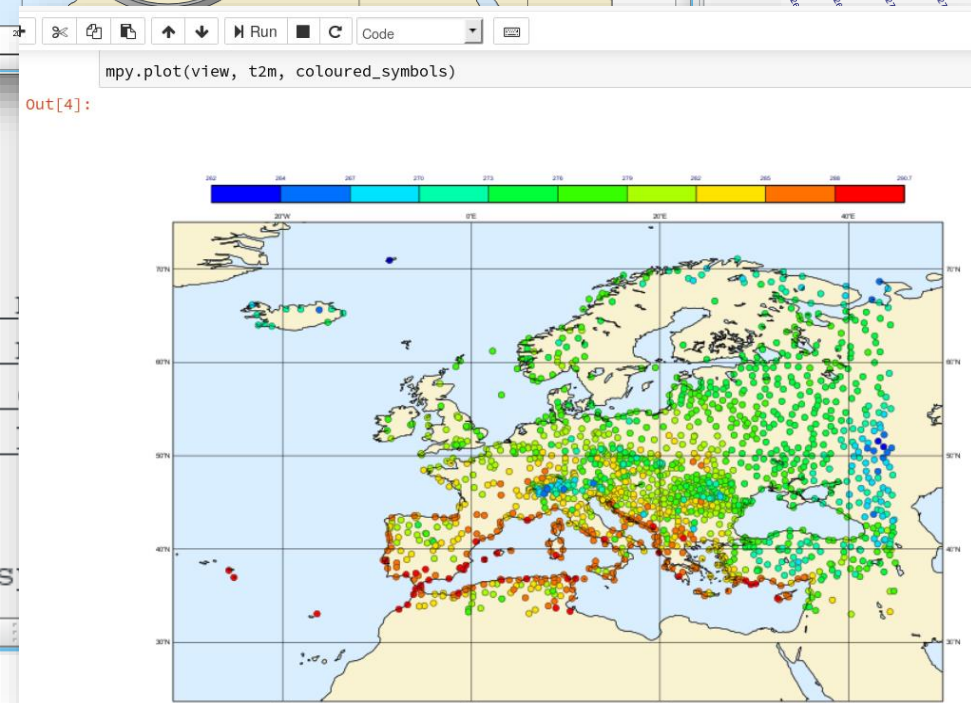
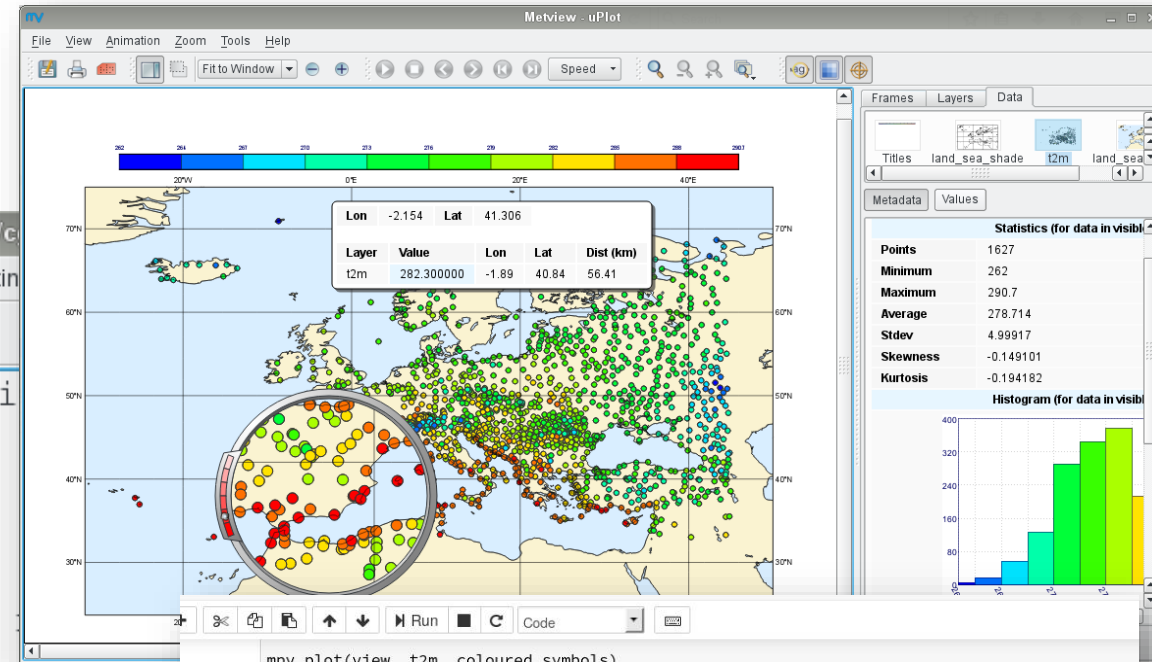
- To embrace xarray for all our field data, we needed to know that we could handle all our GRIB 1 & 2 data
 - Therefore it was important for us to have a solution based on ecCodes
- This work allows users of GRIB data now to have a high level access of our data and use all PyData community has to offer



The Metview Python framework

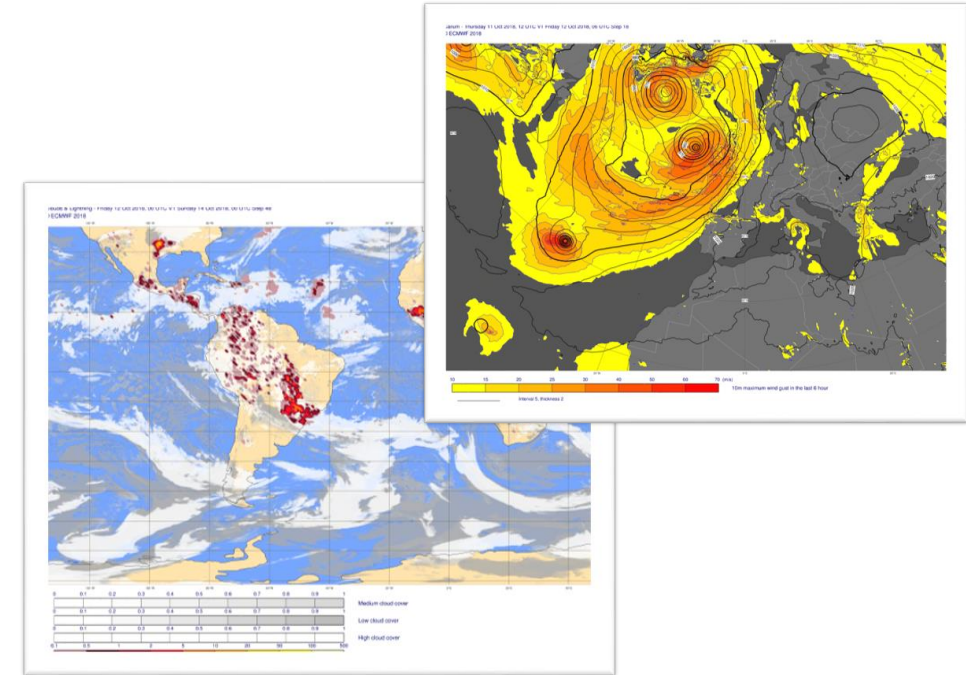
- Aim is to allow users of Metview to use easily the power of Python but still have all functionality of Metview; including visualisation
- We have now beta release with all Metview functionality available from Python
- Close co-operation CDS toolbox

```
Python Script.py* - /home/graphics/c
File Edit View Insert Program Settin
18 map_area_defini
19 area
20 coastlines
21 )
22
23 coloured_symbols =
24 legend
25 symbol_type
26 symbol_table_mode
27 symbol_outline
28 symbol_outline_colour
29 symbol_advanced_table
30 symbol_advanced_table
31 symbol_advanced_table
32 symbol_advanced_table
33 )
34
35 plot(view, t2m, coloured_s
```



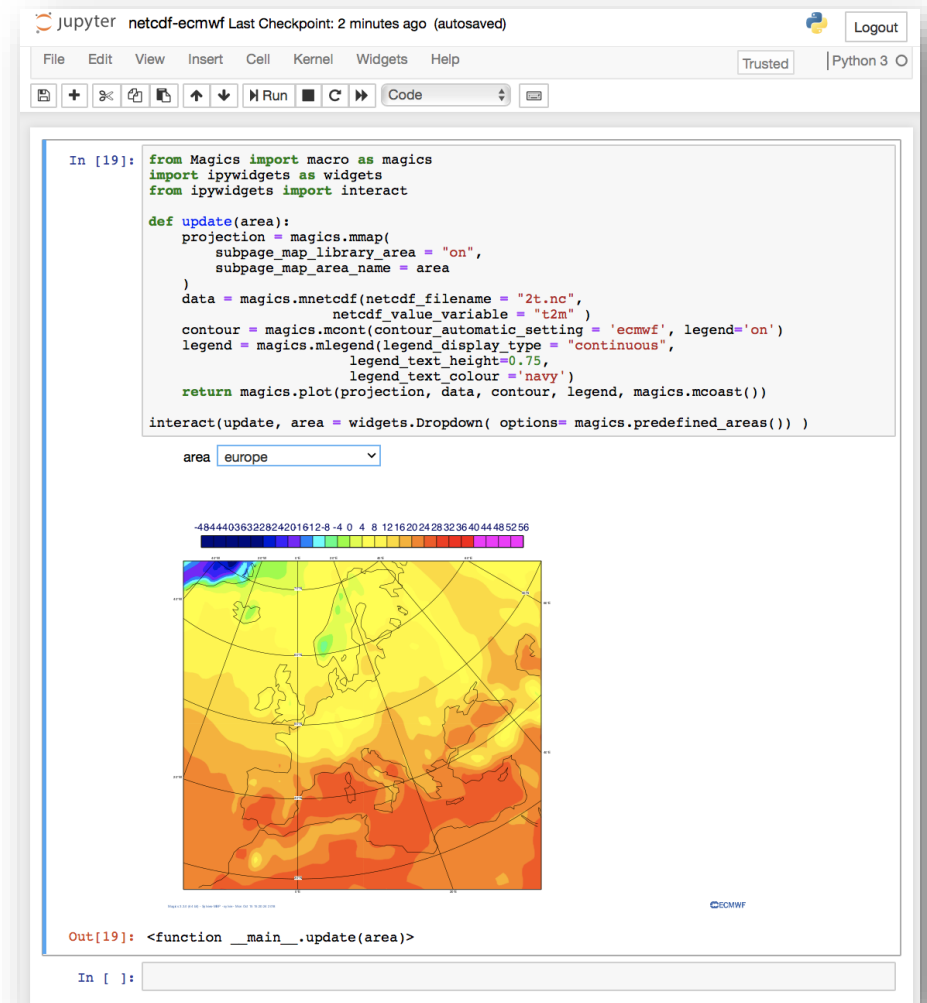
Improve automatic styling

- With more distributed (cloud) environments, it is even more of a challenge to keep a common (corporate) look-and-feel
- We need to make easy for users to apply these styles
 - Best automatic
 - Take meta data information from xarray
 - Detect internal representation of (NetCDF) data to perform geo referencing



Challenges ahead ...

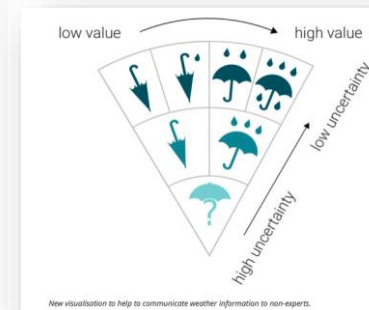
- Would be nice if we could provide examples of our frameworks working together
 - E.g. Metview with MetPy
- What is the best way to offer a user environment? (especially in the cloud)
 - Jupyter/Lab?
 - Docker?
 - SaaS – JupyterHub / CDS toolbox?
- Reproducible workflows → Workshop @ECMWF in 2019
- How we can scale heavy processing tasks
 - Dask & Pangeo ... especially in the cloud



ECMWF Summer of Weather Code (ESoWC)



Innovative visualisations



New visualisation to help to communicate weather information to non-experts.

Enhanced GIS widget



Enhanced GIS widget to select and display areas on a map.

- First year was a great success to engage with new communities
- We look for new ideas by January 2019 ...
... perhaps our discussions will come up with some ideas ...