



## GISC Offenbach WIS2 Pilot



18.01.2022

GISC Offenbach WIS2 Pilot

---

## Background

- Task Team on CF-NetCDF
  - Developed WMO extensions to CF-NetCDF conventions
  - Test and Evaluate CF-NetCDF for Ocean and Radar data
  - Asking for easy distribution and access to available data for users
  
- WMO Members
  - Whats the TTAAii for this?
  - Can we have that on the GTS?
  - We need to extend the Manual on the GTS to include a new T1
  
- WIS2 Group
  - Working on retirement of TTAAii headers and manual routing
  - Does it make sense to add new data in new data formats to something we want to retire



---

# Background

## → Idea

- Have a gateway that is collecting the data and metadata and that is implementing WIS2 protocols
- Second Part: Provide a Webservice for data access / data reduction



---

## Project Objectives

Experiment with international real-time exchange of data

- using pub/sub protocols (and topic structure from TT-Protocols)
- distributed storage of the original data along with a central, mirrored repository
- discoverable using WIS Metadata
- exposure of this data to commercial search engines.

Experiment with Data Reduction Web Service (Original Idea GISC Beijing Pilot Project)

- Provide Access to DWD ICON Model data
- Possibility to cut area and elements



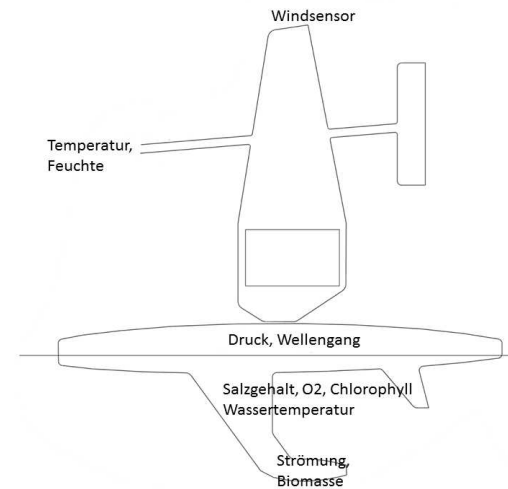
---

## Data / Metadata Standards

- Task Team on CF-NetCDF
  - Develop WMO extensions to CF-NetCDF conventions
  - Codify extensions for data profiles in WMO Data Manual
  - Increase standardization over conventional CF to support operational needs of WMO
    - Necessary to support exchange of data from multiple sources of similar data
  - CF-NetCDF profiles to act as templates for reporting data from different sources
  
- Metadata: Modified version of WCMP 1.3



# What is a saildrone



---

## Workflow

- Metadata from NOAA
  - Converted to WCMP 1.3
  - Added MQP connection information
  - Ingested into WIS
  
- DWD polling NOAA Webservice
  - New data: Pub Message on MQP Broker (AMQP + MQTT)



---

## Workflow

- Users search for netCDF data via WIS Catalogue,
- read pubSub information from metadata record, and then
- can use a subscribe client script to receive notifications about new data for this product.
- The actual data download is done via URLs sent in the messages direct from data originator



# Workflow

## WMO data distribution test, NOAA drones

Testing data distribution of Saildrone data to support WIS 2.0 pilot project  
(sd1031, sd1040, sd1048, sd1060, sd1065, sd1066)

### Synopsis

<b>PID</b>	gov.noaa.pmel.saildrone.wmo
<b>Title</b>	WMO data distribution test, NOAA drones
<b>Originator</b>	NOAA/PMEL
<b>Abstract</b>	Testing data distribution of Saildrone data to support WIS 2.0 pilot project (sd1031, sd1040, sd1048, sd1060, sd1065, sd1066)
<b>Code form</b>	( )
<b>Web URL</b>	<a href="https://data.pmel.noaa.gov/generic/erddap/tabledap/">https://data.pmel.noaa.gov/generic/erddap/tabledap/</a> <a href="https://data.pmel.noaa.gov/generic/erddap/tabledap/">https://data.pmel.noaa.gov/generic/erddap/tabledap/</a> <a href="amqps://oflkd013.dwd.de:5671">amqps://oflkd013.dwd.de:5671</a>



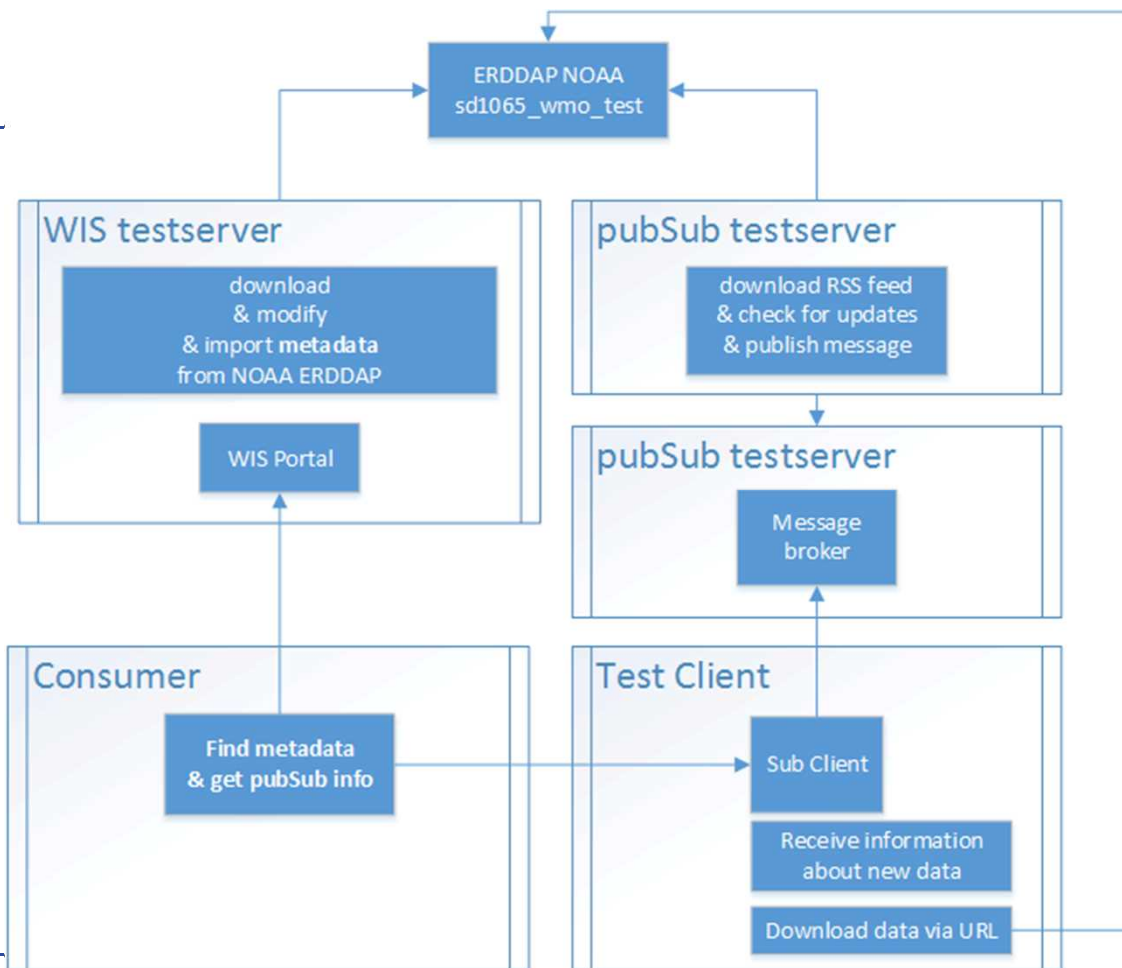
# Workflow



The screenshot shows a configuration window with a tree view. The root node is 'transferOptions', which is expanded to show 'onLine', which is further expanded to show 'linkage'. The 'linkage' node contains a 'URL' field with the value 'http://amqps://oflkd013.dwd.de:5671'. Below the 'linkage' node, there are three fields: 'protocol' (AMQPS), 'name' (exchange: netcdf\_pilot, routing\_key: v03/WIS/de/offenbach\_met\_com\_centre/observation/sea/surface/), and 'description' (WMO Information System, pub/sub messaging for new meteorological data, download products/data via link contained in the message (baseUrl+relPath). Please ask GISC Offenbach for registration to get username/password. Topic structure based on https://github.com/wmo-im/GTStoWIS2).

<b>URL</b>	http://amqps://oflkd013.dwd.de:5671
<b>protocol</b>	AMQPS
<b>name</b>	exchange: netcdf_pilot, routing_key: v03/WIS/de/offenbach_met_com_centre/observation/sea/surface/
<b>description</b>	WMO Information System, pub/sub messaging for new meteorological data, download products/data via link contained in the message (baseUrl+relPath). Please ask GISC Offenbach for registration to get username/password. Topic structure based on https://github.com/wmo-im/GTStoWIS2





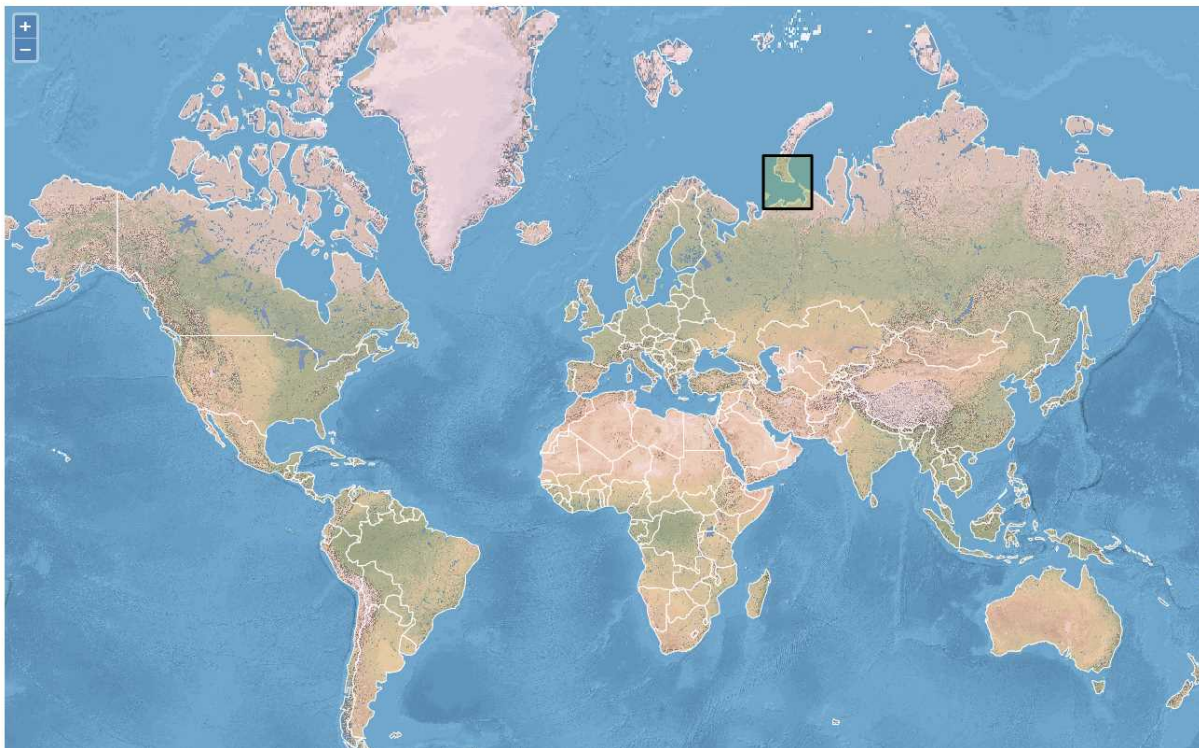
## ncdump noaa/v03/WIS/de/offenbach\_met\_com\_centre/ observation/sea/surface/sd1065\_wmo\_test.ncCF

```
netcdf sd1065_wmo_test {  
dimensions:    trajectory = 1 ;    obs = 1920 ;  
variables:     float trajectory(trajectory) ;  
                trajectory:_FillValue = NaNf ;  
                trajectory:actual_range = 1065.f, 1065.f ;  
                trajectory:cf_role = "trajectory_id" ;  
                trajectory:comment = "A trajectory is a single deployment of a drone" ;  
                trajectory:gts_ingest = "true" ;  
                trajectory:long_name = "Trajectory/Drone ID" ;  
int rowSize(trajectory) ;  
    rowSize:ioos_category = "Identifier" ;  
    rowSize:long_name = "Number of Observations for this Trajectory" ;  
    rowSize:sample_dimension = "obs" ;  
double TEMP_AIR_MEAN(obs) ;  
    TEMP_AIR_MEAN:_FillValue = NaN ;  
    TEMP_AIR_MEAN:actual_range = 23.06, 23.93 ;
```



## Part 2: Webservice

Choose Area for Model filter



### Box Coordinates:

firstPoint: 48.33687405159337,73.06471272298768  
 secondPoint: 48.33687405159337,67.89215683175325  
 thirdPoint: 62.54021244309565,67.89215683175325  
 fourthPoint: 62.54021244309565,73.06471272298768

### Request Parameters:

lon\_first

lon\_end

lat\_first

lat\_end

resolution

levelType

levels (optional)

steps (optional)

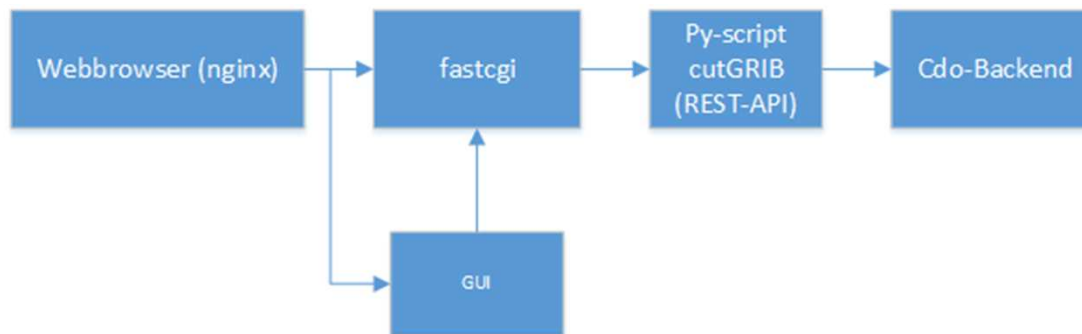
param

run

1) separator: "," | empty - all



# Architecture



GUI: <http://oflks422.dwd.de:8080/chooseArea2.html>

GRIB data from: <https://opendata.dwd.de>



# REST-API

Documentation (Swagger / OpenAPI):

[http://oflks422.dwd.de:8080/cdo\\_swagger.html](http://oflks422.dwd.de:8080/cdo_swagger.html)

## ICON CDO <sup>1.0.0</sup>

[ Base URL: oflks422.dwd.de:8080/fastcgi ]

Webservice to get single elements of DWD's ICON model on a regular lat/lon grid.

Schemes

HTTP

### /cutGRIB Retrieve selected GRIB fields

GET /cutGRIB Retrieve selected GRIB fields

#### Parameters

Name	Description
<b>lat_first</b> * required number (query)	Latitude of first grid point <input type="text" value="lat_first - Latitude of first grid point"/>
<b>lat_end</b> * required number (query)	Latitude of last grid point <input type="text" value="lat_end - Latitude of last grid point"/>
<b>lon_first</b> * required number (query)	Longitude of first grid point <input type="text" value="lon_first - Longitude of first grid point"/>
<b>lon_end</b> * required number (query)	Longitude of last grid point <input type="text" value="lon_end - Longitude of last grid point"/>



---

## REST-API

URL (Example):

<http://oflks422.dwd.de:8080/fastcgi/cutGRIB?>

[lon\\_first=9.823975720789116&lon\\_end=15.286798179059222&](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?lon_first=9.823975720789116&lon_end=15.286798179059222&)

[lat\\_first=48.48902527268359&lat\\_end=51.81260638543509&](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?lat_first=48.48902527268359&lat_end=51.81260638543509&)

[resolution=0.125&](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?resolution=0.125&)

[param=p&](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?param=p&)

[levelType=model-level&levels=1](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?levelType=model-level&levels=1)

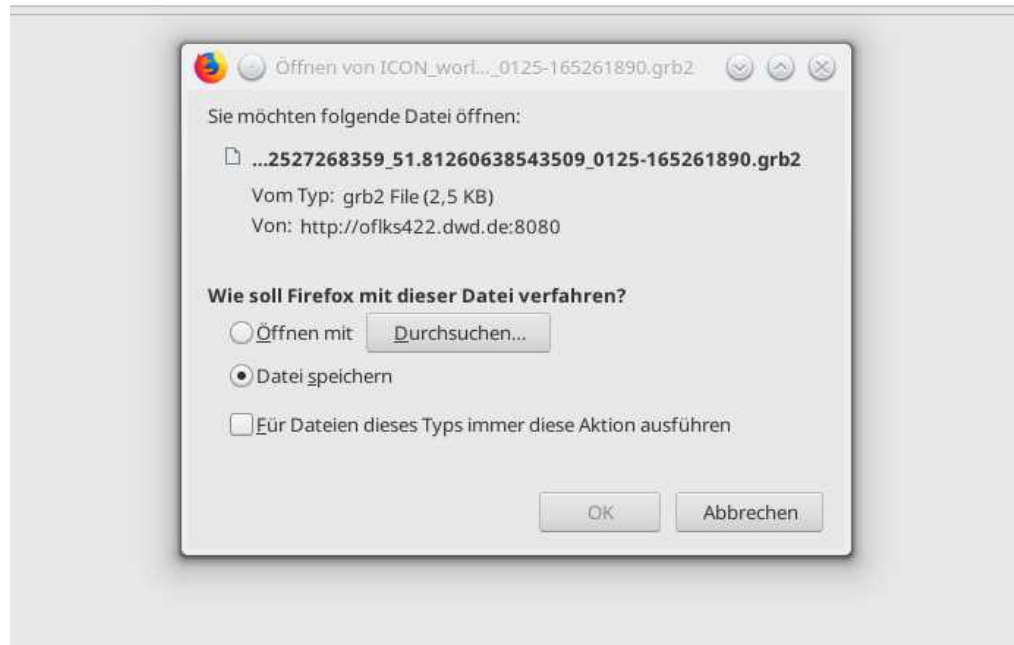
[run=00&steps=000](http://oflks422.dwd.de:8080/fastcgi/cutGRIB?run=00&steps=000)





# REST-API

oflks422.dwd.de:8080/fastcgi/cutGRIB?lon\_first=9.823975720789116&lon\_end=15.286798179059222&lat\_firs



---

## Software

- Nginx (fastcgi)
- Python
- cdo
- eccodes
- jQuery
- openlayers

---

ご清聴ありがとうございました

質問？

