

From Metview to Earthkit

Enabling fast and easy access to ECMWF's forecasts

Development Section, Forecast Department, ECMWF

Metview

Metview is a meteorological workstation software package for the purpose of **accessing, inspecting, processing** and **visualising** meteorological data. It combines interactive and batch usage, making it suitable for research and operations.

Metview's development was originally a co-operative project between ECMWF and INPE (Brazil) with assistance from Météo-France.

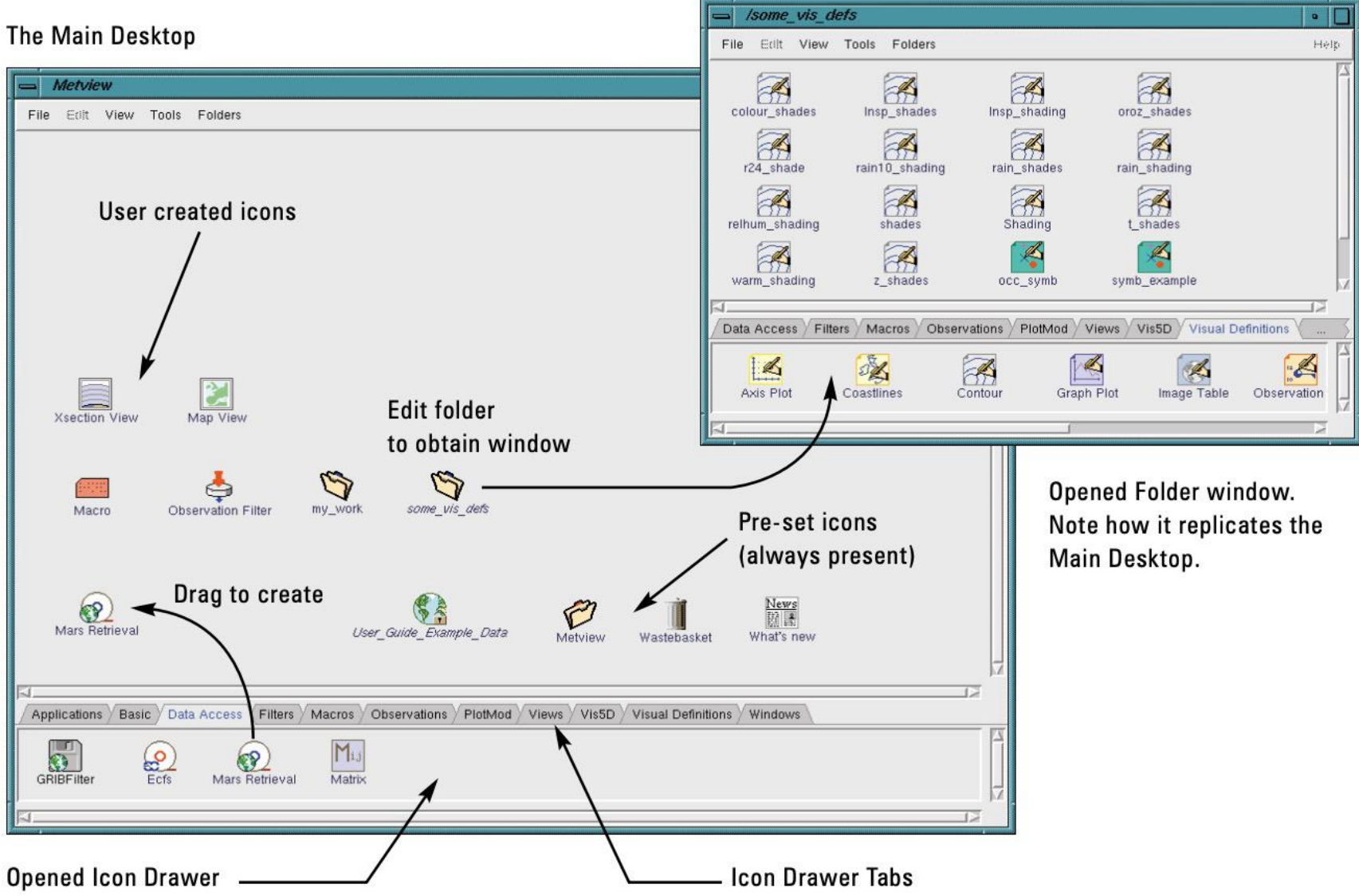
1990

Metview's development was announced in June 1990

Metview
There are plans to develop a general and unique system for the visualization of meteorological data at ECMWF which should serve the scientist and the operational analyst alike. The Metview concept will provide a standard framework within which applications relating to the retrieval, processing and visualization of meteorological data can be implemented, and will enable both Operations and research

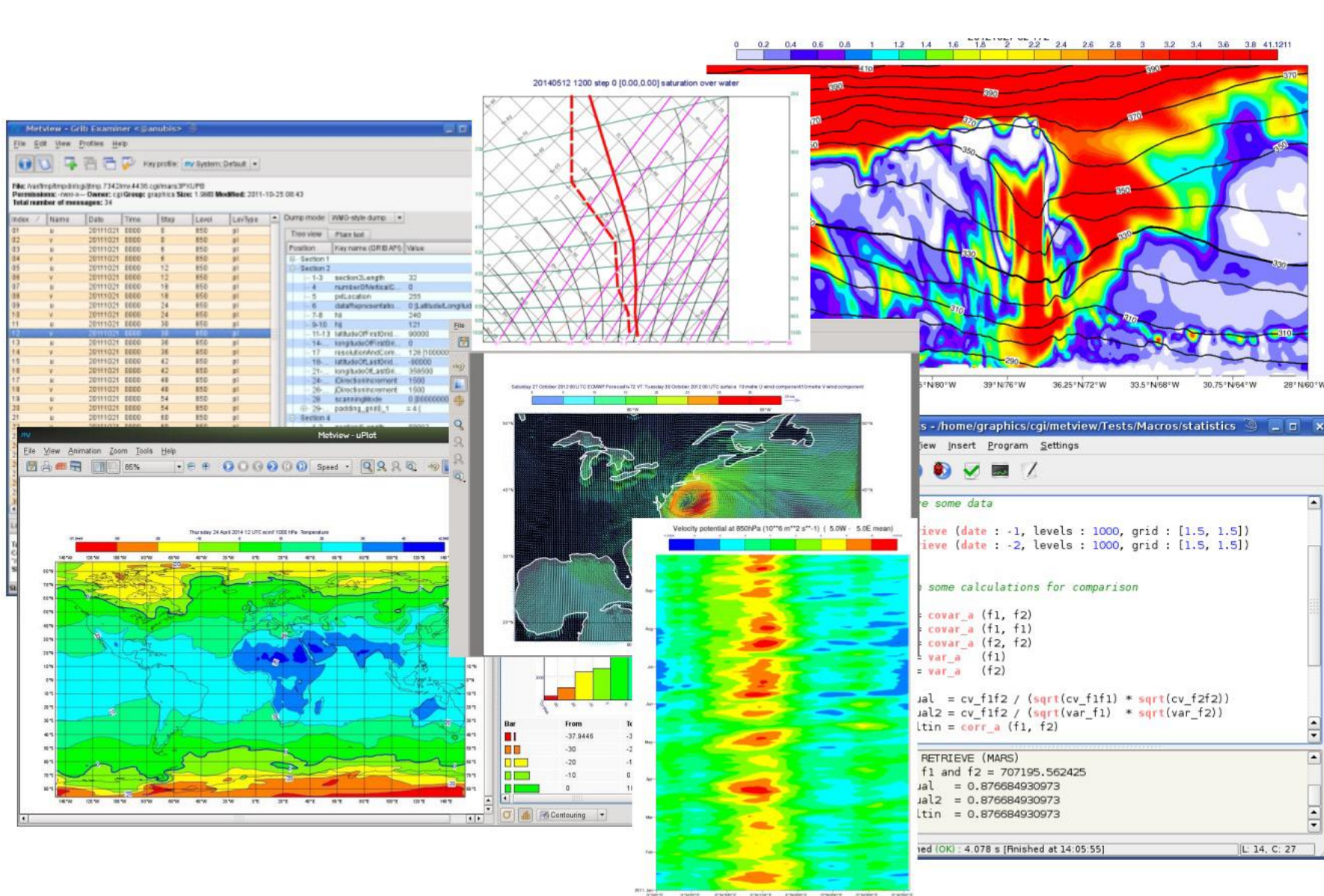
1993 / 1993 / 1998

Metview versions 1.0, 2.0 and 3.0 are released with progressively advanced technologies



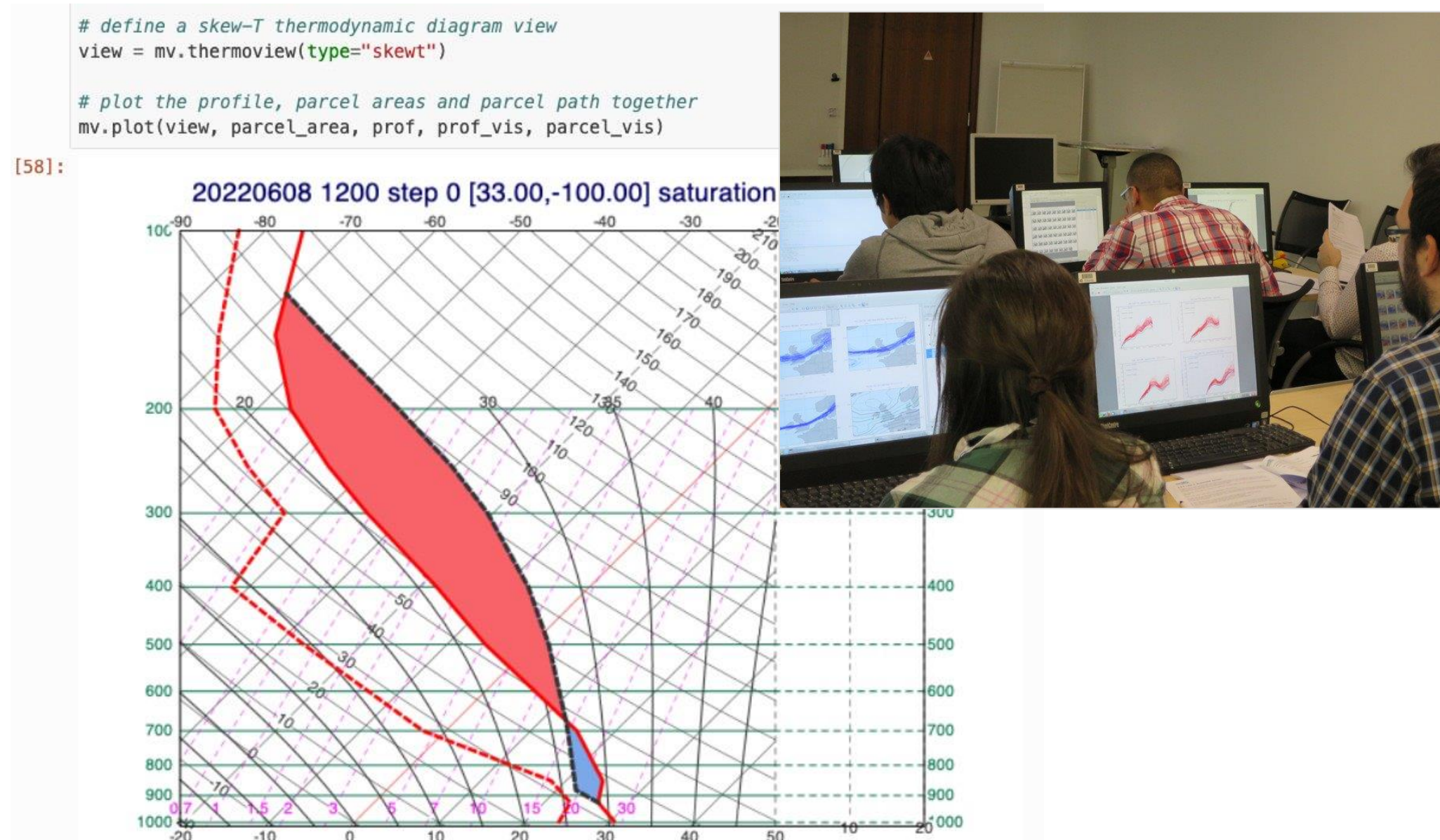
2012

Metview version 4.0 is released as Open Source, and with a new user interface



2017

Metview version 5.0 is released with a Python interface



2023

ECMWF's Software Strategy and Roadmap is published, emphasising Open Development, greater reusability and componentisation of software, and scalability on modern hardware. Those principles, merged with the philosophy of Metview, formed the basis of Earthkit.

Earthkit

Earthkit is a new set of high-level Python packages that work together to provide an ecosystem suitable for research and operations. Also providing functionality for data access, analysis, processing and plotting, Earthkit will be the natural successor to Metview.



```
import earthkit as ek

ds_fc = ek.data.from_source("sample", "fc_storm_st_jude.grib") # hi-res forecast
ds_en = ek.data.from_source("sample", "ens_storm_st_jude.grib") # ensemble forecast

# select wind gust fields and convert them to Xarray
ds_fg = ds_en.sel(param="10fg3")
fg = ds_fg.to_xarray()

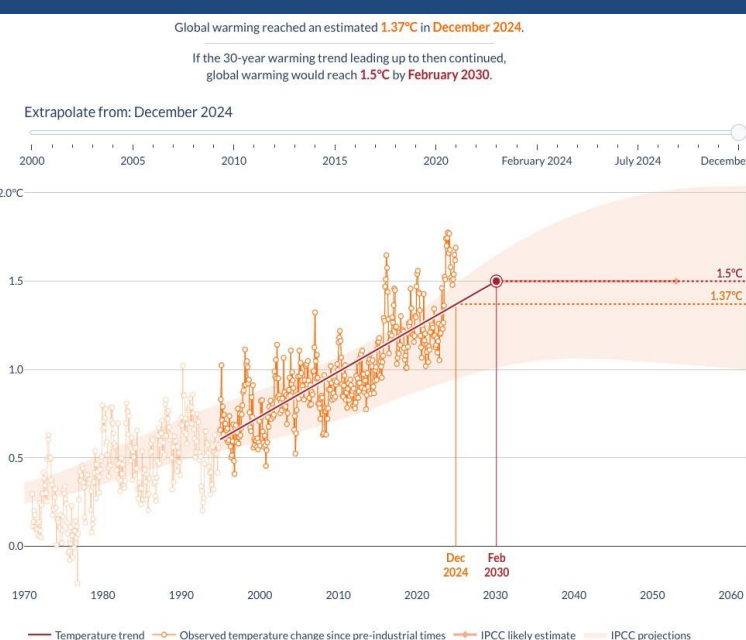
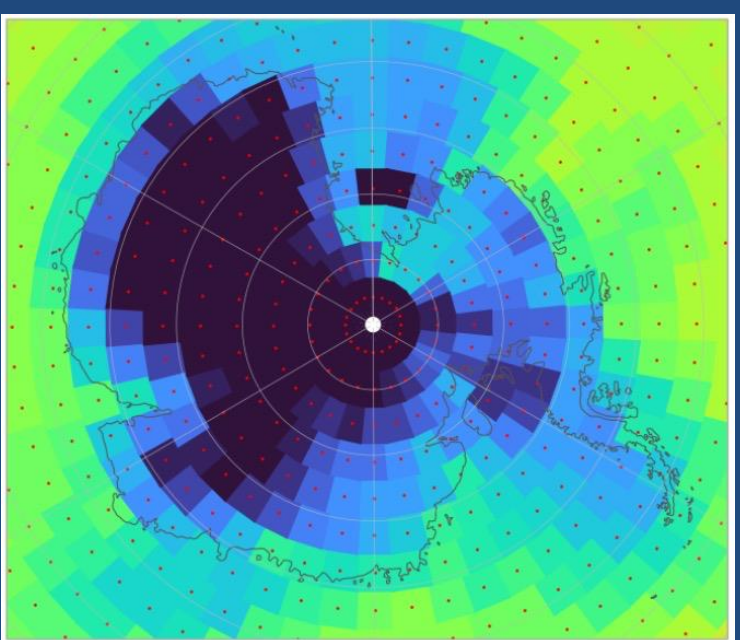
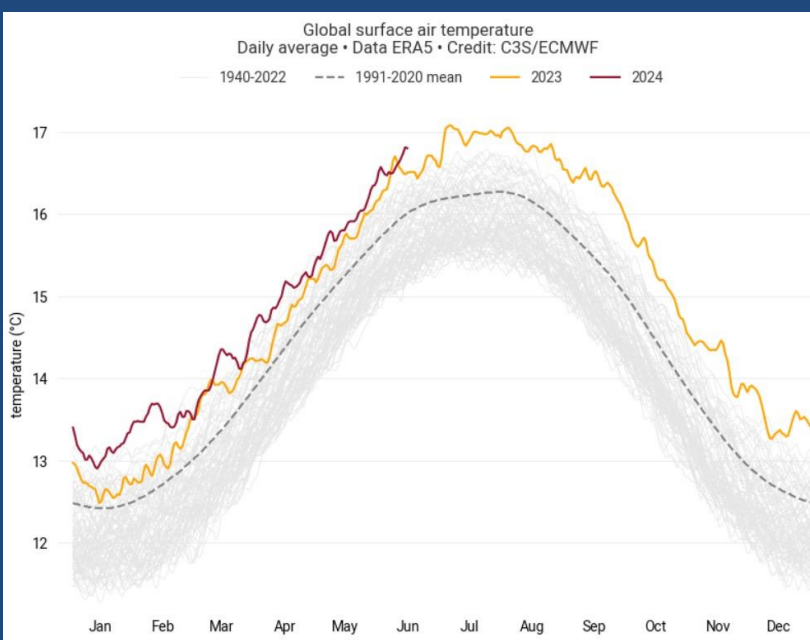
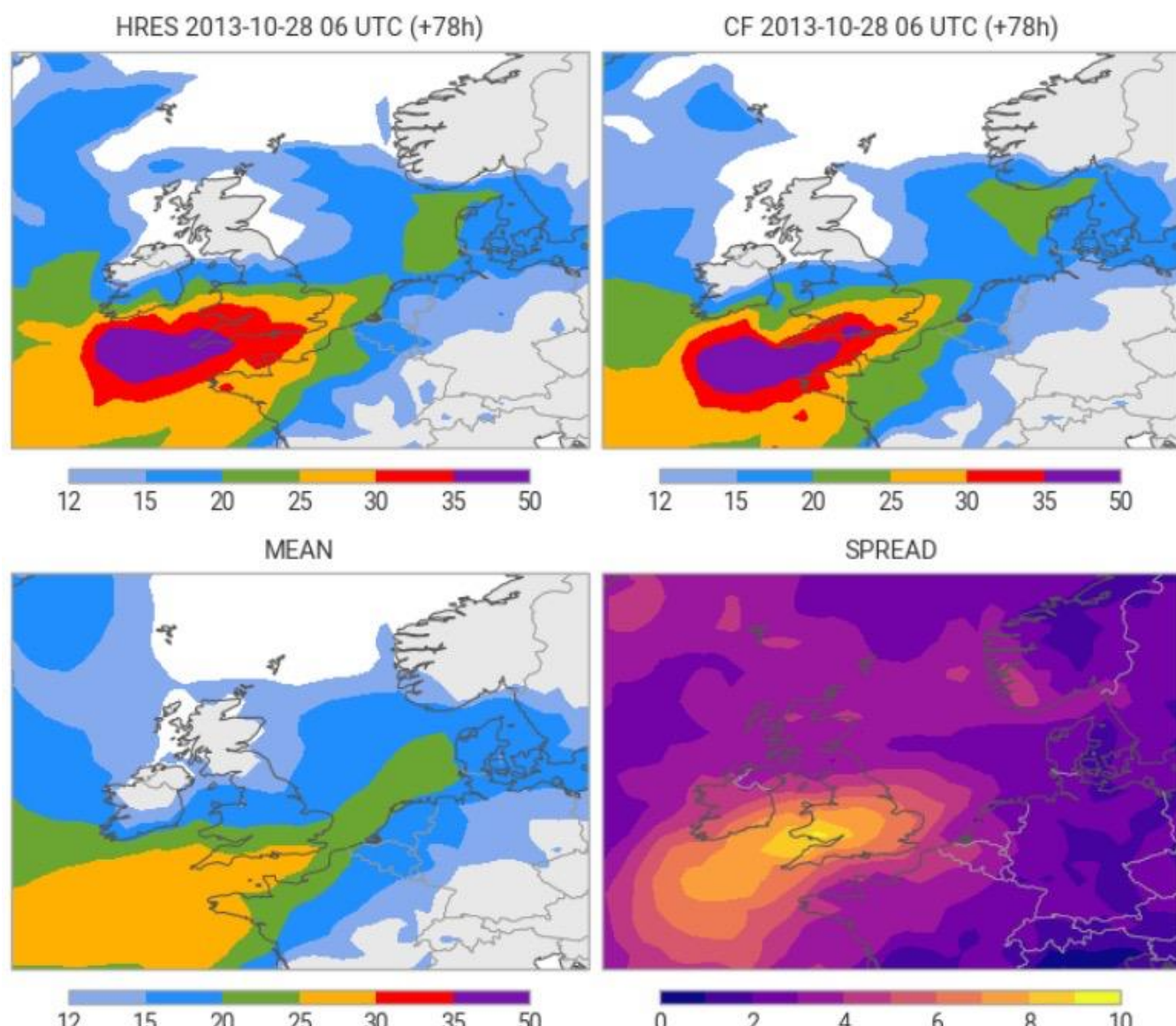
# compute ensemble mean and spread
fg_mean = ek.transforms.ensemble.mean(fg)
fg_spread = ek.transforms.ensemble.std(fg)

import cartopy.crs as ccrs
import datetime

# define step
step = 78

figure = ek.plots.Figure(crs=ccrs.PlateCarree(),
                        domain=[-15,15,65,45], size=(7, 6), rows=2, columns=2)

gust_style = ek.plots.styles.Style(
    colors=["#B5AAEE", "#288BFC", "#6CA632", "#FFB000", "#FF0000", "#7A1181"],
    levels=[12, 15, 20, 25, 30, 35, 50],
    units="m s-1",
)
```



Feature	Metview	Earthkit
Development started	1991	2023
Programming language	C++ & Python	Python (uses underlying C/C++ ECMWF stack)
Computations	CPU	CPU & GPU
Temporary data storage	Disk	Memory & Disk
Field input types	GRIB	GRIB, NetCDF, Xarray, Numpy
Data sources	File, MARS, ECFS, URL	File, MARS, URL, S3, FDB, ECFS, CDS, ADS, Polytope, ...
Code organisation	Monolithic	Package per category of functionality
Plotting	Based on Magics (bespoke C++)	earthkit-plots uses Matplotlib