

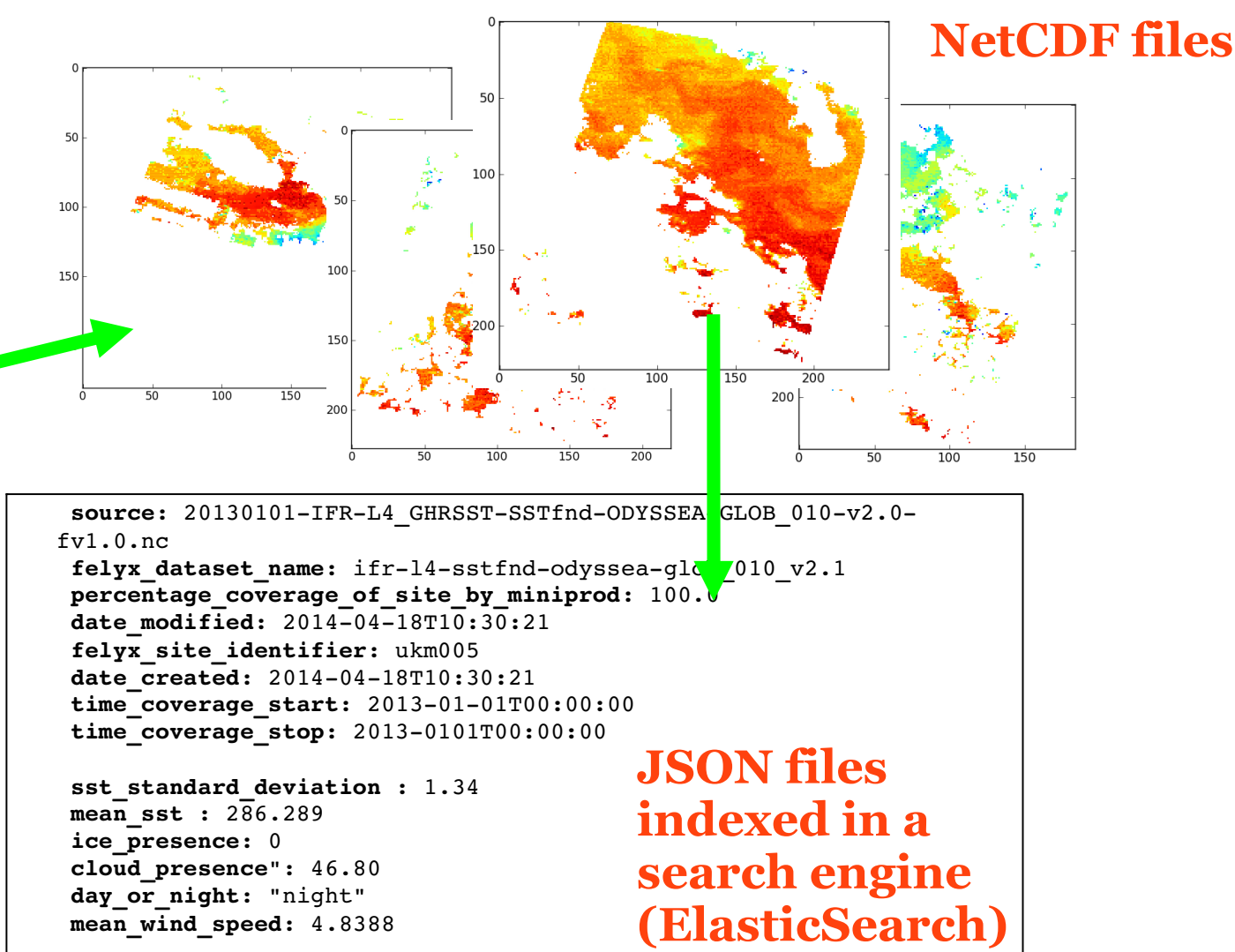
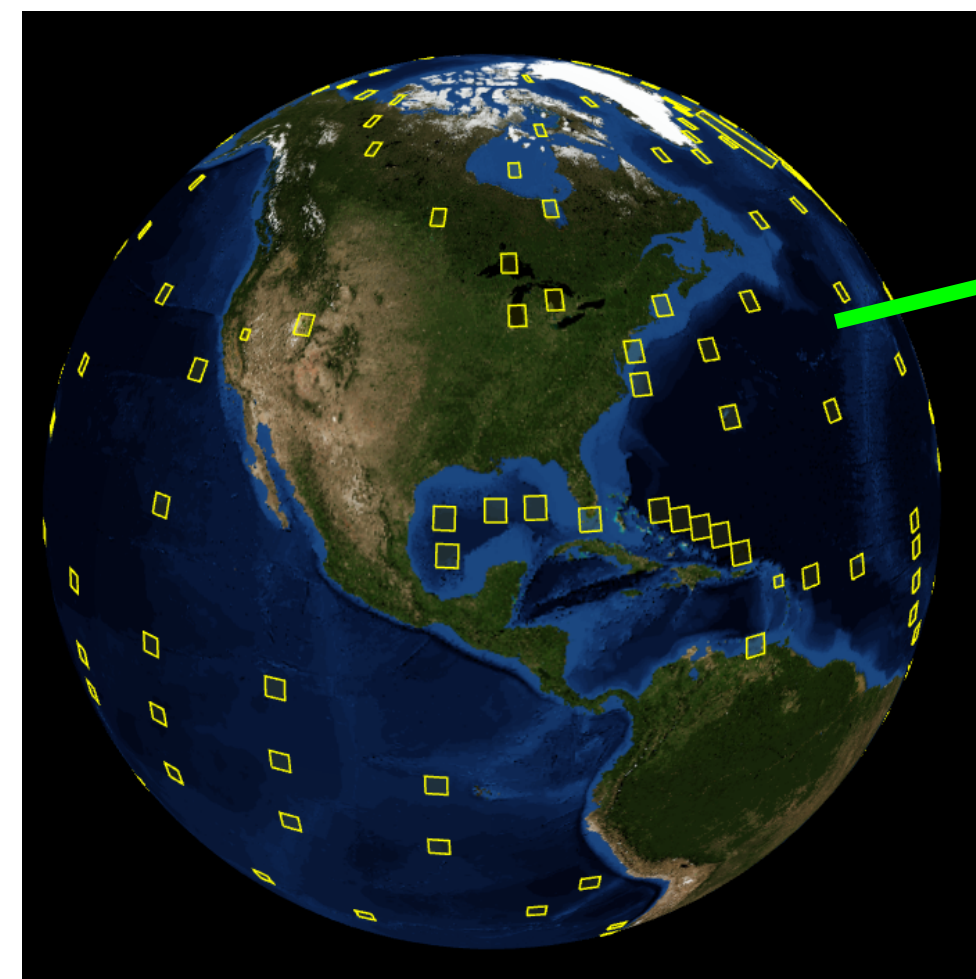


Jean-François Piollé (Ifremer, jfpiolle@ifremer.fr),
Sylvain Herlédan (OceanDataLab), Jamie Shutler (University of Exeter), Peter Walker (PML), Dave Poulter (Pelamis), Philippe Goryl (ESA), Craig Donlon (ESA), Veronica Guidetti (ESA)

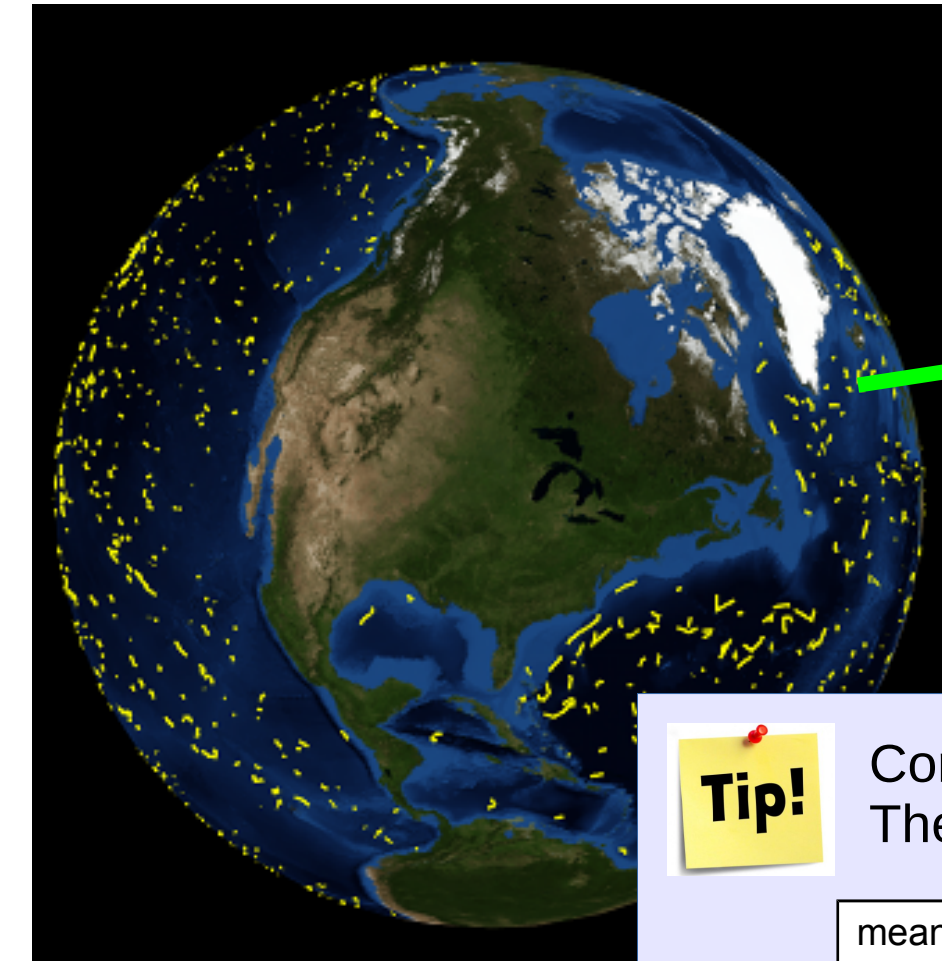
felyx as an extraction tool

In order to allow long term analysis or monitoring of voluminous EO datasets, felyx extract subsets (« **miniprods** ») from the original full size files. These subsets can be extracted over **static** (fixed location, polygonal shape) or **dynamic** sites (moving target such a drifting buoy or a hurricane), grouped into user or community **site collections**. felyx is domain agnostic (ocean, land, atmosphere,...) and can extract miniprods from swath, 2d or 3d grids, trajectory or along-track observation data, in a wide range of input formats (extensible through plugins). All miniprods are saved with all their fields in a unique netCDF4 format, in their native projection and resolution.

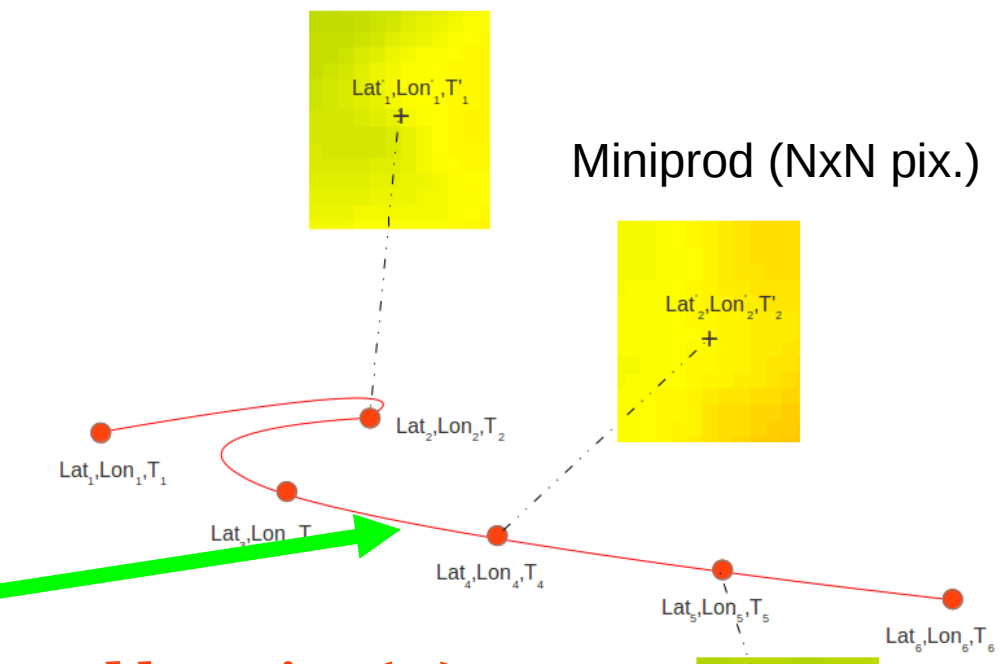
Statistical, qualitative or quantitative metrics can be computed over the data content of each extracted miniprods. They are stored in a nosql database (ElasticSearch) to allow query by users or analytics tools.



The trajectory files defining the dynamic sites are ingested into felyx through import web service (CSV file)



The extracted box size (N), colocation radius, maximum temporal difference to the trajectory point can be adjusted for each dataset independently



felyx data access

Let the fun begin ! The miniprods and metrics generated can later be queried and analysed in different ways through **felyx front-end API**.

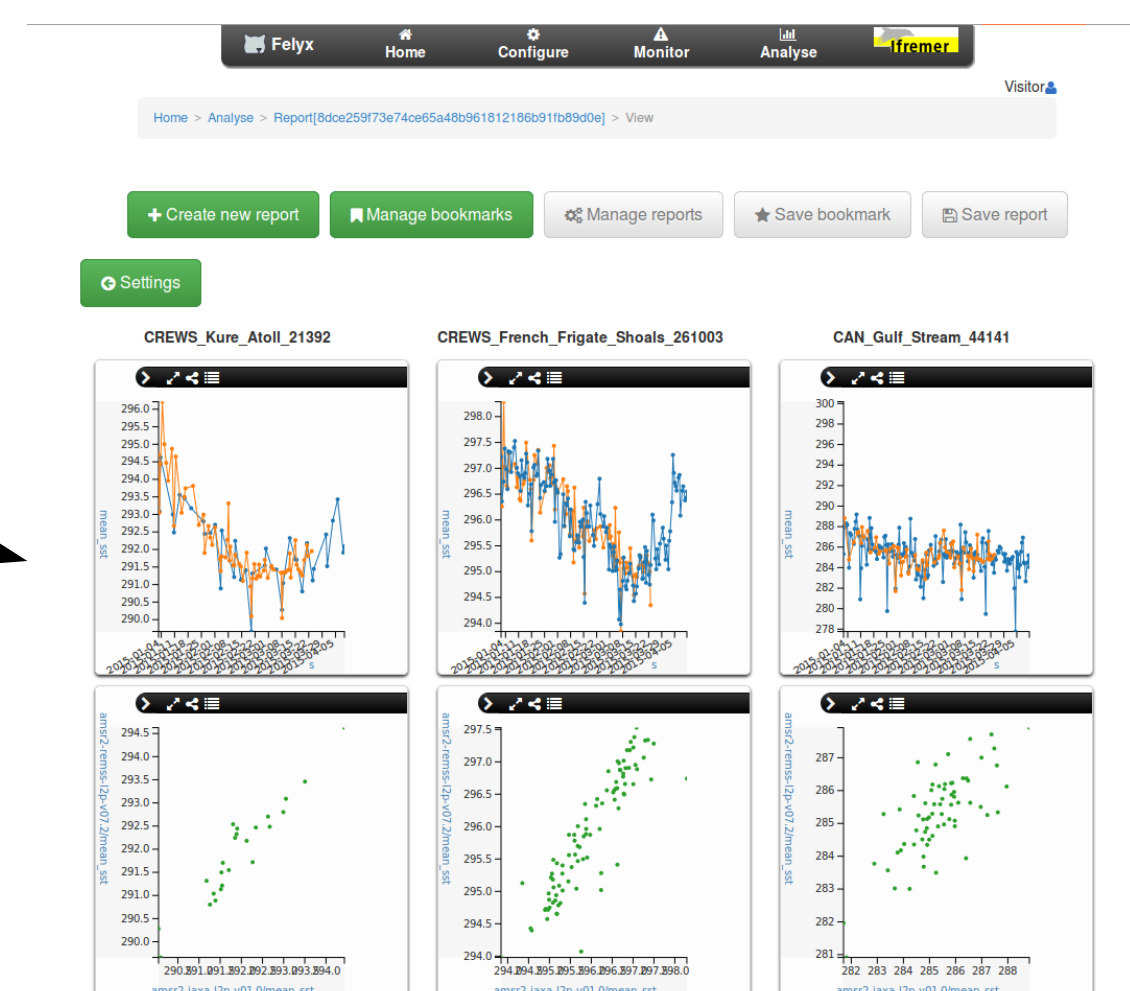
```
curl -XPOST felyx.cersat.fr/extraction/extraction/ -d '{
  "extraction": {
    "selection": {
      "dataset_list": ["arc-ups-l2p-water-v2.1"],
      "start_time": "2002-01-01T00:00:00",
      "stop_time": "2012-04-30T00:00:00",
      "site_list": ["TEKK000", "LYGH001"],
      "metric_list": ["mean_sst"]
    }
  }
}
```

RESTful API : request data in single http request (json query), from any language script/program.

```
from felyx_instance import Instance
from felyx_instance import Selection, Metrics

# instantiate felyx instance to query from (here felyx)
inst = Instance(url="http://www.ifremer.fr/cersat/fr/felyx/")
# instantiate a selection query
result = Selection(
    start="2002-01-01T00:00:00",
    stop="2012-04-30T00:00:00",
    dataset_list=["arc-ups-l2p-water-v2.1"],
    metric_list=["mean_sst"],
    site_list=["TEKK000", "LYGH001"],
    start_datetime="2002-01-01T00:00:00",
    end_datetime="2012-04-30T00:00:00"
)
# perform query
res = inst.metrics.select(
    inst.selection
)
# from here you can start working with the returned data
# ...
# print columns result
Metrics.data_metrics_display(res)
```

Python client API (pyfelyx) – a more user friendly way to express simple queries from your analysis and plotting scripts.



Felyx dashboard – use the built-in felyx dashboard to explore, build and share your own set of diagnostics.

iPython notebook – a more pleasant ride with a more user friendly way to express simple queries

Tip! Complex metrics can be defined, using for instance conditions on ancillary fields. They can also be computed on the fly from the miniproduct through the query API.

mean sea surface temperature (quality >= acceptable)	mean({"field": "sea_surface_temperature", "must_have": [{"operator": "greater_equal", "field": "quality_level", "value": 4}]})
mean sses bias (quality >= acceptable)	mean({"field": "sses_bias", "must_have": [{"operator": "greater_equal", "field": "quality_level", "value": 4}]})
mean sses standard deviation (quality >= acceptable)	mean({"field": "sses_standard_deviation", "must_have": [{"operator": "greater_equal", "field": "quality_level", "value": 4}]})
mean wind speed	mean({"field": "wind_speed"})
day or night status	day_or_night({})
sea surface temperature standard deviation (quality >= acceptable)	standard_deviation({"field": "sea_surface_temperature", "must_have": [{"operator": "greater_equal", "field": "quality_level", "value": 4}]})
ice presence	ice_presence({})

Miniprods can be access through **FTP** and **OpenDAP** like any data file (depending on host policy)

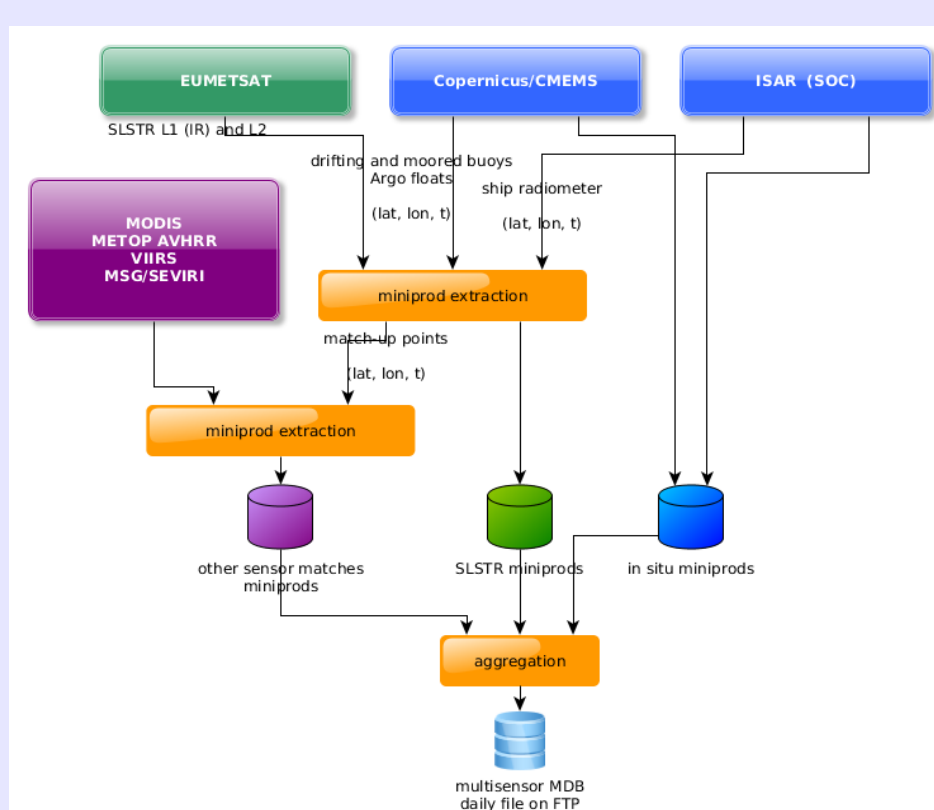
The API allows to **query, process and display programmatically** metrics and miniprods (from your own scripts) : **advanced workflows** can be built, including steps such as reprojection, differences, on the fly processing of new metrics,...

Federated queries can be performed across different felyx server : compare for instance metrics from different datasets served by different providers.

External on-the-shelf **analytics** can be used also, like those provided by ElasticSearch community (such as kibana). They can also automatically detect anomalies and raise alerts. We will investigate this in future to take advantage of these big data technologies.

felyx in action

The Sentinel-3 / SLSTR Match-Up Database



- Colocation window : 2h, 5km
- 21 x 21 pixel boxes
- ± 6h of in situ data history

In situ data :

- Copernicus/CEMS (Coriolis)
- ISAR radiometer on opportunity ships (delayed-mode)

Sentinel-3 data :

- L1 infra-red channels
- L2 (SST) – all fields, incl. meteo and ancillary fields

Other sensor data : Metop-B/AVHRR, MODIS, VIIRS, MSG/SEVIRI

The **Sentinel-3 SLSTR match-up database** is an activity funded by EUMETSAT in the context of OSI-SAF to support the cal/val.

The European Copernicus Marine Service provides a NRT in situ data flow (moorings, drifters and Argo float) adapted for Sentinel-3 cal/val. These data are routinely collected and filtered for felyx.

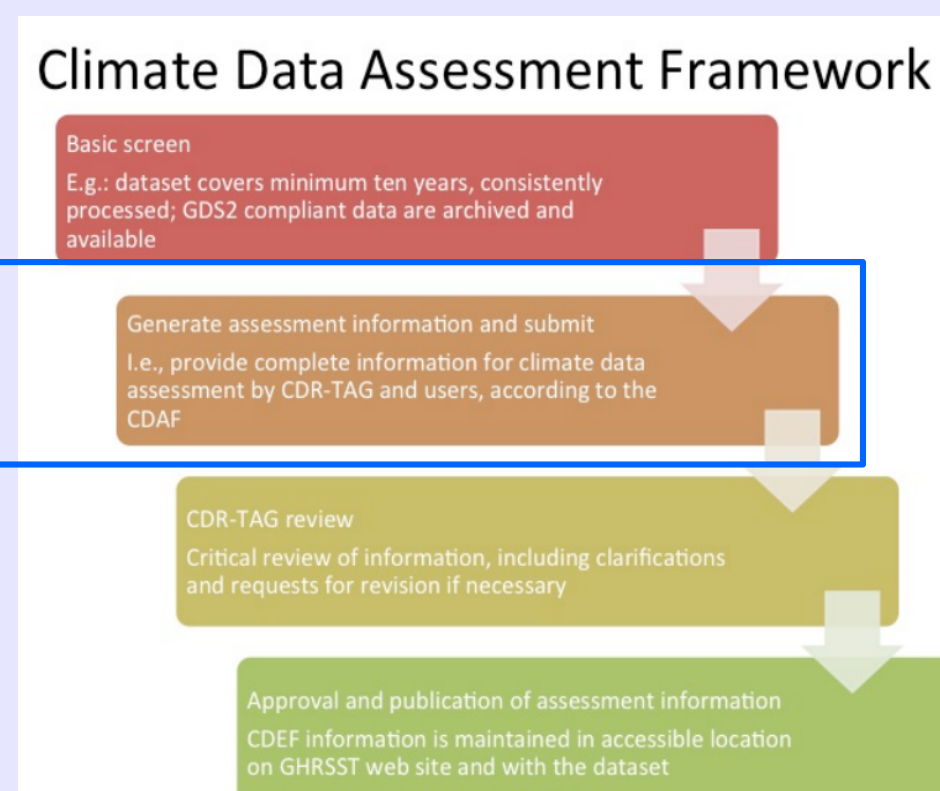
The in situ data are ingested daily into felyx, and miniprods and metrics are extracted from Sentinel-3 (and other satellite sensors) L1 and L2 files (provided by Eumetsat).

A felyx command line tool generates every day a file containing all the processed match-ups for a given type of in situ data, available on FTP to users.

Proposal for the Climate Data Record Assessment Framework

Evaluate if a dataset qualifies as a **climate data record** through an **integrated tool suite**.

- Felyx tool suite is provided to data producers on a pre-configured virtual machine (virtualbox), shipped with a reference in situ dataset selected & provided by GHRSSST
- Run on your data in GHRSSST format and produce match-ups and metrics vs in situ
- Analyse the metrics and produce CDAF diagnostics
- Publish the results on GHRSSST central web site



acknowledgements

Felyx development is funded by the **European Space Agency**.



Note : felyx is a distributed system : full installation and deployment from the source code on a cluster may require IT support.



