PORT OF SPAIN WIS2 TRAINING WORKSHOP · June 2026

Data conversion and data ingestion in wis2box

Maaike Limper
WMO Secretariat · WIS Section · ESDP Department

WORLD METEOROLOGICAL ORGANIZATION

wis2box workflow

wis2box workflow relies on Datasets to publish WIS2 notifications:

- A Dataset is described by **Discovery Metadata** published as WCMP2-record
- A Dataset connects data and metadata using the **metadata identifier**
- A Dataset defines the **topic hierarchy** to publish the WIS2 data notification
- *wis2box: A Dataset contains **data plugins** used to **transform** and **publish** the data*

WORLD METEOROLOGICAL ORGANIZATION

Datasets are stored in the pygeoapi 'discovery-metadata' collection
see: <wis2box-url>/oapi/collections/discovery-metadata/items?f=json

How to configure datasets in wis2box

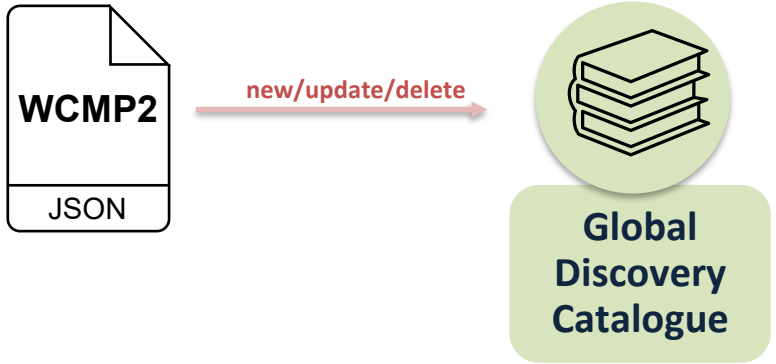
Two ways to configure a new dataset in the wis2box:

- Use the **dataset-editor** in the wis2box-webapp
- … or share an MCF file with the wis2box-management container and execute '`wis2box dataset publish <file-path>`'

new WCMP2 notification on origin/a/wis2/<centre-id>/metadata
for every new dataset published

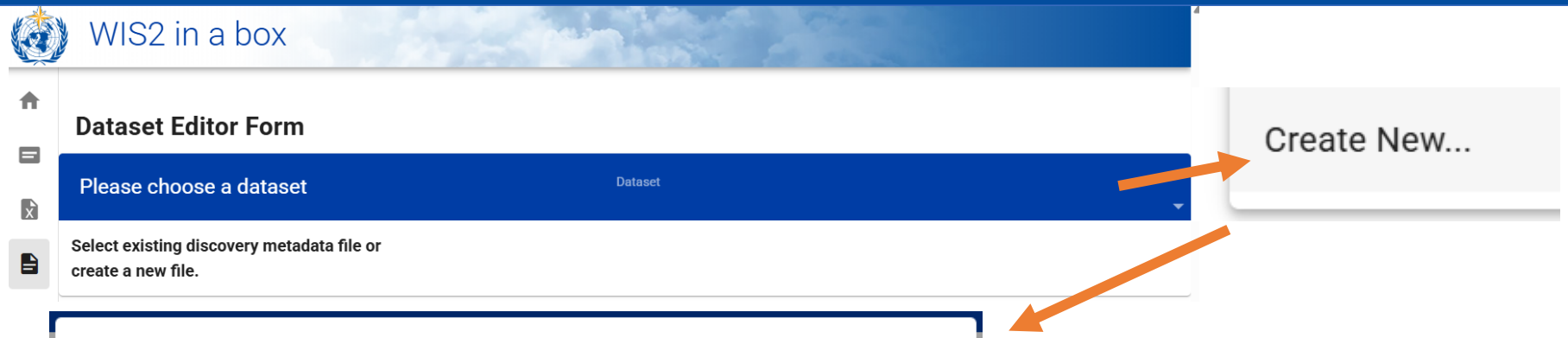
update WCMP2 notification on origin/a/wis2/<centre-id>/metadata
for every updated dataset published

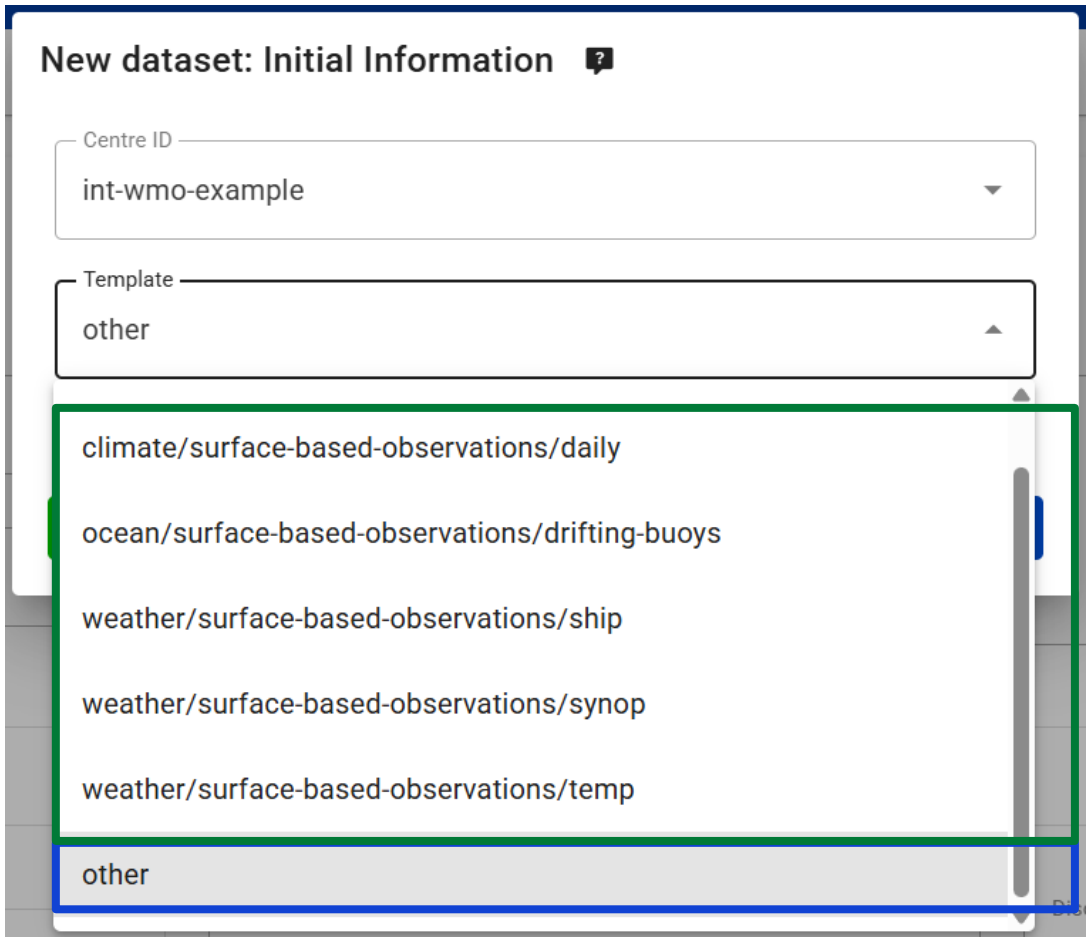
delete WCMP2 notification on origin/a/wis2/<centre-id>/metadata
whenever a dataset is unpublished



WORLD METEOROLOGICAL ORGANIZATION

How to configure datasets in wis2box: using the dataset editor





Centre ID:
Select from dropdown or type any valid string
defines the metadata-id and centre-id in topic

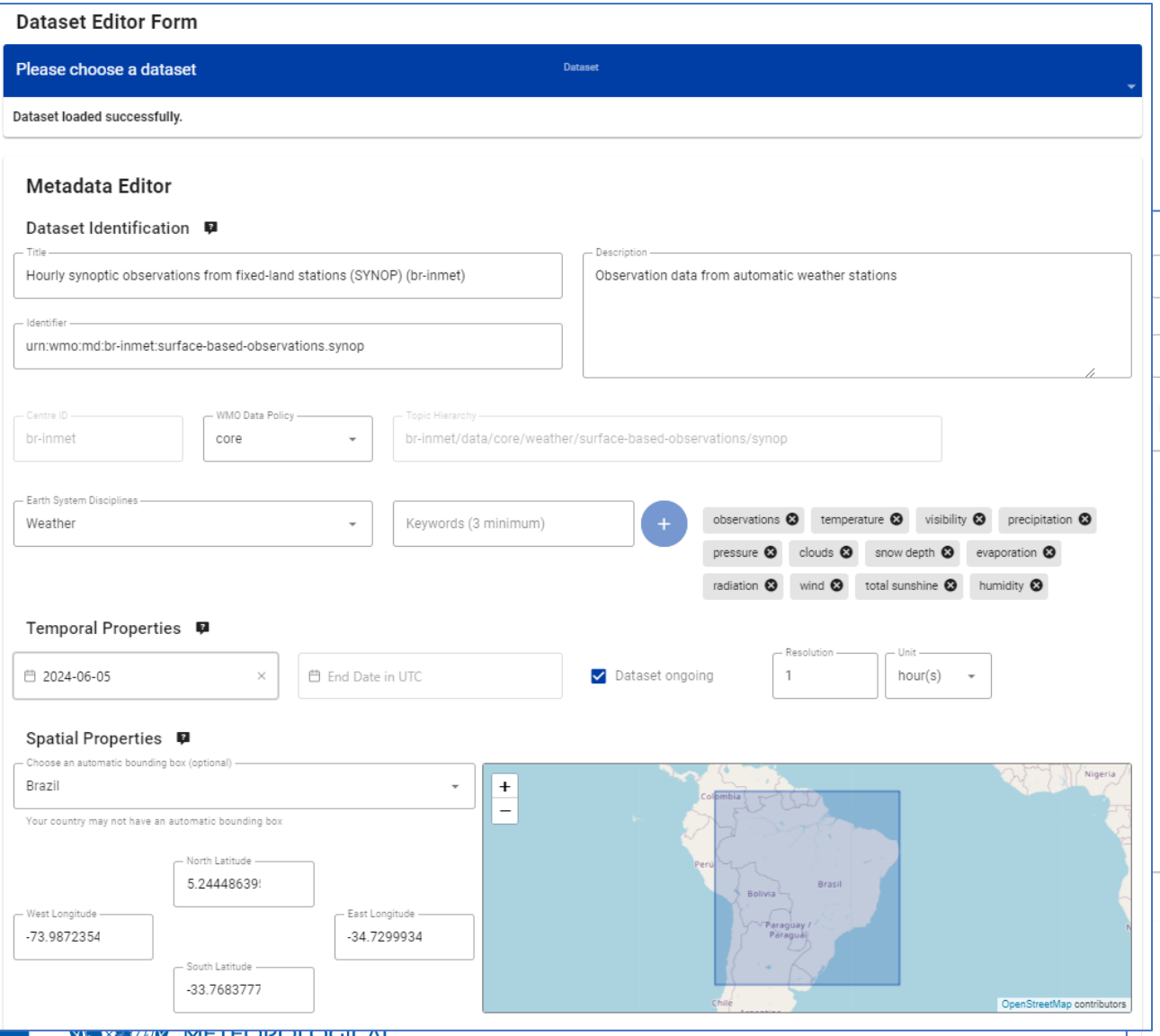
Template:
Select option from dropdown

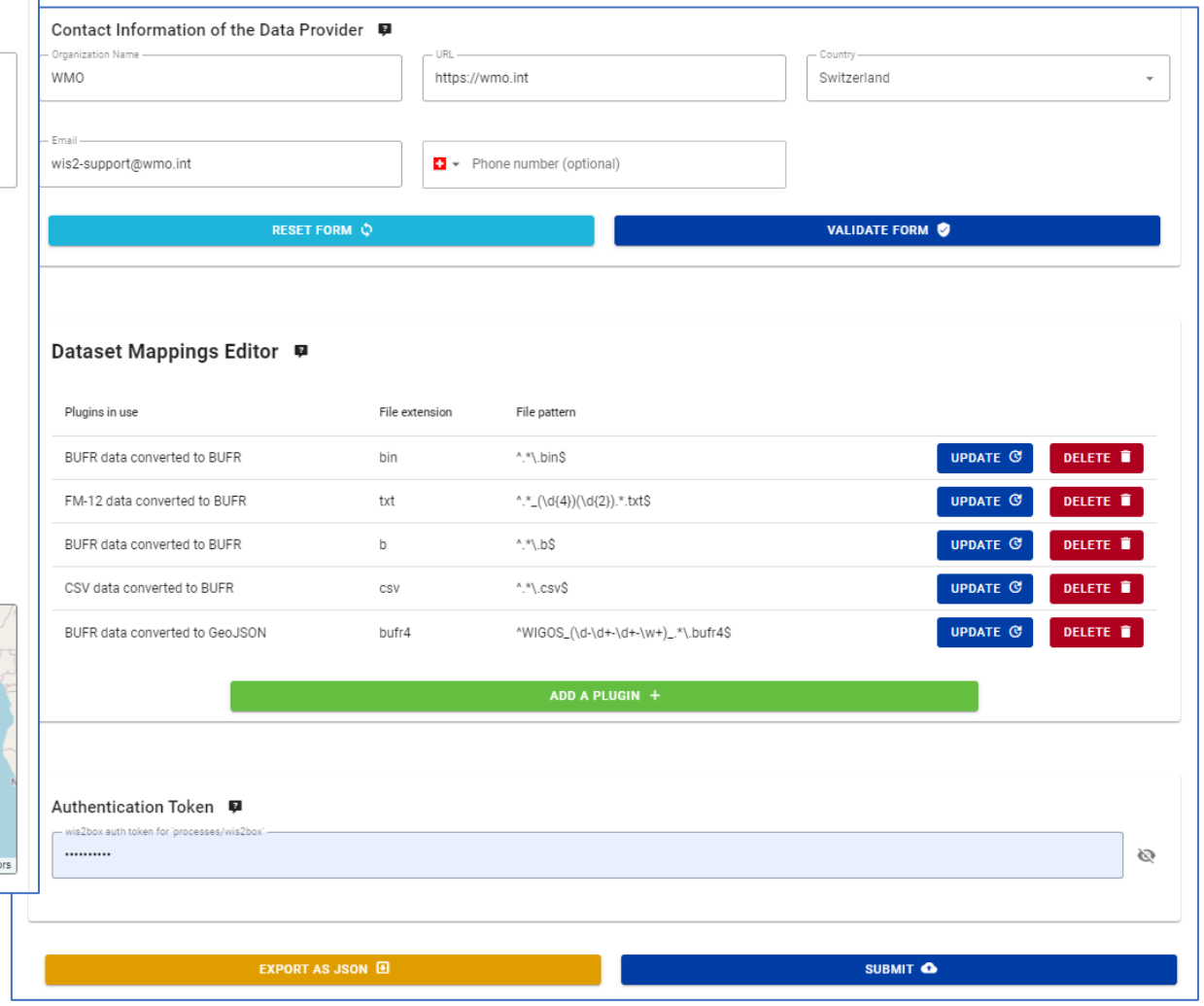
Select **template** to initialize the dataset with a fixed topic and pre-defined values for keywords and data mappings

Select '**other**' for any types of data you wish to publish, and you will select a topic based on the WIS2 Topic Hierarchy

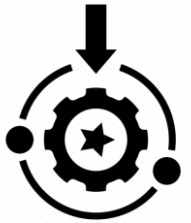
How to configure datasets in wis2box: using the dataset editor

Step 1. Define metadata and validate form
Step 2. Define data plugins
Step 3. Provide Authentication Token (created by wis2box-admin)
Step 4. Submit the dataset for publication





Datasets in wis2box: data plugins



Data Mappings map a specific **dataset** to a set of **data plugins**

Data plugins use an abstract model/approach to enable extensibility and reuse

A **data plugin** defines the actions taken to **transform** and **publish** the data

See github.com/wmo-im/wis2box/tree/main/wis2box-management/wis2box/data

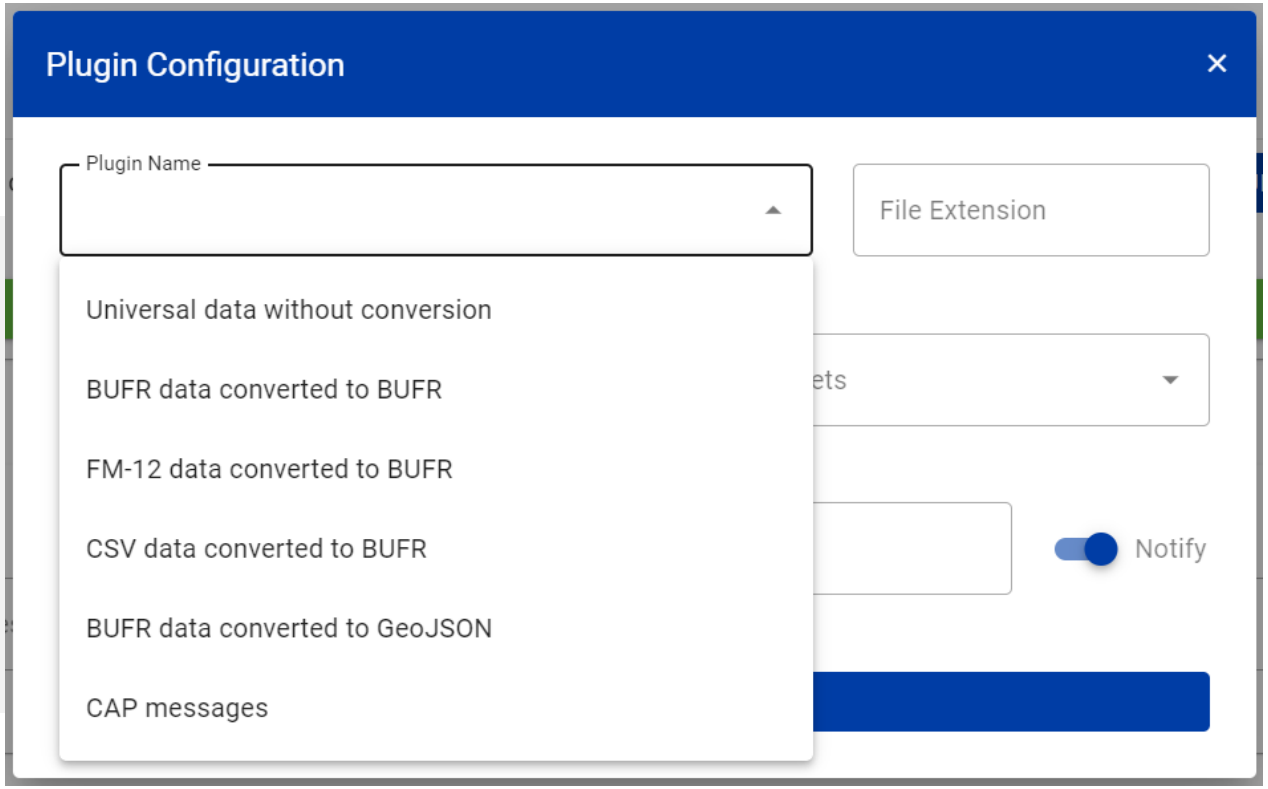
NOTE:
Preconfigured data-plugins in data mappings for template="weather/surface-based-observations" in dataset-editor, can be adjusted as required

Plugins in use	File extension	File pattern
BUFR data converted to BUFR	bin	^.*\.bin$
FM-12 data converted to BUFR	txt	^.*_(\d{4})(\d{2}).*.txt$
BUFR data converted to BUFR	b	^.*\.b$
CSV data converted to BUFR	csv	^.*\.csv$
BUFR data converted to GeoJSON	bufr4	^WIGOS_(\d-\d+-\d+-\w+)_.*\.bufr4$

wis2box data plugins

wis2box contains the following built-in data plugins:

- wis2box.data.universal.UniversalData
- wis2box.data.cap_message.CAPMessageData
- wis2box.data.bufr4.ObservationDataBUFR
- wis2box.data.synop2bufr.ObservationDataSYNOP2BUFR
- wis2box.data.csv2bufr.ObservationDataCSV2BUFR
- wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON



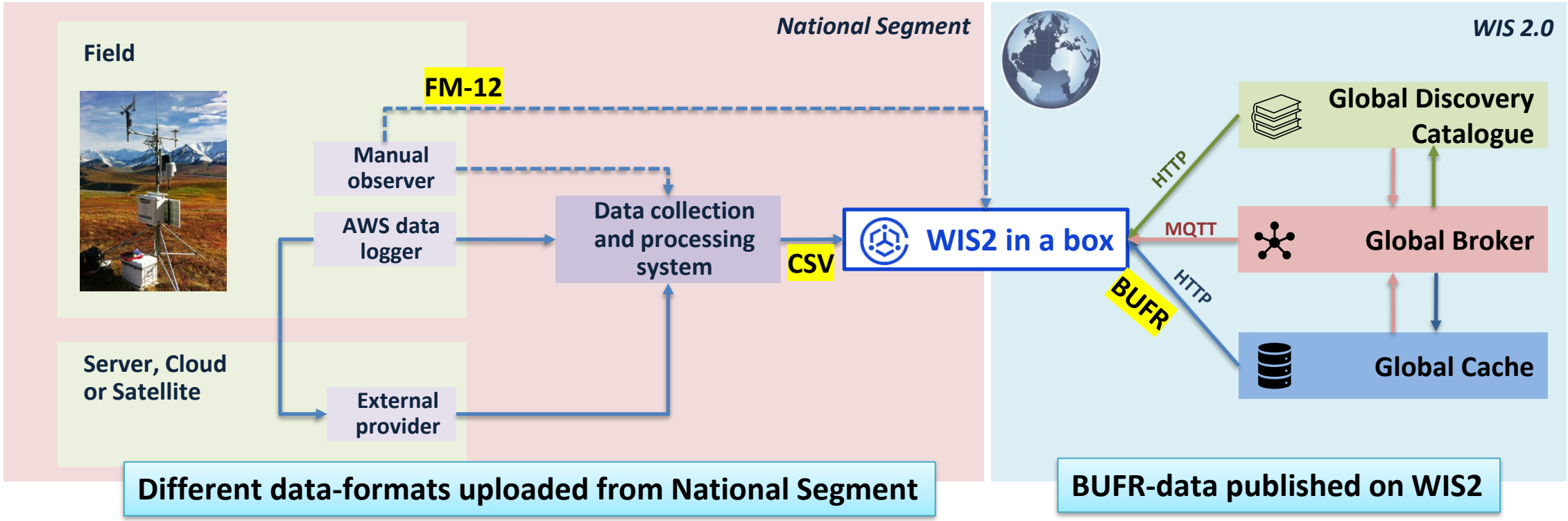
Developers are encouraged to contribute new data plugins to wis2box!

WORLD METEOROLOGICAL ORGANIZATION

BUFR tools in wis2box

wis2box data plugins use the BUFR tools to enable data transformation prior to publication

- **csv2bufr** enables any system to prepare a data-extract without needing local BUFR conversion tools
- **synop2bufr** enables direct publication of FM-12 synop reports from manual observers
- **bufr2bufr** splits bulletins and ensure data has a valid WIGOS-station-identifier



WORLD METEOROLOGICAL ORGANIZATION

BUFR ?

BUFR (Binary Universal Form for the Representation of meteorological data)

- a flexible binary format
- mainly used to encode in situ and satellite observations

How BUFR Works:

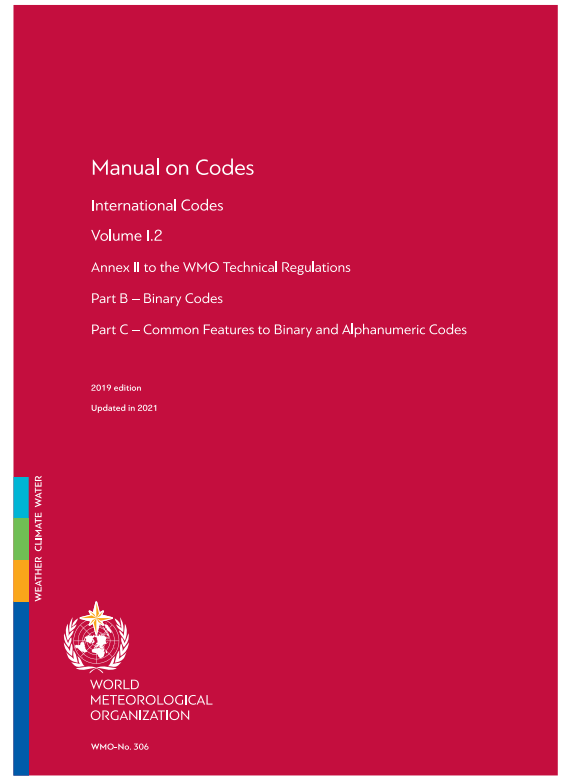
- **BUFR message** consists of a **header and a body containing data and metadata.**
- **BUFR descriptors** define how data is structured and encoded within a BUFR message
- **Compression for efficient storage** to reduce transmission times.

BUFR Table System

BUFR descriptors are stored in **BUFR Tables** (Tables A, B, C, and D).

- **Table B** contains all **element descriptors** with their definitions.
- **Table D** defines **combinations of descriptors** into predefined **sequences**

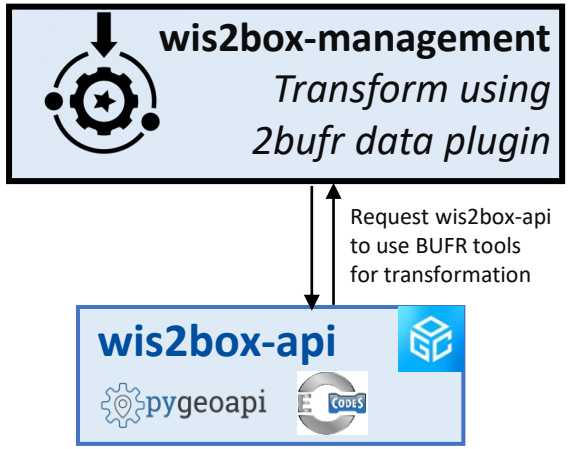
BUFR tables are part of the **WMO Manual on Codes**:
https://library.wmo.int/idurl/4/35625 new version is released every six months



WORLD METEOROLOGICAL ORGANIZATION

BUFR tools in wis2box

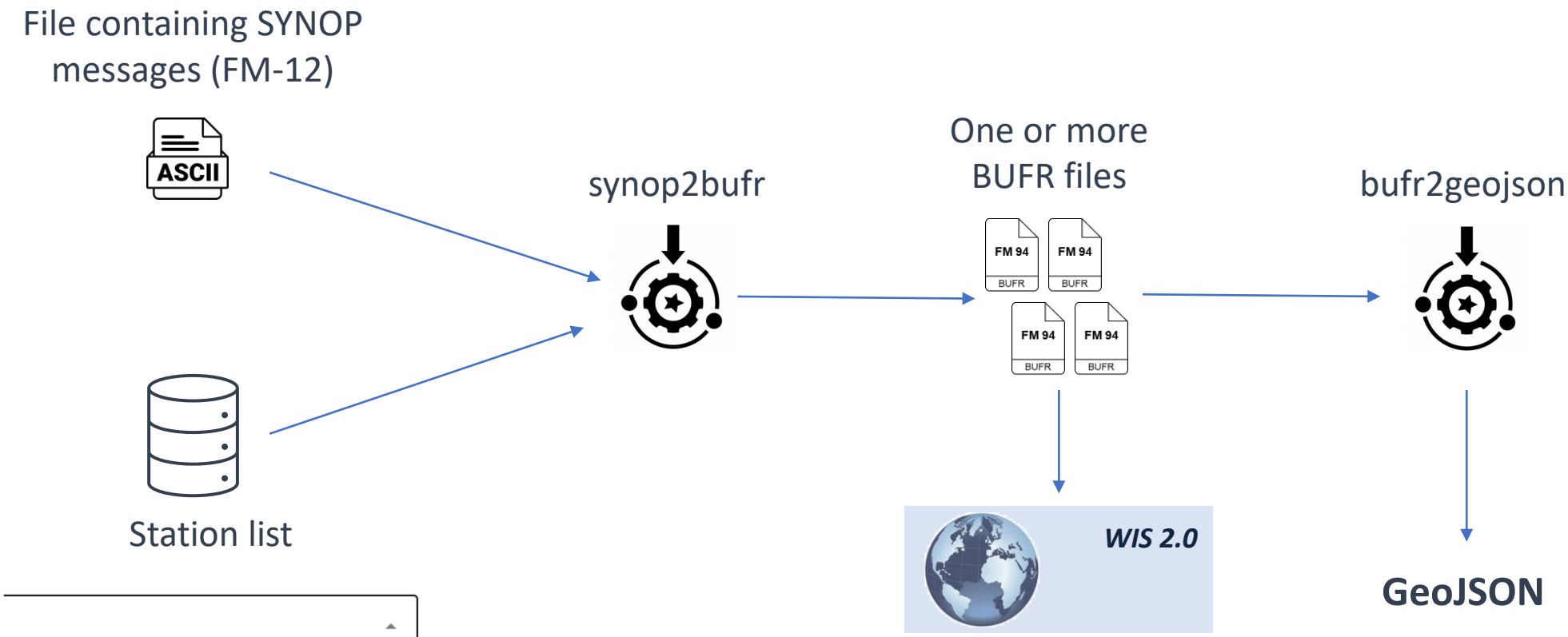
‘wis2box-api’ container is based on a Docker image containing ecCodes software library
… the container also includes several BUFR-conversion tools that work with ecCodes

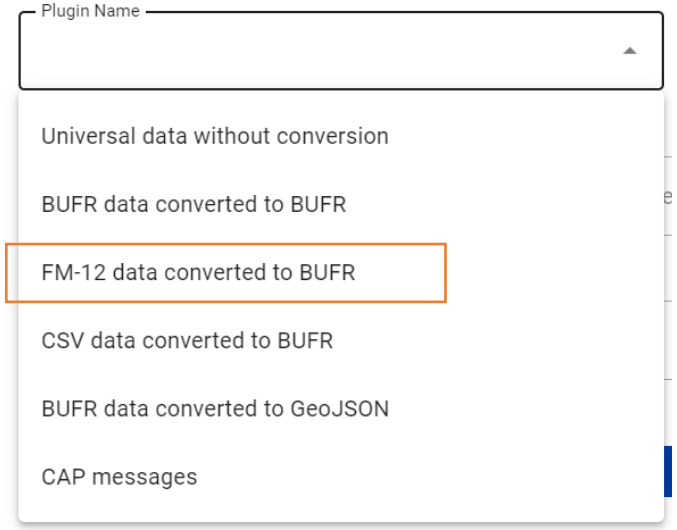


BUFR tools used within wis2box-api available to be used standalone:
https://github.com/World-Meteorological-Organization/csv2bufr
https://github.com/World-Meteorological-Organization/synop2bufr
https://github.com/World-Meteorological-Organization/bufr2geojson

WORLD METEOROLOGICAL ORGANIZATION

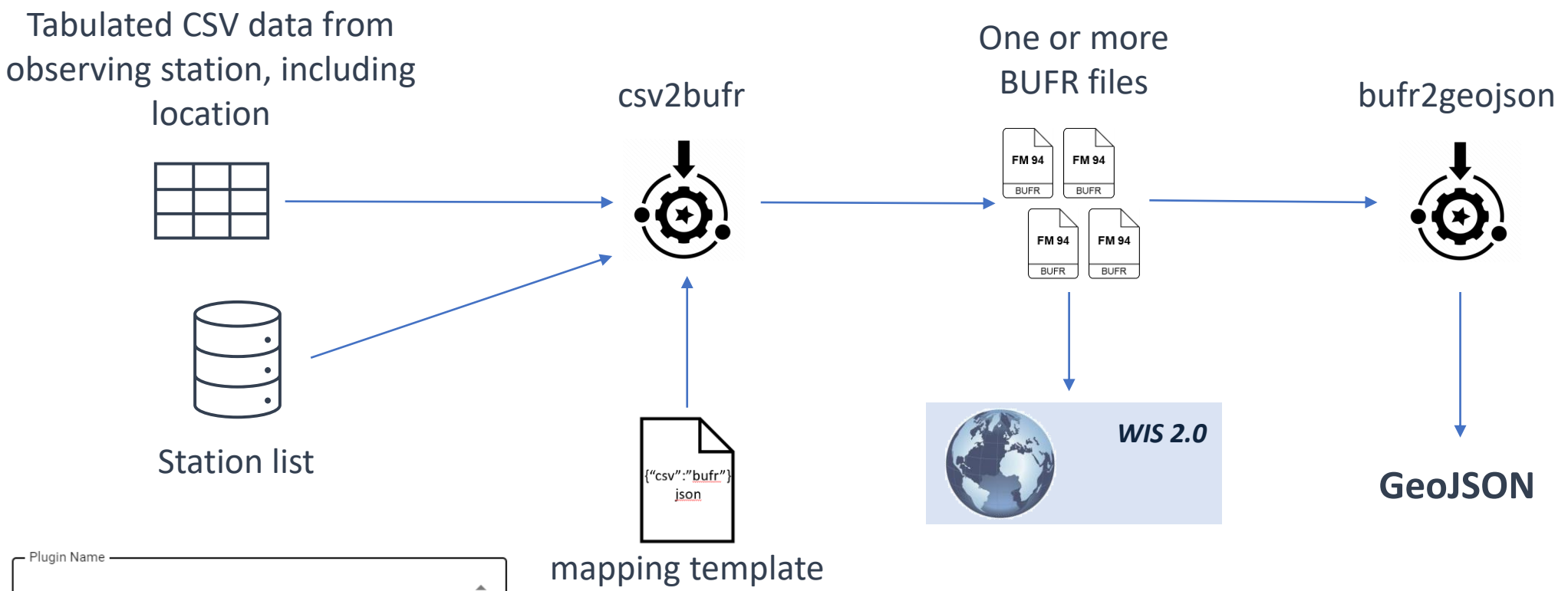
wis2box data plugins: synop2bufr

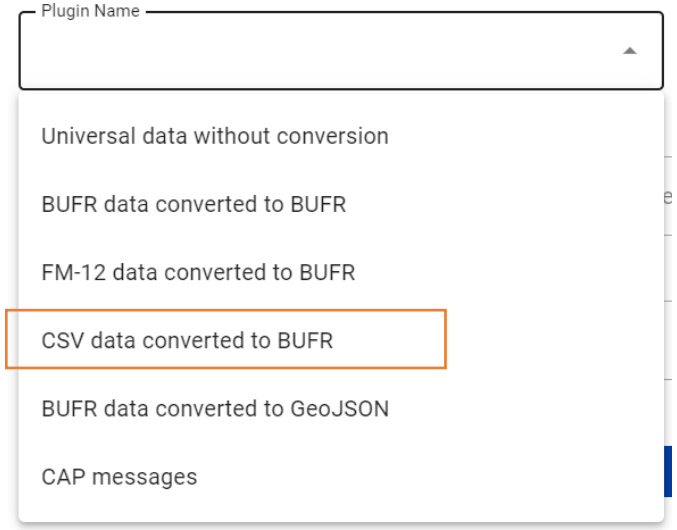





```
plugins:
    txt:
      - plugin: wis2box.data.synop2bufr.ObservationDataSYNOP2BUFR
        notify: true
        file-pattern: '^*_(\d{4})(\d{2}).*\.txt$'
    bufr4:
      - plugin: wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
        file-pattern: ' ^WIGOS_(\d-\d+-\d+-\w+)_.*\.bufr4$'
```

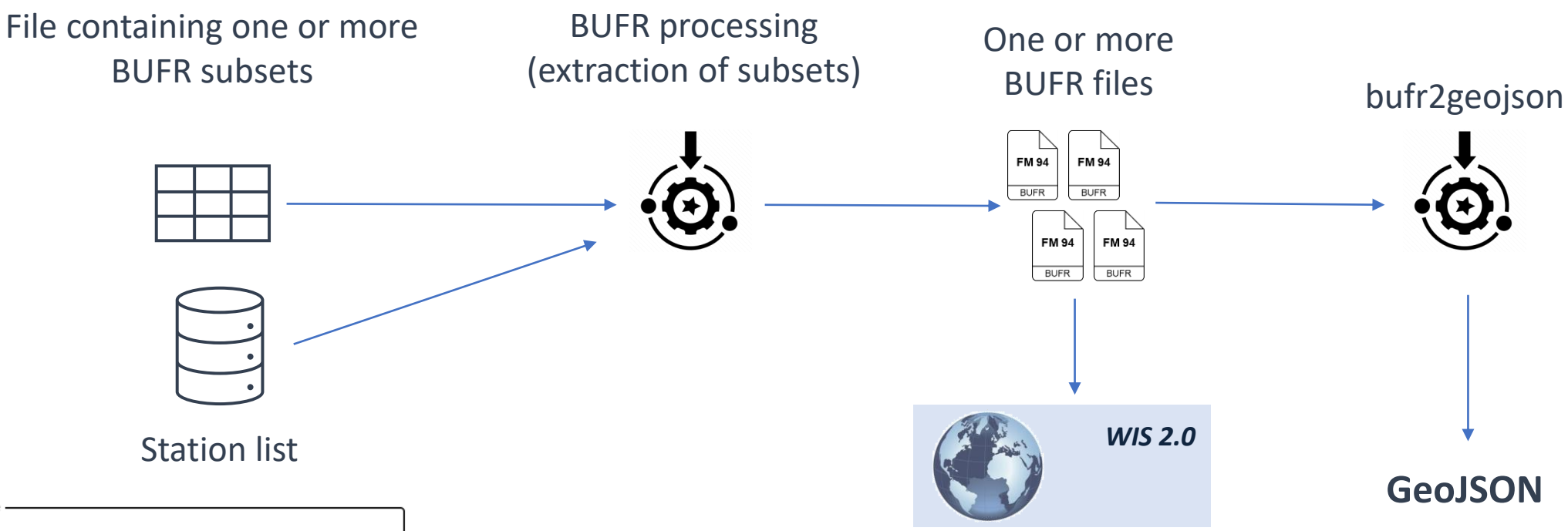
wis2box data plugins: csv2bufr

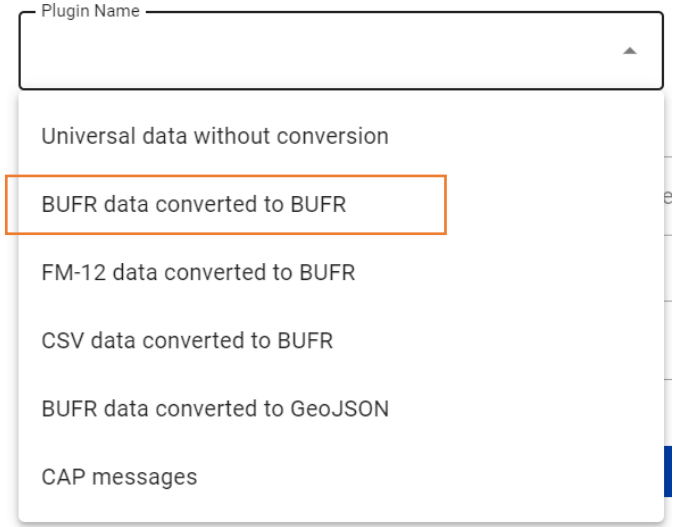





```
plugins:
    csv:
      - plugin: wis2box.data.csv2bufr.ObservationDataCSV2BUFR
        template: aws-template.json
        notify: true
        file-pattern: '^.*\.csv$'
    bufr4:
      - plugin: wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
        file-pattern: ' ^WIGOS_(\d-\d+-\d+-\w+)_.*\.bufr4$'
```

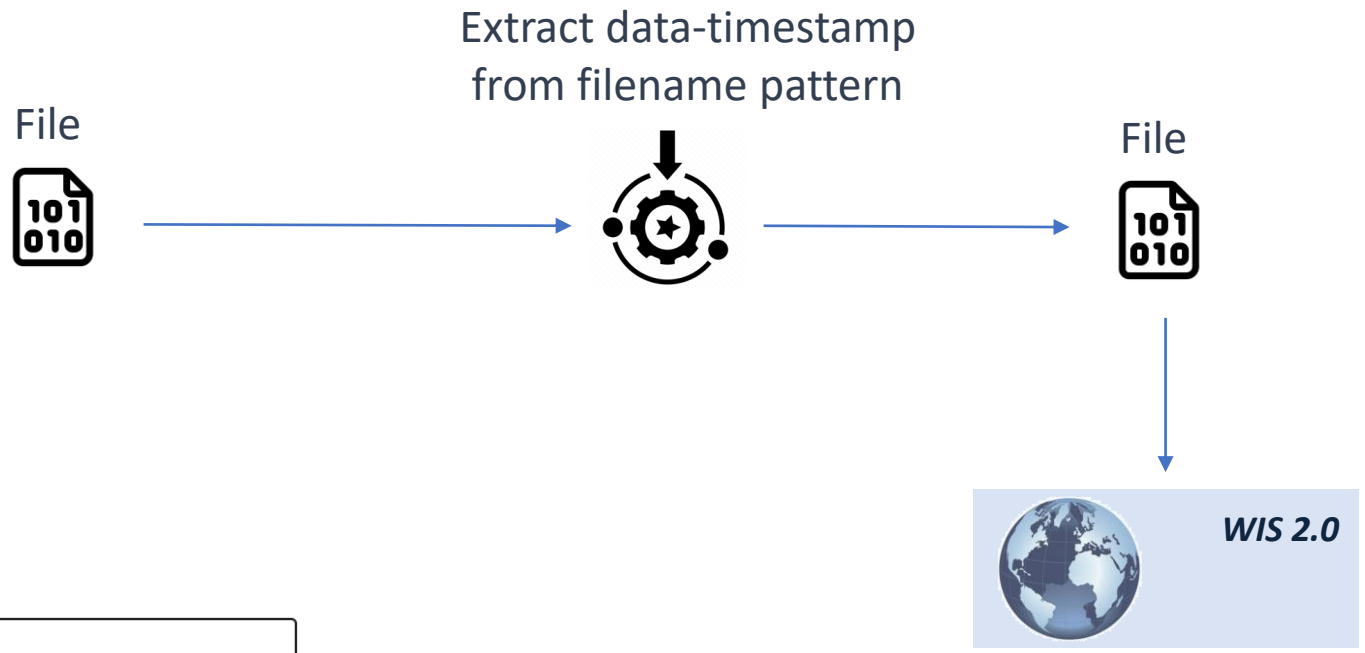
wis2box data plugins: bufr2bufr

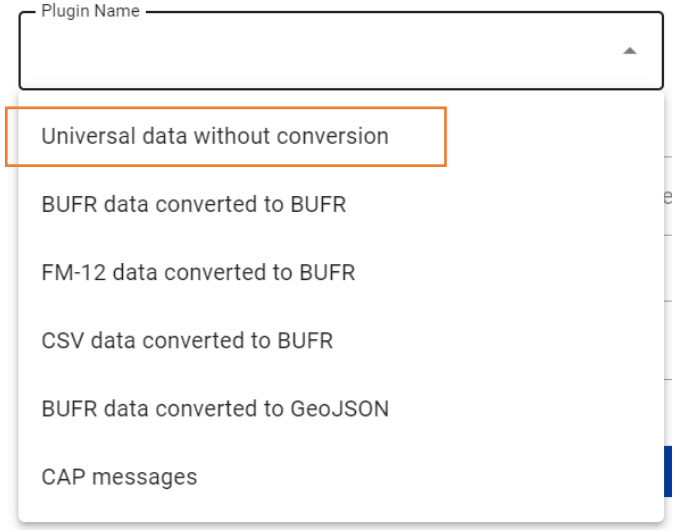





```
plugins:
    bin:
      - plugin: wis2box.data.bufr4.ObservationDataBUFR
        notify: true
        file-pattern: '^.*\.bin$'
    bufr4:
      - plugin: wis2box.data.bufr2geojson.ObservationDataBUFR2GeoJSON
        file-pattern: ' ^WIGOS_(\d-\d+-\d+-\w+)_.*\.bufr4$'
```

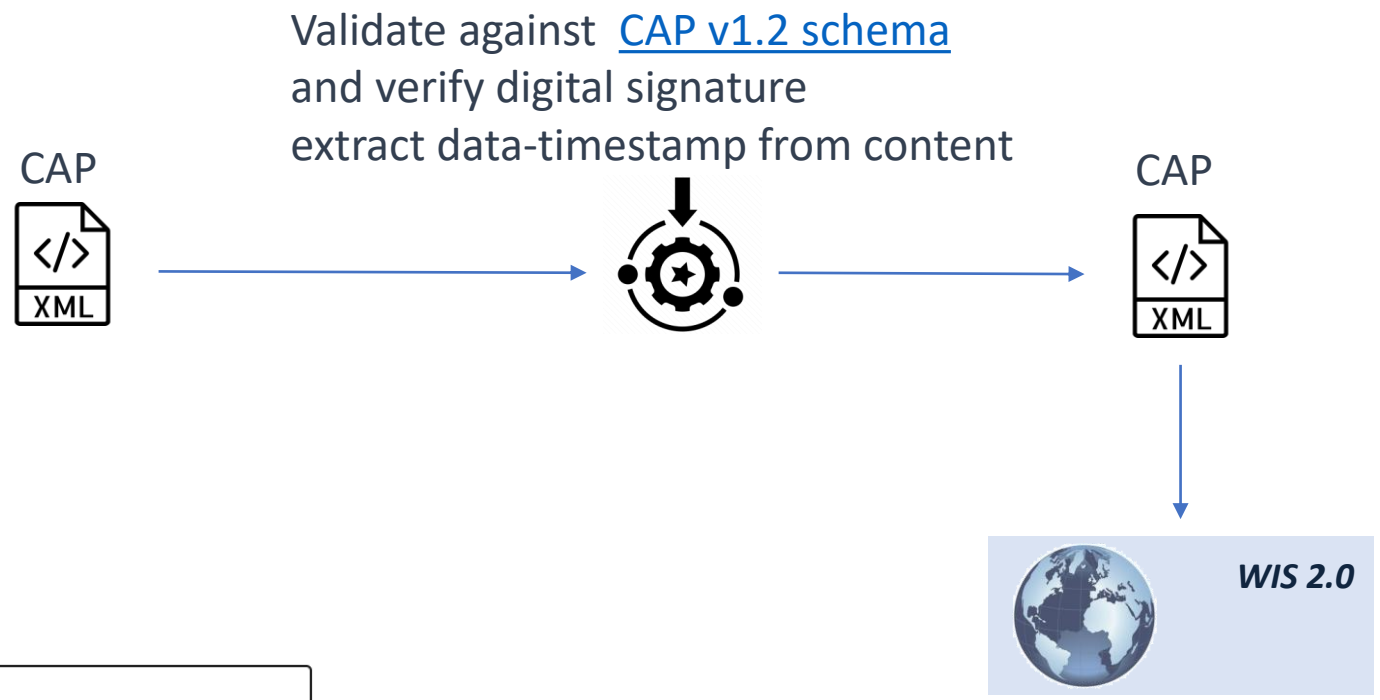
wis2box data plugins: universal/passthrough

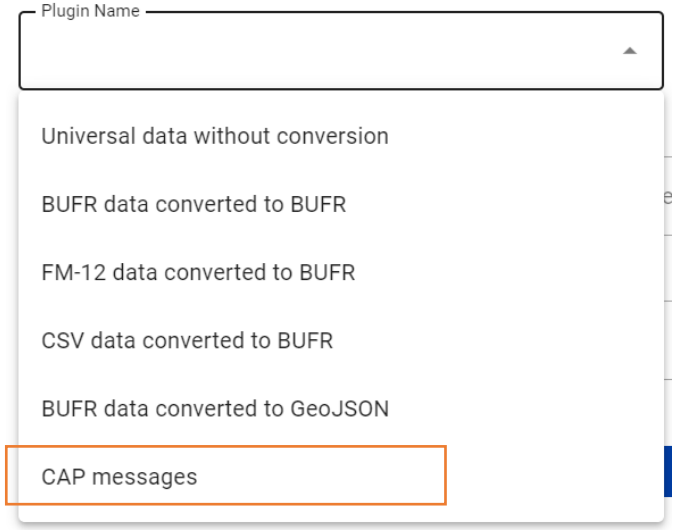





```
plugins:
    grib2:
      - plugin: wis2box.data.universal.UniversalData
        notify: true
        buckets:
            - ${WIS2BOX_STORAGE_INCOMING}
        file-pattern: '^.*_(\d{8})\d{2}.*\.grib2$'
```

wis2box data plugins: Common Alerting Protocol (CAP)





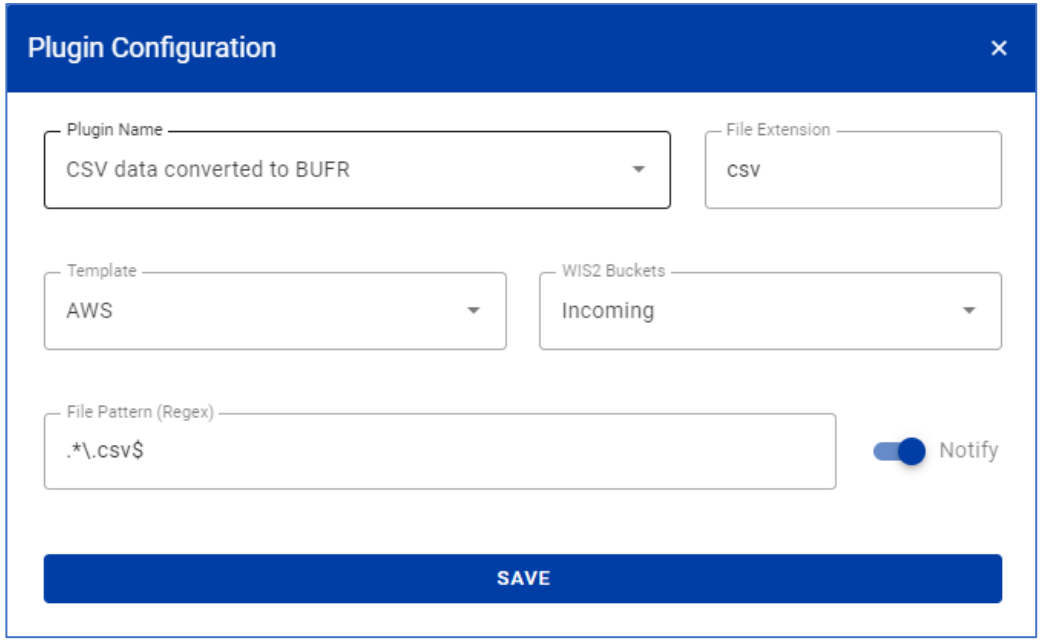

```
plugins:
    xml:
      - plugin: wis2box.data.cap_message.CAPMessageData
        notify: true
        buckets:
            - ${WIS2BOX_STORAGE_INCOMING}
        file-pattern: '^.*\.xml$'
```

CSV-to-BUFR introduction

wis2box contains a built-in data-plugin for "CSV data converted to BUFR"

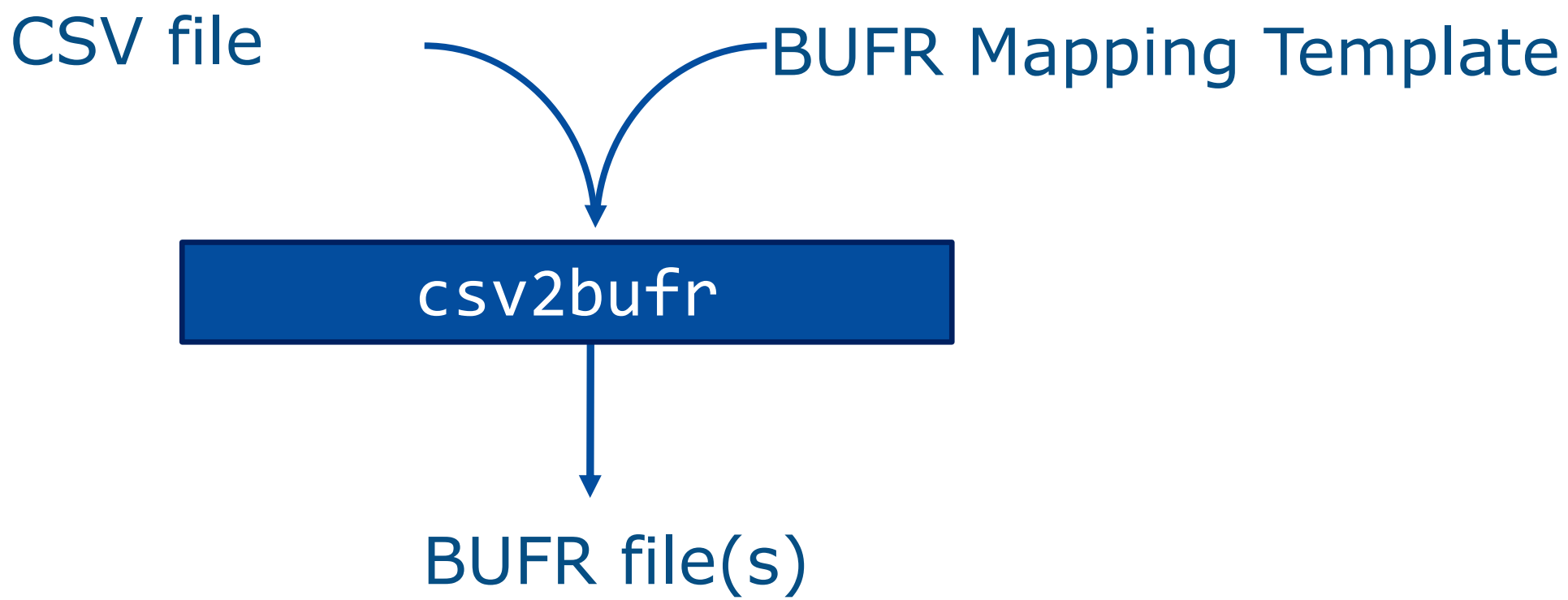
- Enables ingesting data in CSV format while publishing WIS2 notifications containing BUFR data
- Uses a mapping template to map input columns to BUFR codes

csv2bufr-transformation is provided by the 'csv2bufr' module that can also be used outside of the wis2box
https://github.com/wmo-im/csv2bufr



WORLD METEOROLOGICAL ORGANIZATION

CSV-to-BUFR key ingredients



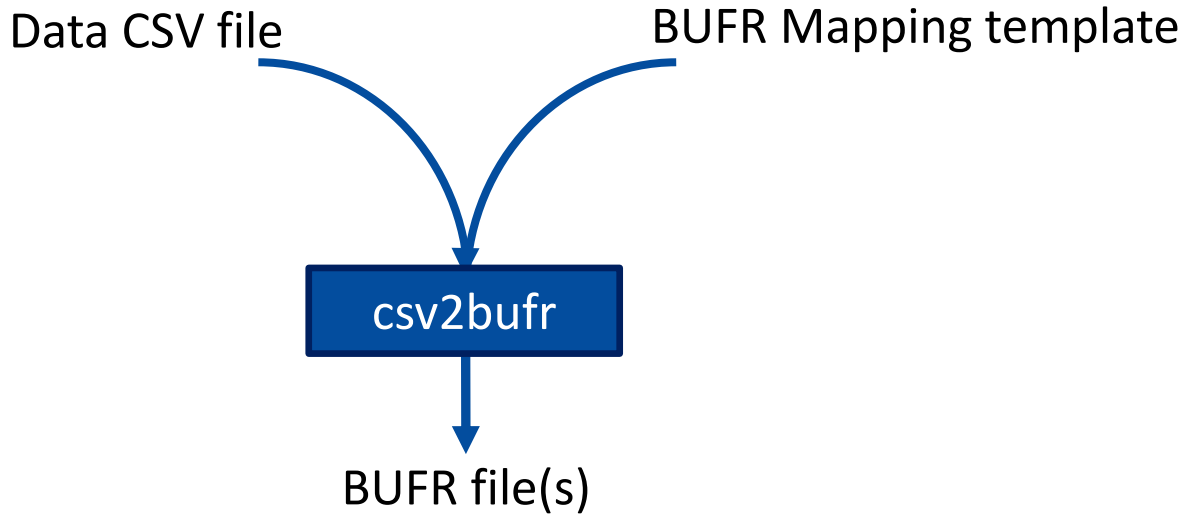
WORLD METEOROLOGICAL ORGANIZATION

BUFR mapping template

Template: JSON file defining mapping between columns in the input CSV data to codes encoded in the output BUFR data

... using the human-readable eccodes_key rather than the 6 digit BUFR FXXYYY code


```
"data": [
 ....
 {"eccodes_key": "#1#nonCoordinatePressure", "value": "data:station_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},
 {"eccodes_key": "#1#pressureReducedToMeanSeaLevel", "value": "data:msl_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},
 {"eccodes_key": "#1#nonCoordinateGeopotentialHeight", "value": "data:geopotential_height", "valid_min": "const:-1000", "valid_max": "const:130071"},
  {"eccodes_key": "#1#heightOfSensorAboveLocalGroundOrDeckOfMarinePlatform", "value": "data:thermometer_height", "valid_min": "const:0", "valid_max": "const:655.35"},
  {"eccodes_key": "#1#airTemperature", "value": "data:air_temperature", "valid_min": "const:193.15", "valid_max": "const:333.15"},
   ... ]
```

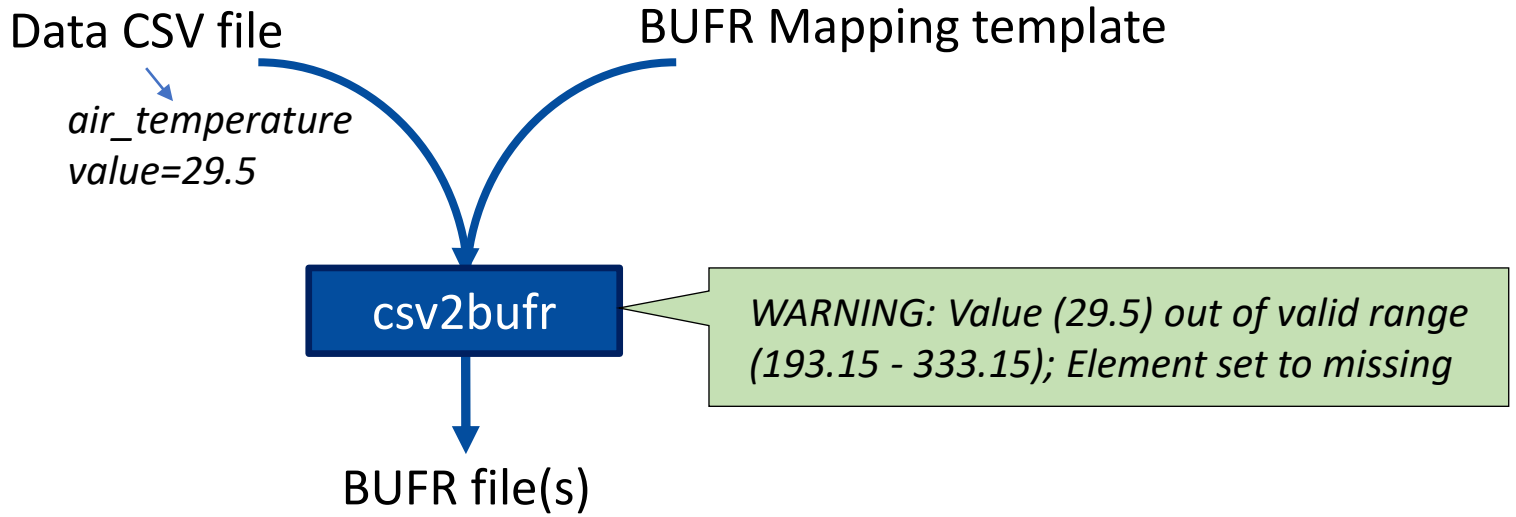
WORLD METEOROLOGICAL ORGANIZATION

BUFR mapping template

valid_min: Minimum tolerable value
valid_max: Maximum tolerable value

csv2bufr uses 'valid_min' and 'valid_max' to catch input errors: invalid values will not be encoded in the BUFR


```
"data": [
 ....
 {"eccodes_key": "#1#nonCoordinatePressure", "value": "data:station_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},
 {"eccodes_key": "#1#pressureReducedToMeanSeaLevel", "value": "data:msl_pressure", "valid_min": "const:50000", "valid_max": "const:150000"},
 {"eccodes_key": "#1#nonCoordinateGeopotentialHeight", "value": "data:geopotential_height", "valid_min": "const:-1000", "valid_max": "const:130071"},
  {"eccodes_key": "#1#heightOfSensorAboveLocalGroundOrDeckOfMarinePlatform", "value": "data:thermometer_height", "valid_min": "const:0", "valid_max": "const:655.35"},
  {"eccodes_key": "#1#airTemperature", "value": "data:air_temperature", "valid_min": "const:193.15", "valid_max": "const:333.15"},
   ... ]
```

WORLD METEOROLOGICAL ORGANIZATION

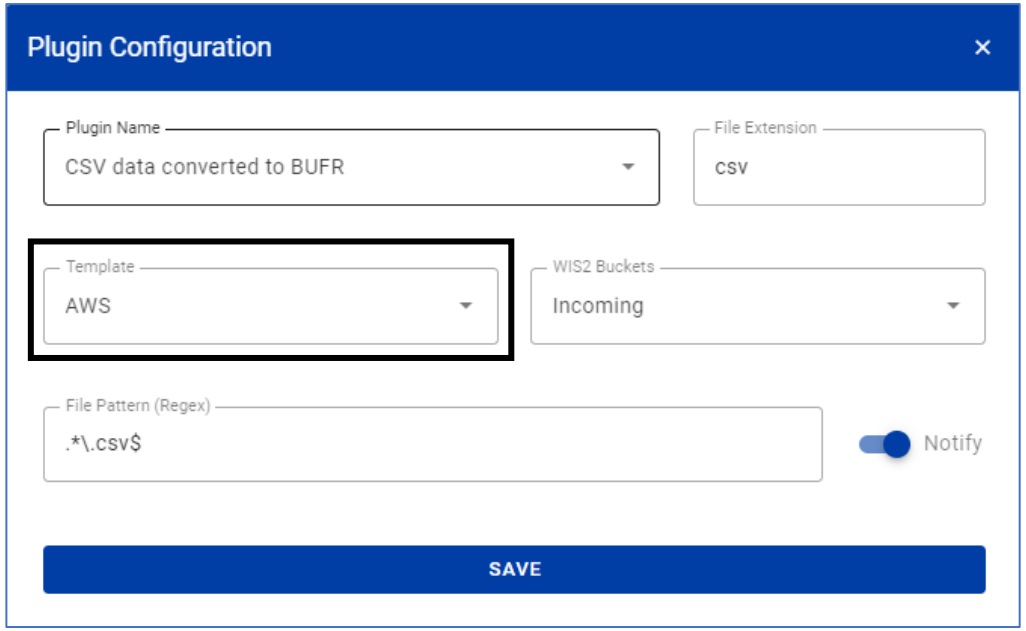
The AWS template

AWS Template:
Standardized CSV format to ingest data from Automatic Weather Stations in support of GBON reporting requirements

The format is intended for use with automatic weather stations reporting a minimum number of parameters, including pressure, air temperature and humidity, wind speed and direction and precipitation

Based on SI units, with no unit prefixes, e.g. Pa rather than hPa, Kelvin rather than degrees Celsius etc

Template = AWS in the Plugin Configuration refers to the 'aws-template.json' as defined in https://github.com/wmo-im/csv2bufr-templates



WORLD METEOROLOGICAL ORGANIZATION

The DAYCLI template

DAYCLI Template:
Standardized CSV format for converting daily climate data to BUFR sequence 307075

The format is intended for use with Climate Data Management Systems to publish data on WIS2

Requirement for daily climate observations to monitor extremes

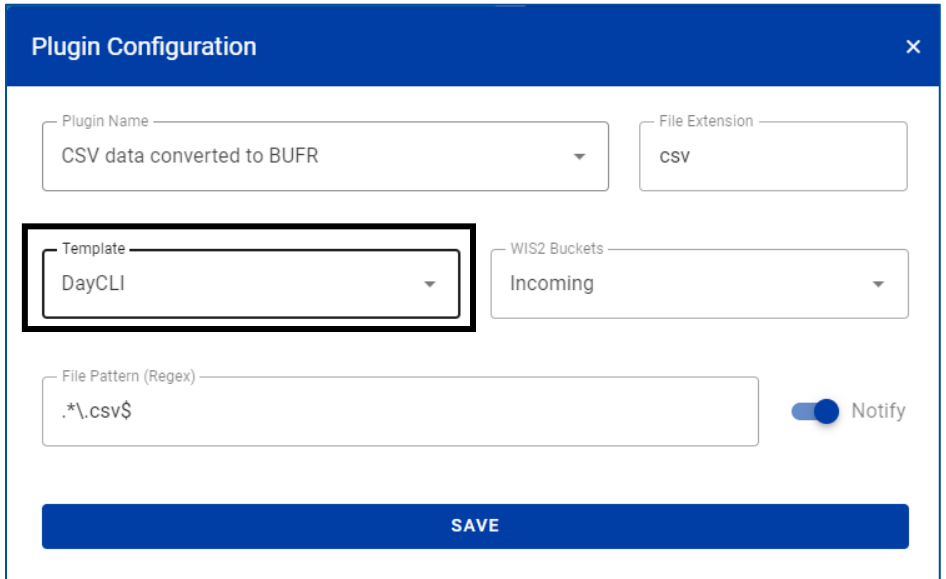
Daily observations of:

- Minimum, maximum and average temperature over 24 hours period
- Total accumulated precipitation over 24 hours period
- Total snow depth at time of observation
- Depth of fresh snow over 24 hours period

Recognition of different reporting practices by Members, explicit reporting of 24 hours period used

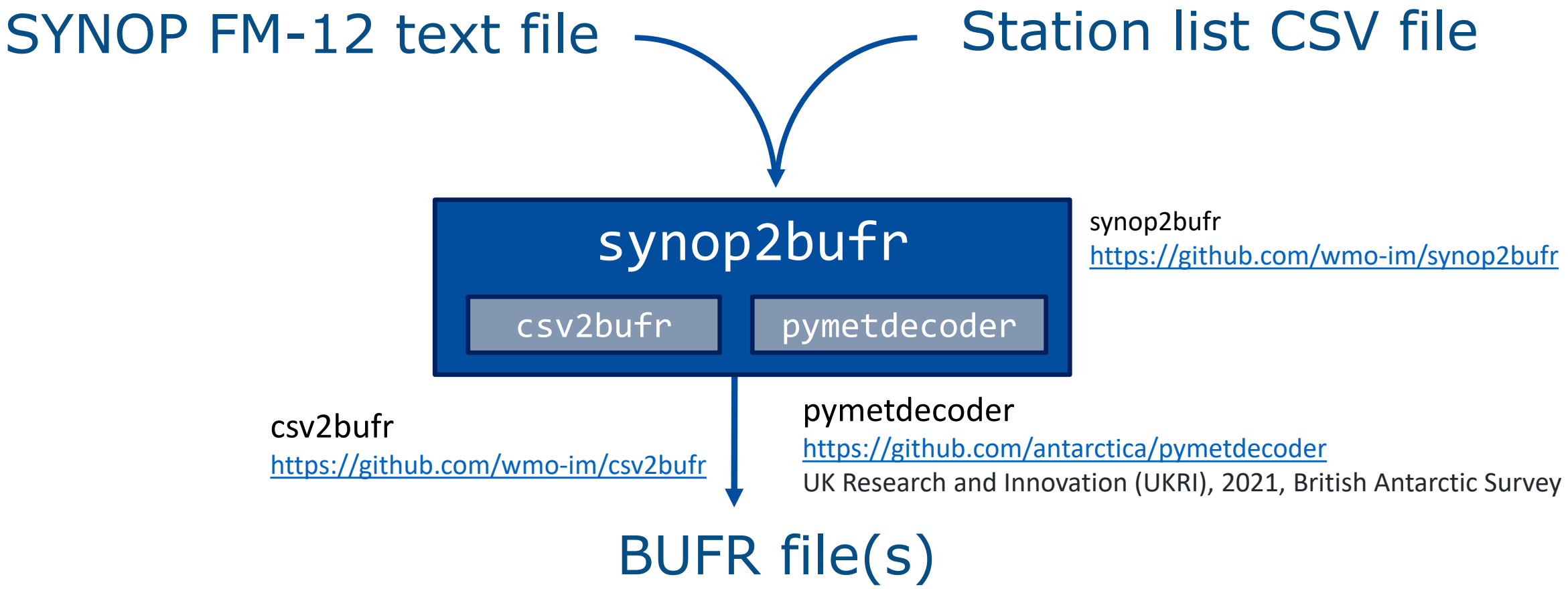
Additional metadata: method of calculating average temperature; sensor and station heights; exposure and measurement quality classification

Template = AWS in the Plugin Configuration refers to the 'daycli-template.json' as defined in https://github.com/wmo-im/csv2bufr-templates



WORLD METEOROLOGICAL ORGANIZATION

SYNOP-to-BUFR



synop2bufr
https://github.com/wmo-im/synop2bufr

csv2bufr
https://github.com/wmo-im/csv2bufr

pymetdecoder
https://github.com/antarctica/pymetdecoder
UK Research and Innovation (UKRI), 2021, British Antarctic Survey

WORLD METEOROLOGICAL ORGANIZATION

SYNOP-to-BUFR

FM-12 SYNOP reports

AAXX = data from fixed land station

```
SMRO01 YRBK 211200
AAXX 21121
15015 02999 02501 10103 21090 39765 42952 57020 60001 333 4/000 55310
0////
22591 3//// 60007 91003 91104=
15020 02997 23104 10130 21075 30177 40377 58020 60001 81041 333 4/000
55310
0//// 22547 3//// 60007 91008 91111=
```

YY = 21 of the month
GG = 12 hour of observation
iw = 1 Wind speed obtained from anemometer in m/s
TSI
TSI

Station list metadata

station_name	wigos_station_identifier	traditional_station_identifier	facility_type	latitude	longitude	elevation
OCNA SUGATAG	0-20000-0-15015	15015	Land (fixed)	47.7770616	23.9404603	503
BOTOSANI	0-20000-0-15020	15020	Land (fixed)	47.7356532	26.6455502	161
IASI	0-20000-0-15090	15090	Land (fixed)	47.1633333	27.6272222	74.29
CEAHLAU TOACA	0-20000-0-15108	15108	Land (fixed)	46.97751	25.94994	1897
CLUJ-NAPOCA	0-20000-0-15120	15120	Land (fixed)	46.7777705	23.5713053	410
BACAU	0-20000-0-15150	15150	Land (fixed)	46.5577778	26.8966667	174
MIERCUREA CIUC	0-20000-0-15170	15170	Land (fixed)	46.3713167	25.7726167	661
ARAD	0-20000-0-15200	15200	Land (fixed)	46.1335164	21.3536215	116.59
DEVA	0-20000-0-15230	15230	Land (fixed)	45.864923	22.8988062	240
SIBIU	0-20000-0-15260	15260	Land (fixed)	45.79018	24.036245	450
VARFU OMU	0-20000-0-15280	15280	Land (fixed)	45.4457928	25.456691	2504
CARANSEBES	0-20000-0-15292	15292	Land (fixed)	45.41667	22.22917	241
GALATI	0-20000-0-15310	15310	Land (fixed)	45.4729181	28.0323011	69
TULCEA	0-20000-0-15335	15335	Land (fixed)	45.1905065	28.8241608	4.36
RAMNICU VALCEA	0-20000-0-15346	15346	Land (fixed)	45.0888211	24.3628139	237
BUZAU	0-20000-0-15350	15350	Land (fixed)	45.1326633	26.8517319	97
SULINA	0-20000-0-15360	15360	Land (fixed)	45.1623111	29.7268286	69
DROBETA-TURNU SEVERIN	0-20000-0-15410	15410	Land (fixed)	44.6264587	22.6260737	77
BUCURESTI BANEASA	0-20000-0-15420	15420	Land (fixed)	44.510433	26.0781904	90

WORLD METEOROLOGICAL ORGANIZATION

Station metadata used in transformation to BUFR

FM-12 to BUFR (synop2bufr):

- **Add the WIGOS-station-ID**, derived by matching with traditional station identifier in FM-12 input
- **Add the latitude, longitude**, altitude and barometer height above sea-level in the BUFR content

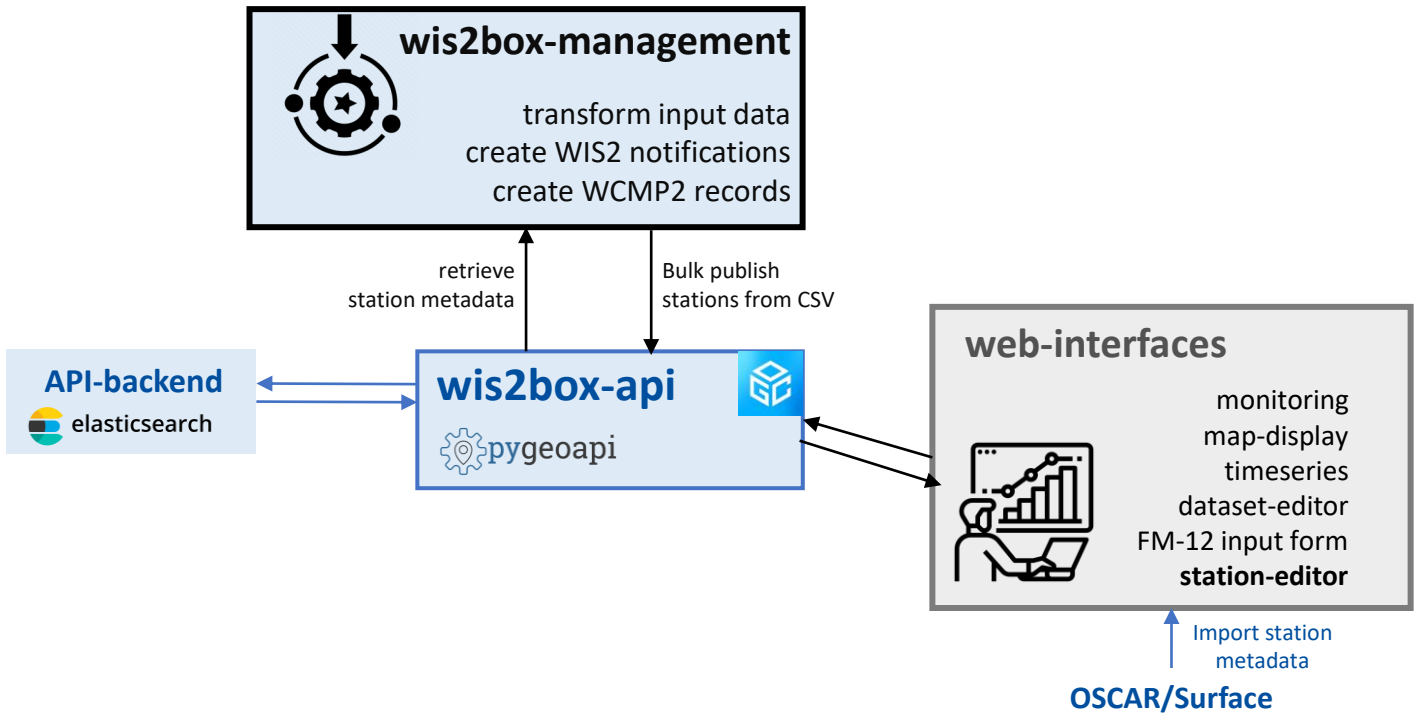
CSV to BUFR (csv2bufr):

- Check **WIGOS-station-ID** in input data is in station-list for associated topic
- For fixed land stations: check input-data-location is less than 1km from station metadata location

BUFR to BUFR (bufr2bufr):

- Splits by bulletin and **check WIGOS-station-ID** in station-list for associated topic
- If WIGOS-station-ID missing add it (by matching with traditional station identifier)
- if the location or station elevation are missing in the input BUFR use values from station metadata

Station metadata in the wis2box



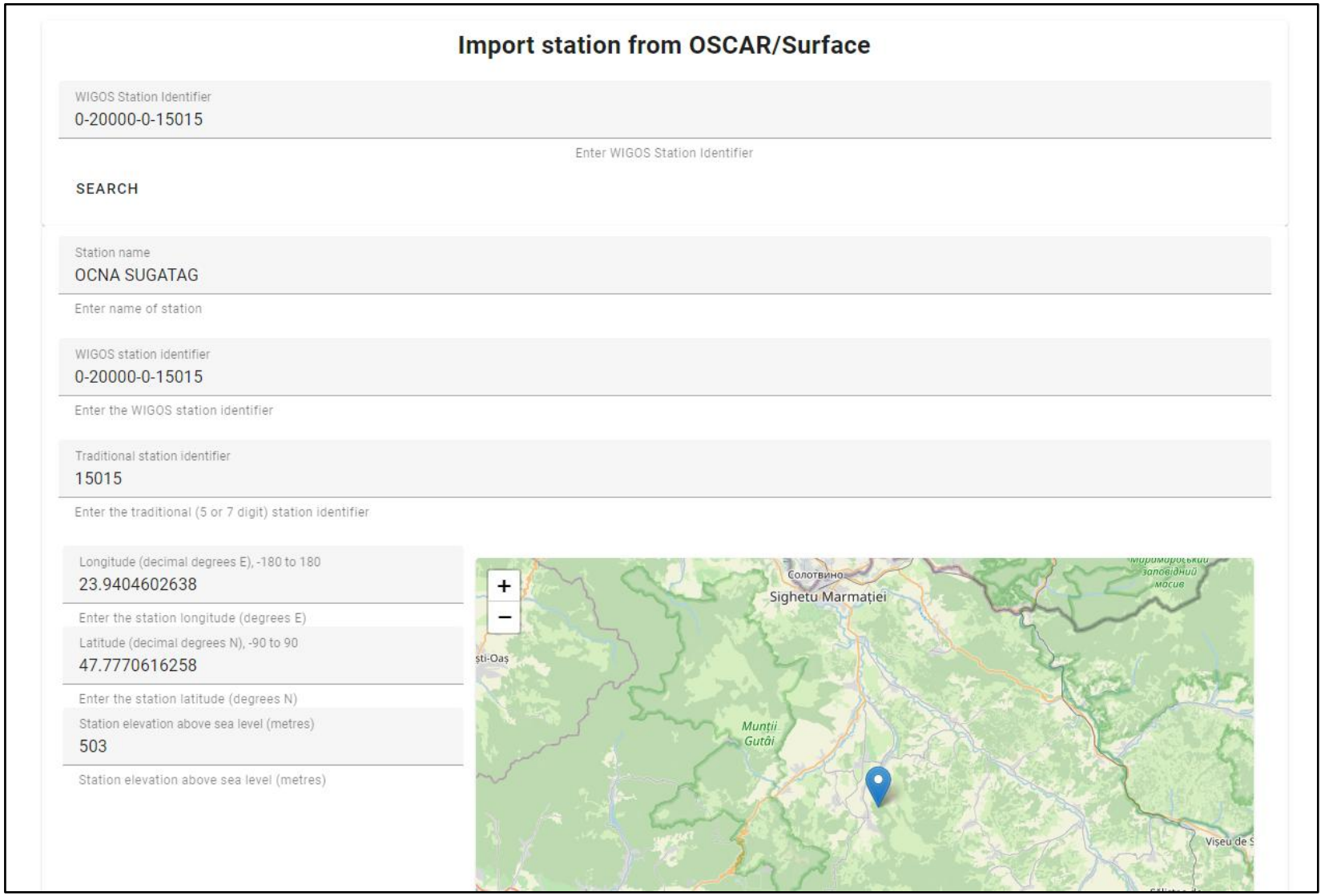
“stations” collection in wis2box-api can be populated in multiple ways:

- Provide station list as CSV and use command line inside wis2box-management container
- Interactively using the station editor inside the wis2box-webapp

WORLD METEOROLOGICAL ORGANIZATION

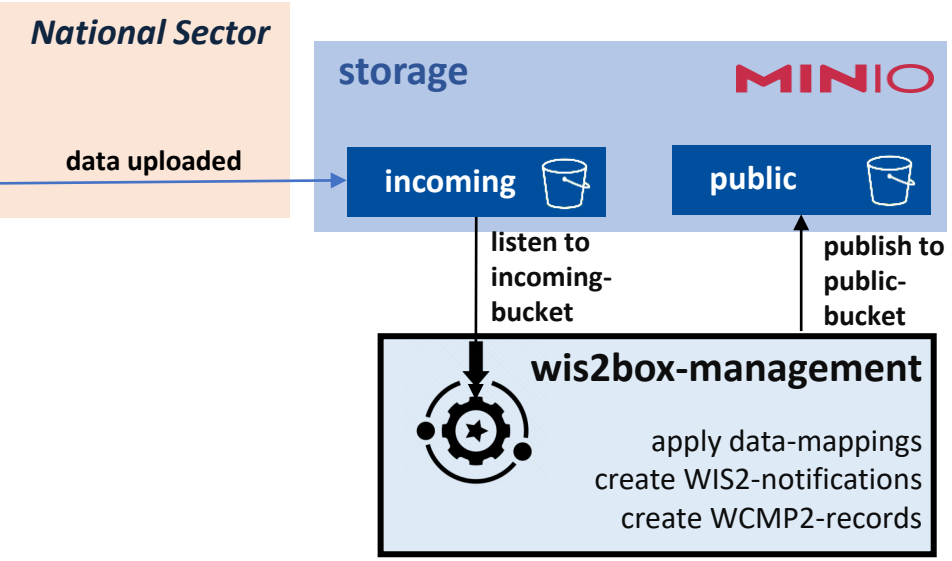
How to configure stations in wis2box-webpp

You can update the wis2box station-list using the station editor in the wis2box webapp, importing data from OSCAR <wis2box-host-url>/wis2box-webapp/station



WORLD METEOROLOGICAL ORGANIZATION

wis2box data ingest workflow

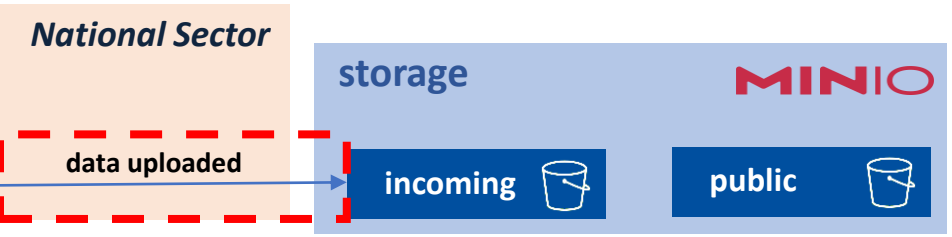


Dataset driven workflow triggered when wis2box-management receives a notification from the storage service

wis2box-management **matches incoming data** with **datasets** based on **file-path**
data plugins are applied to **transform & publish the data**

WORLD METEOROLOGICAL ORGANIZATION

How to upload data to wis2box



Data can be uploaded to wis2box-incoming in different ways

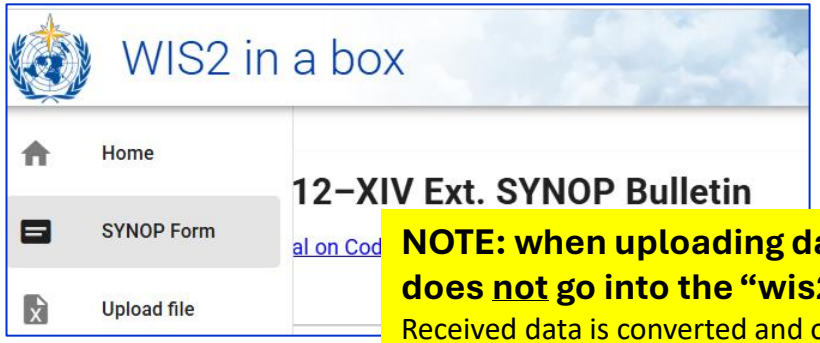
Optimal solution will depend on systems used in national sector

Manual upload methods:

- **Interfaces in wis2box-webapp:** manual submission using FM-12 form or manual file upload interface
- **Using MinIO UI:** manually create path and upload data, useful for testing data-workflow

Automating data upload (recommended!):

- **Scripts:** write code to upload data using MinIO client software
- **Using SFTP protocol:** forward data to MinIO SFTP-endpoint



NOTE: when uploading data in wis2box-webapp, received data does not go into the "wis2box-incoming" bucket

Received data is converted and output is stored in "wis2box-public" bucket directly

WORLD METEOROLOGICAL ORGANIZATION

Summary

To publish data using wis2box you must **first define a dataset** and **publish WCMP2**

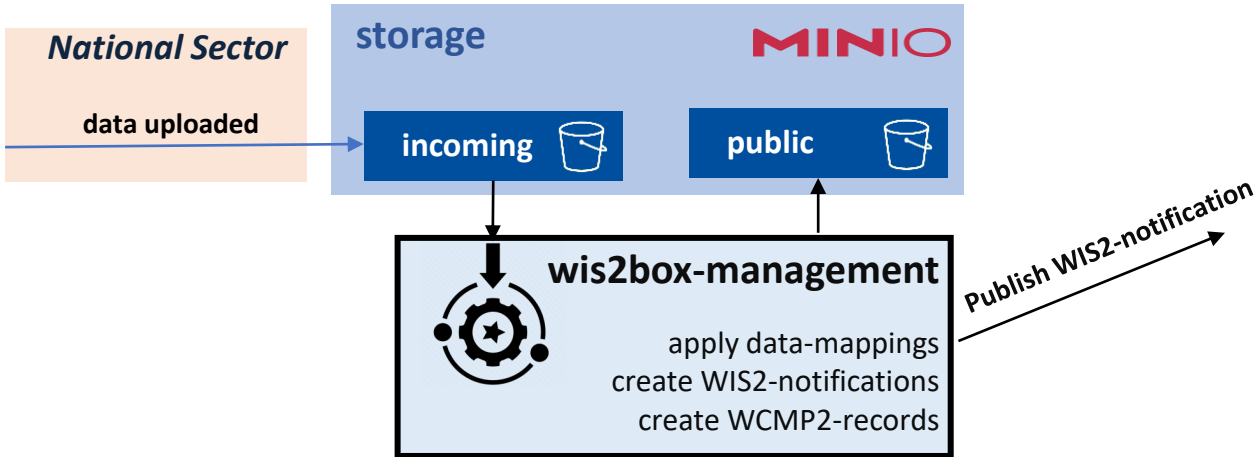
In wis2box, **datasets** are used to define both the **discovery metadata** and the **data plugins**

Data plugins determine the actions taken to **transform** and **publish** the data

wis2box includes pre-defined data plugins to enable users to transform data to BUFR using ecCodes

The **dataset-editor** makes it easier for users to define the correct data-plugin for their data-type by providing "**templates**" that can be selected when creating a dataset

wis2box-management container **listens to** events from **MinIO** to **trigger data processing** The file-path of the incoming data is used to determine the corresponding dataset



WORLD METEOROLOGICAL ORGANIZATION

Our wis2box-training instance

For this week's training we will use a **shared wis2box-instance** (=one server, multiple WIS2 nodes)

- wis2box 1.3.0 (pre-release!), instance hosted on the European Weather Cloud

Homepage (wis2box-ui) at:
https://wis2box-training.wis2dev.io

Interactive Web Interface (wis2box-webapp):
https://wis2box-training.wis2dev.io/wis2box-webapp

WIS2 Node MQTT broker:
over plain MQTT: mqtt://wis2box-training.wis2dev.io (port 1883)
over secure web-sockets: wss://wis2box-training.wis2dev.io/mqtt (port 443)

First practical session:
Everyone to setup their own datasets on this instance and view WCMP2 being published over MQTT

WORLD METEOROLOGICAL ORGANIZATION

Thank you.

WORLD
METEOROLOGICAL
ORGANIZATION

wmo.int