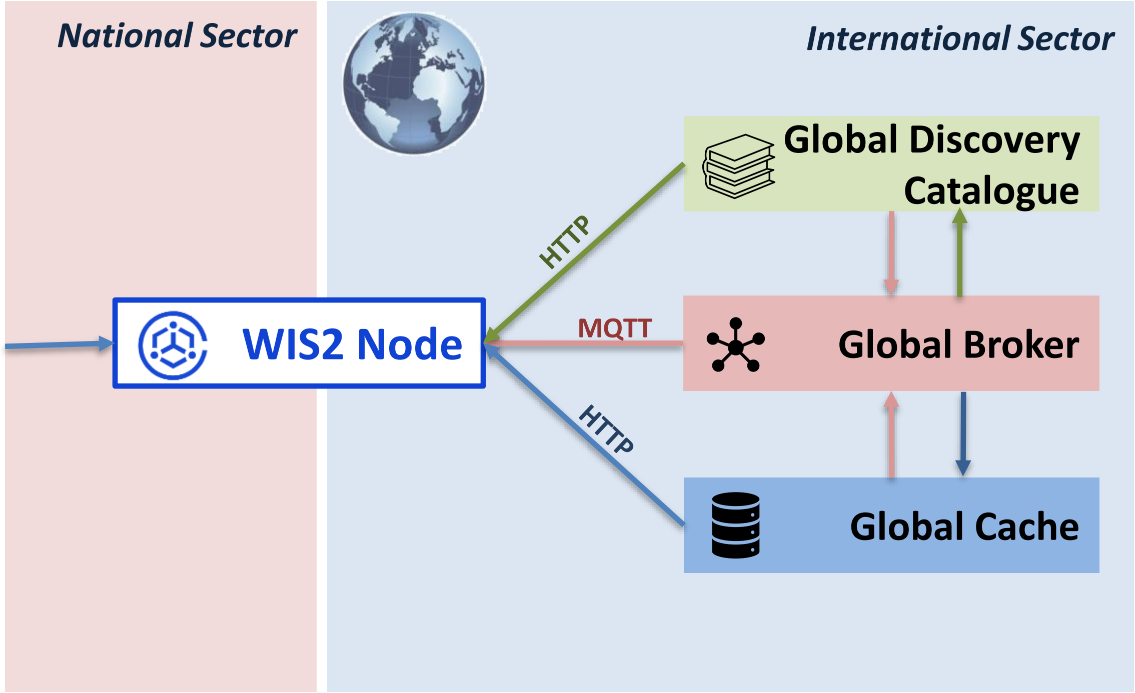
WIS2 in a box (wis2box)

WORLD
METEOROLOGICAL
ORGANIZATION

Reminder: What is a WIS2 Node?

A WIS2 node is composed of 2 endpoints that need to be exposed over the __public internet__:

- ***MQTT broker***: to publish WIS2 notifications for metadata and data
- ***HTTP data server***: to enable the download of data files and metadata records



Global Discovery Catalogues download all valid **WCMP2 records** from the HTTP-endpoint for notifications on topic=*cache/a/wis2/+/metadata*

Global Brokers subscribes to GC and WIS2 Nodes and republishes all valid WIS2-notifications on origin/a/wis2/… and cache/a/wis2/..

Global Caches download data from topic=*origin/a/wis2/+/data/core/# and origin/a/wis2/+metadata* **and publish** *on* topic=***cache/a/wis2/…***

MQTT channel defined by the **__WIS2 Topic Hierarchy__ (WTH)** standard

Discovery Metadata records defined by **__WCMP2__** standard

MQTT payload defined by the **__WIS2 Notification Message (WNM)__** standard

WORLD
METEOROLOGICAL
ORGANIZATION

What is WIS2 in a box?

WIS2-in-a-BOX

- WIS2 in a box (wis2box) is a **Reference Implementation of a WIS2 Node**
- Designed to be **cost-effective** and **low-barrier** to operate*
- Developed as **a Docker Compose stack** using existing **Free and Open Source** software, coming from existing FOSS and wis2box-specific development
  - wis2box source code at: https://github.com/World-Meteorological-Organization/wis2box
- Developed by WMO Secretariat together with Canada to **accelerate the WIS2 implementation**
- Currently over 70 WIS2 Nodes are using wis2box to share data on WIS2

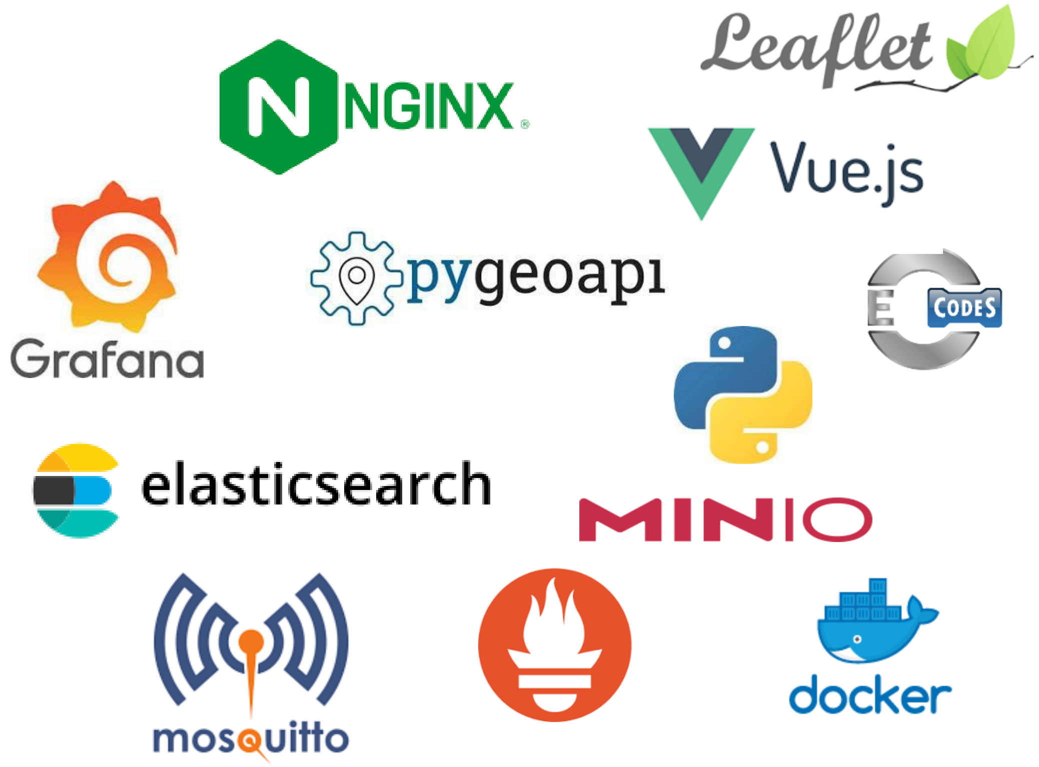
*wis2box hosting requirements:

- minimum 2 vCPUs with 4GB Memory and 24GB of local storage
- requires Python, Docker and Docker Compose pre-installed
- HTTP and MQTT ports routed to a publicly accessible address
- See documentation at https://docs.wis2box.wis.wmo.int

WORLD
METEOROLOGICAL
ORGANIZATION

WIS2 in a box is Free and Open

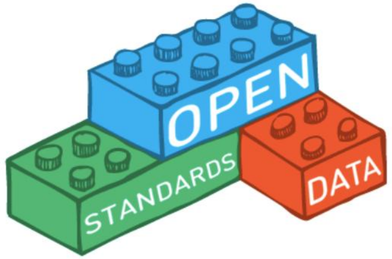
Free and Open Source Software



Open Standards

W3C®
ISO
I E T F®

- MQTT
- GeoJSON
- OGC APIs



Docker and Docker Compose

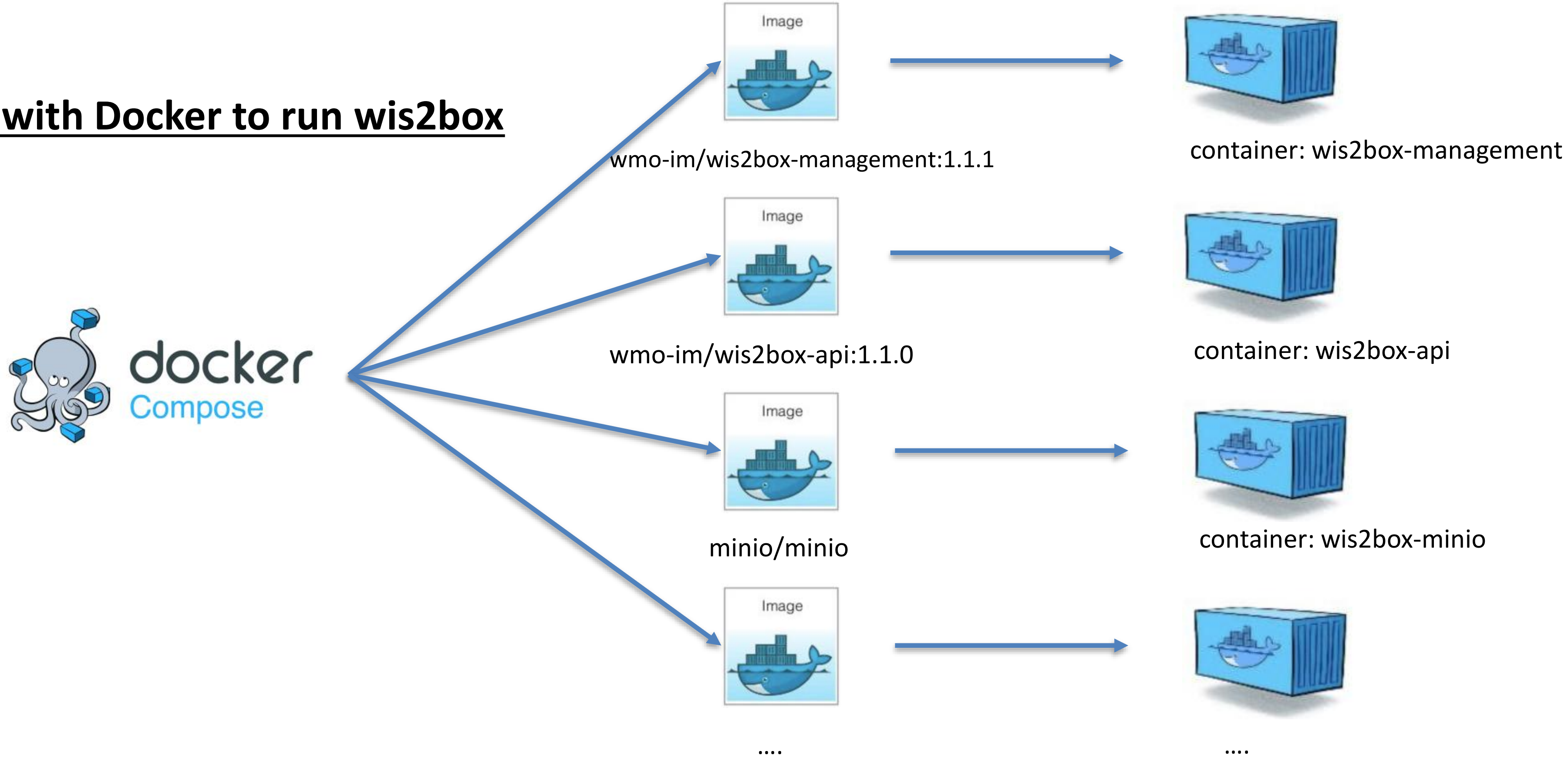
wis2box uses Docker and Docker Compose to define a set of services, using pre-built Docker images

Python script ‘wis2box-ctl.py’ provides a wrapper around Docker Compose commands to interact with wis2box

Software required on the host to run wis2box:

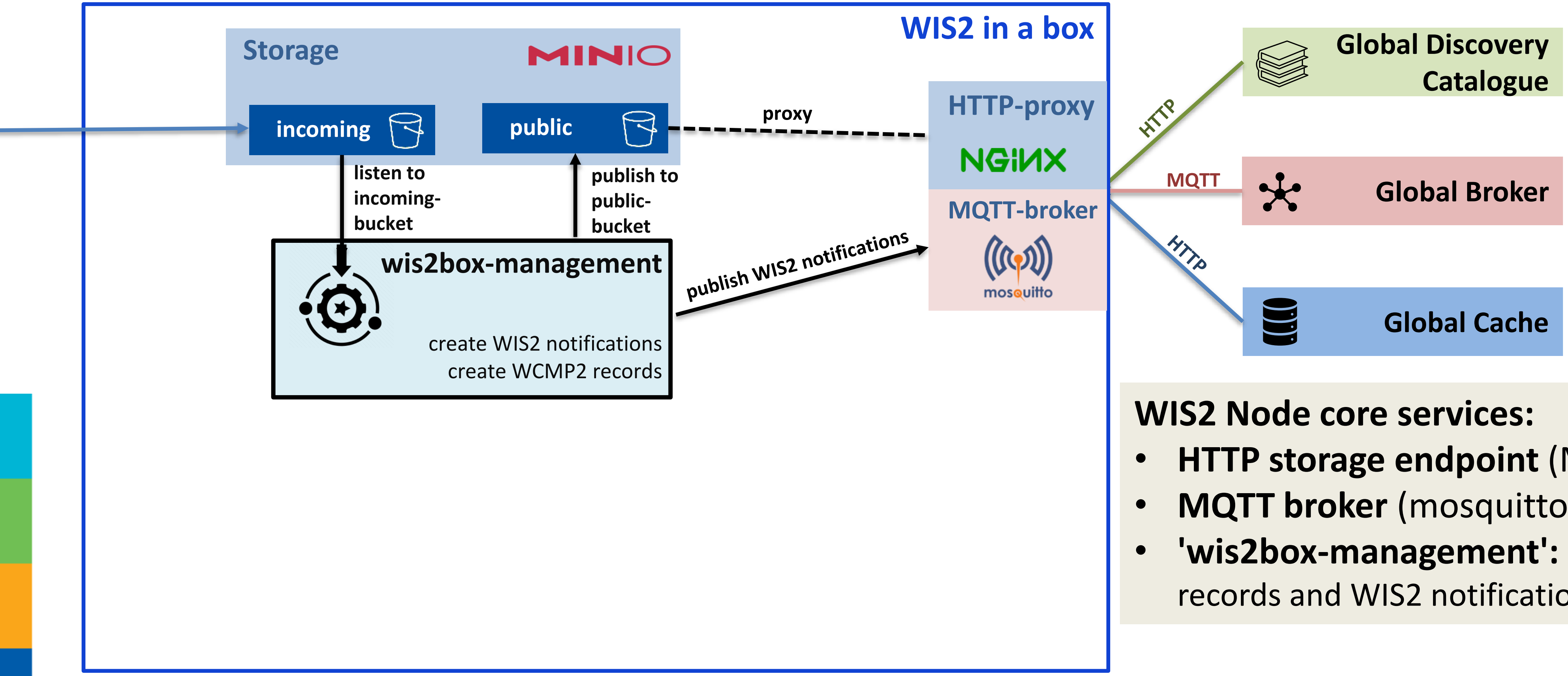
- **Python**
- **Docker**
- **Docker Compose**

<u>you do not need to be familiar with Docker to run wis2box</u>



WORLD METEOROLOGICAL ORGANIZATION

WIS2 in a box core services

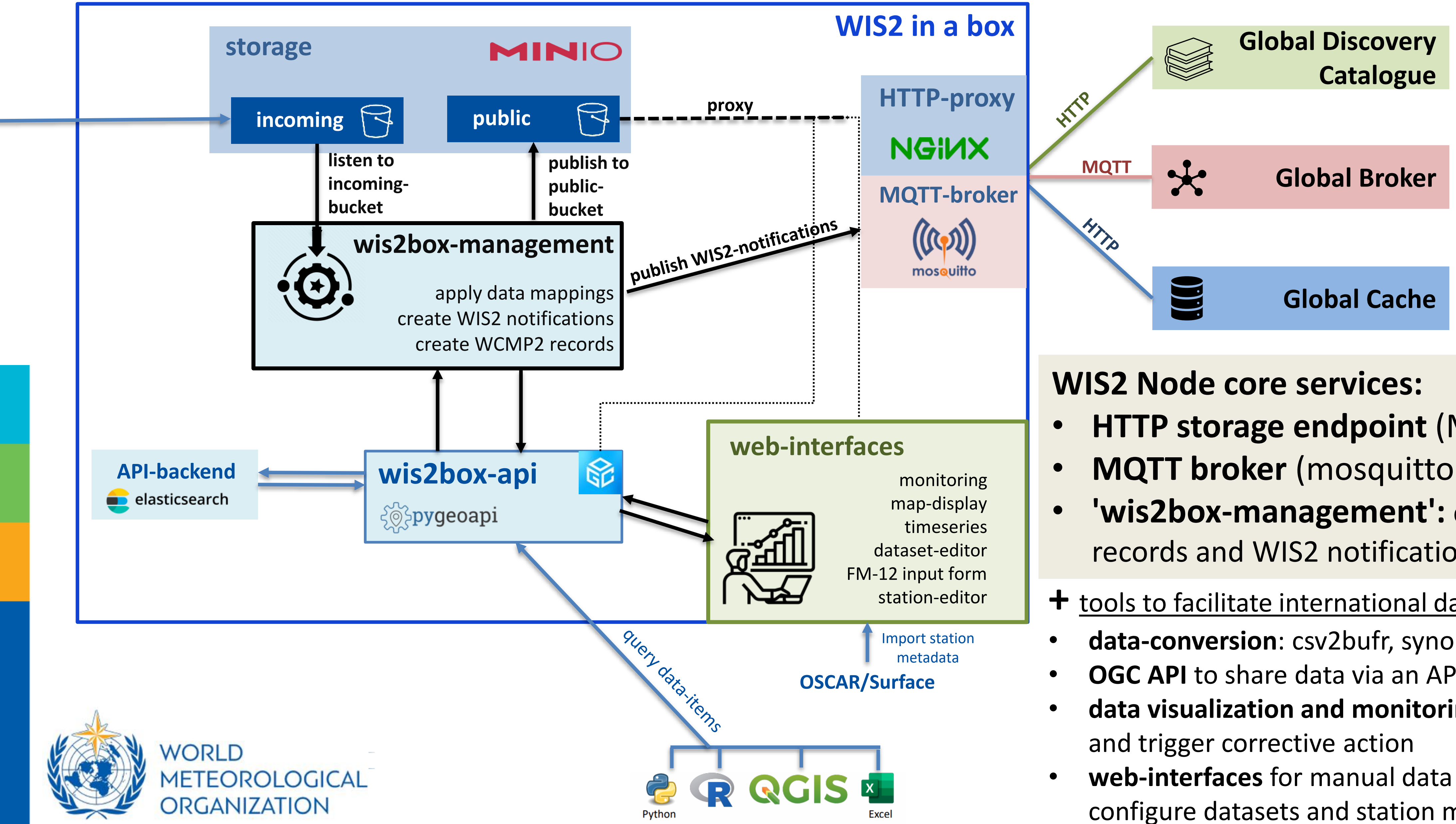


WIS2 Node core services:

- **HTTP storage endpoint** (MinIO+nginx)
- **MQTT broker** (mosquitto)
- **'wis2box-management':** create WCMP2 records and WIS2 notifications

WORLD METEOROLOGICAL ORGANIZATION

WIS2 in a box services



WIS2 Node core services:

- **HTTP storage endpoint** (MinIO+nginx)
- **MQTT broker** (mosquitto)
- **'wis2box-management':** create WCMP2 records and WIS2 notifications

+ tools to facilitate international data sharing:

- **data-conversion**: csv2bufr, synop2bufr, bufr2bufr
- **OGC API** to share data via an API
- **data visualization and monitoring** to detect issues and trigger corrective action
- **web-interfaces** for manual data input (FM-12/CSV), configure datasets and station metadata

WORLD METEOROLOGICAL ORGANIZATION

How wis2box works: datasets in wis2box

Dataset-driven workflow in wis2box-managment: a corresponding dataset needs to be configured to publish WIS2-notifications for incoming data

Datasets in the wis2box consist of two components:

- **Discovery Metadata** to publish a WCMP2 record for the Global Discovery Catalogue
- **Dataset Mappings** that define the plugins used to transform and publish the data

Datasets can be configured using the dataset-editor in wis2box-webapp

... or using YAML files and the command line in wis2box-management

https://training.wis2box.wis.wmo.int/practical-sessions/configuring-wis2box-datasets/

WORLD
METEOROLOGICAL
ORGANIZATION

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

Datasets in the wis2box

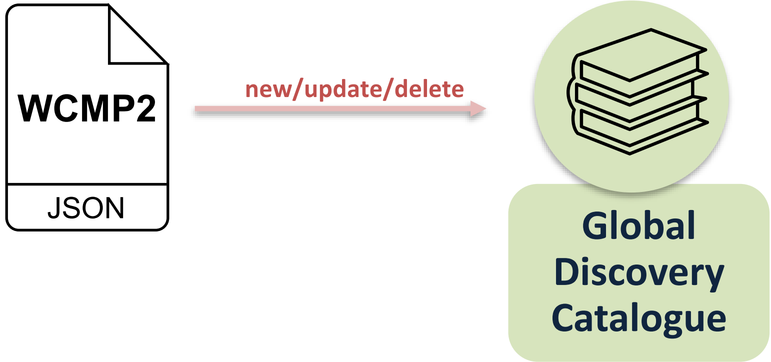
Two ways to configure a new dataset in the wis2box:

- 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

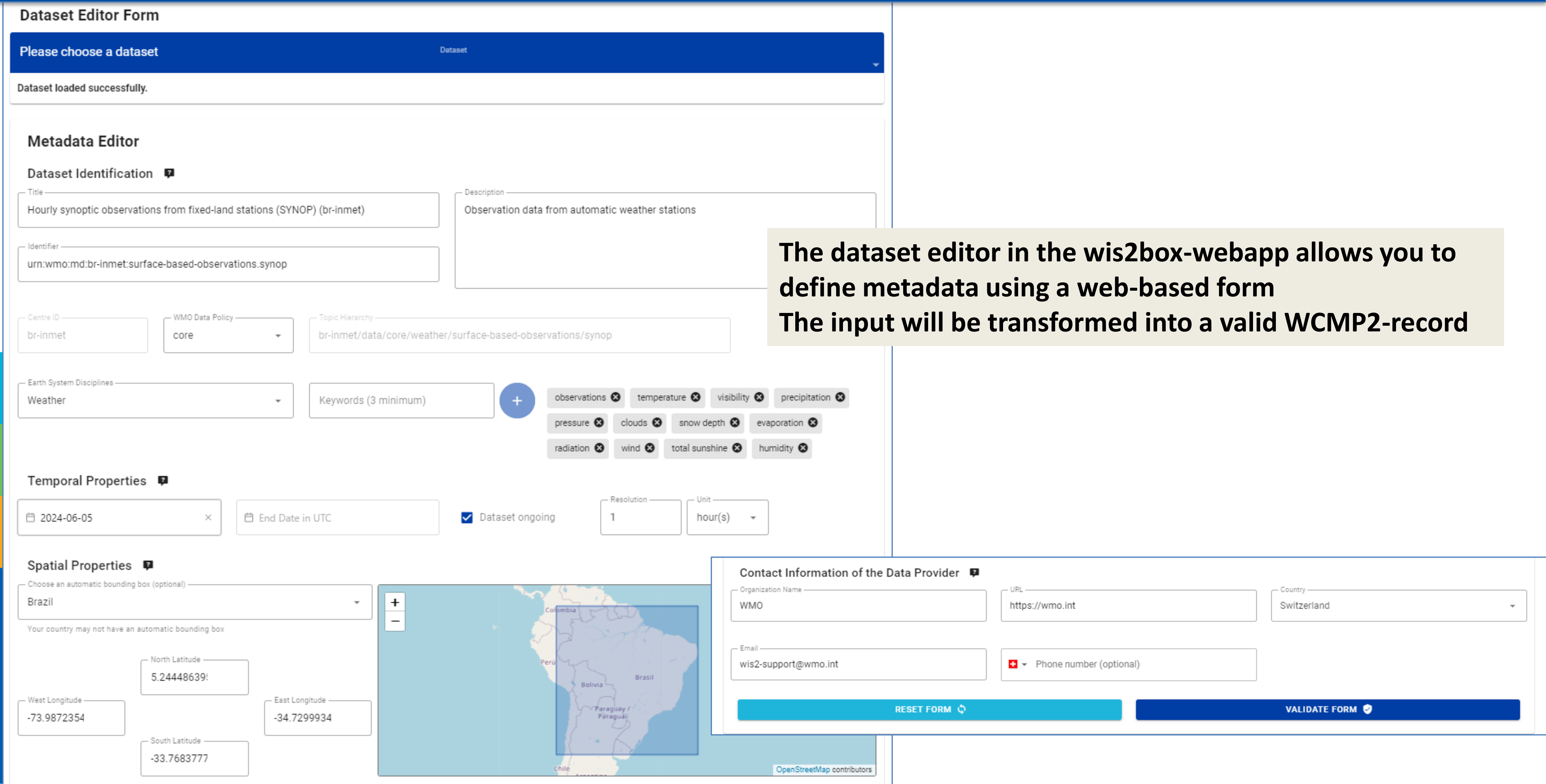
update WCMP2 notification on ***origin/a/wis2/<centre-id>/metadata***
for every updated dataset published

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



WORLD METEOROLOGICAL ORGANIZATION

Discovery Metadata



Dataset Editor Form

Please choose a dataset

Dataset loaded successfully.

Metadata Editor

Dataset Identification

Title: Hourly synoptic observations from fixed-land stations (SYNOP) (br-inmet)

Identifier: urn:wmo:md:br-inmet:surface-based-observations.synop

Description: Observation data from automatic weather stations

Centre ID: br-inmet

WMO Data Policy: core

Topic Hierarchy: br-inmet/data/core/weather/surface-based-observations/synop

Earth System Disciplines: Weather

Keywords (3 minimum) +

observations ⊗ temperature ⊗ visibility ⊗ precipitation ⊗ pressure ⊗ clouds ⊗ snow depth ⊗ evaporation ⊗ radiation ⊗ wind ⊗ total sunshine ⊗ humidity ⊗

Temporal Properties

2024-06-05 ×

End Date in UTC

☑ Dataset ongoing

Resolution: 1

Unit: hour(s)

Spatial Properties

Choose an automatic bounding box (optional): Brazil

Your country may not have an automatic bounding box

North Latitude: 5.24448639

West Longitude: -73.9872354

East Longitude: -34.7299934

South Latitude: -33.7683777

Contact Information of the Data Provider

Organization Name: WMO

URL: https://wmo.int

Country: Switzerland

Email: wis2-support@wmo.int

Phone number (optional)

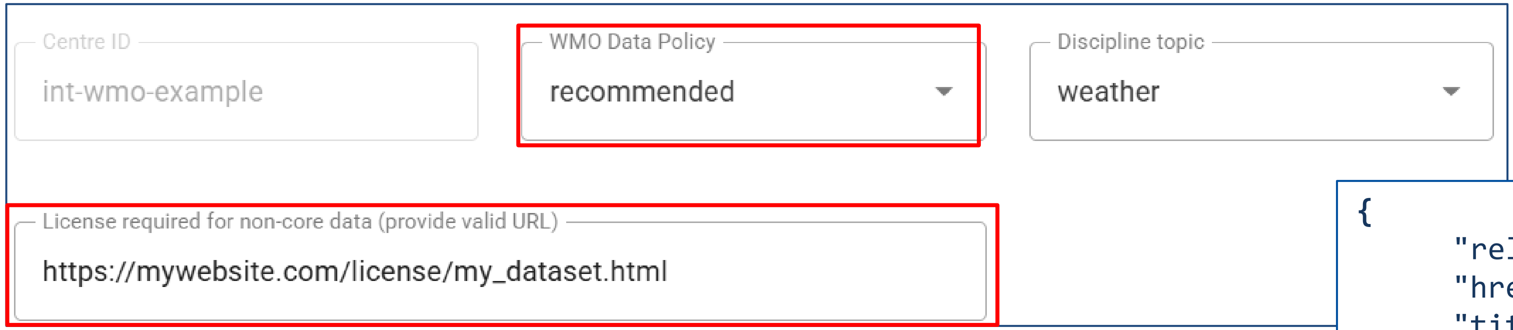
The dataset editor in the wis2box-webapp allows you to define metadata using a web-based form

The input will be transformed into a valid WCMP2-record

Discovery Metadata

WCMP2 for a ‘recommended’-dataset shall include a link with rel=‘licence’

Dataset-editor in wis2box-webapp requires you to provide a license-URL when setting WMO Data Policy = ‘Recommended’



Centre ID: int-wmo-example

WMO Data Policy: recommended

Discipline topic: weather

License required for non-core data (provide valid URL): https://mywebsite.com/license/my_dataset.html

Additional object added in links-array of WCMP2 :

```
{
      "rel": "license",
      "href": "https://mywebsite.com/license/my_dataset.html",
      "title": "License for this dataset",
      "type": "text/html“
}
```

A license URL can be provided by:

- URL on your organization website
- Public copyright license from Creative Commons
- The wis2box-instance itself, using a file in wis2box-public

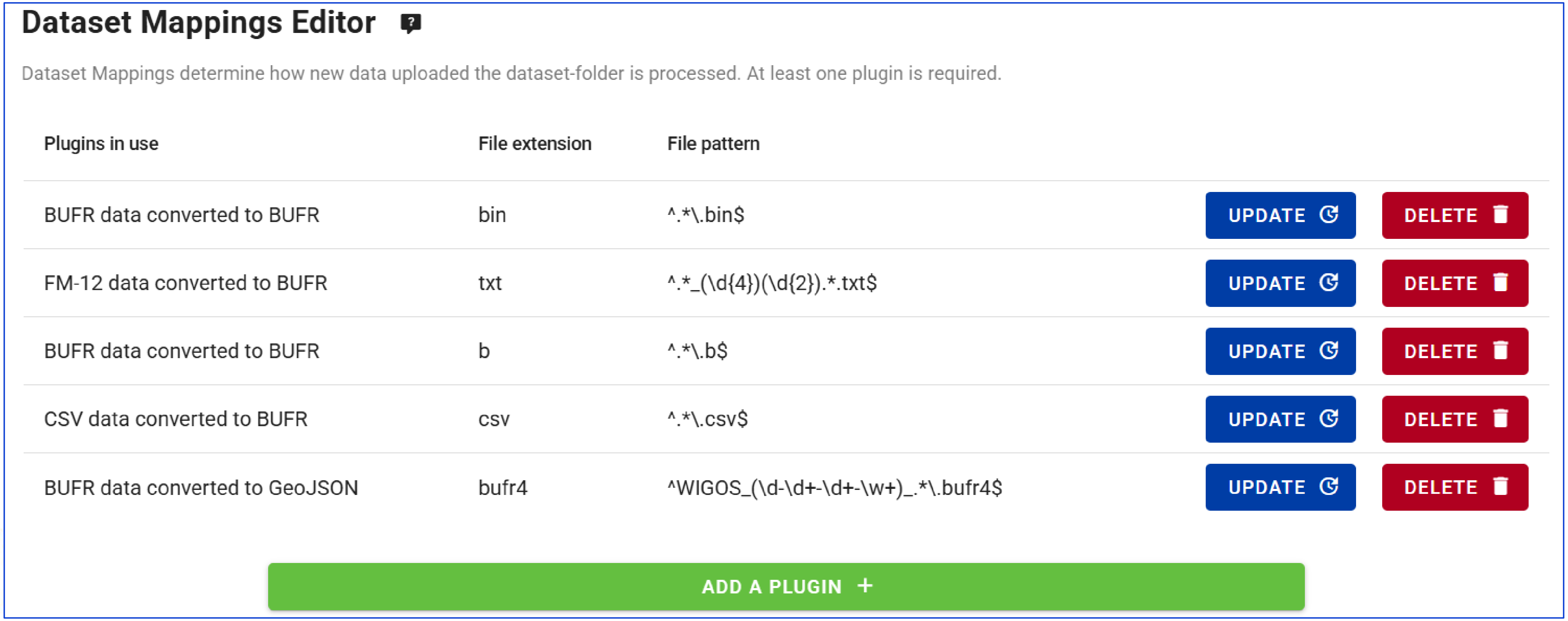
WORLD METEOROLOGICAL ORGANIZATION

Dataset mappings

The wis2box dataset editor includes the *Dataset Mappings Editor*:

Dataset Mappings map each dataset in wis2box to a set of **data plugins**

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



Dataset Mappings Editor

Dataset Mappings determine how new data uploaded the dataset-folder is processed. At least one plugin is required.

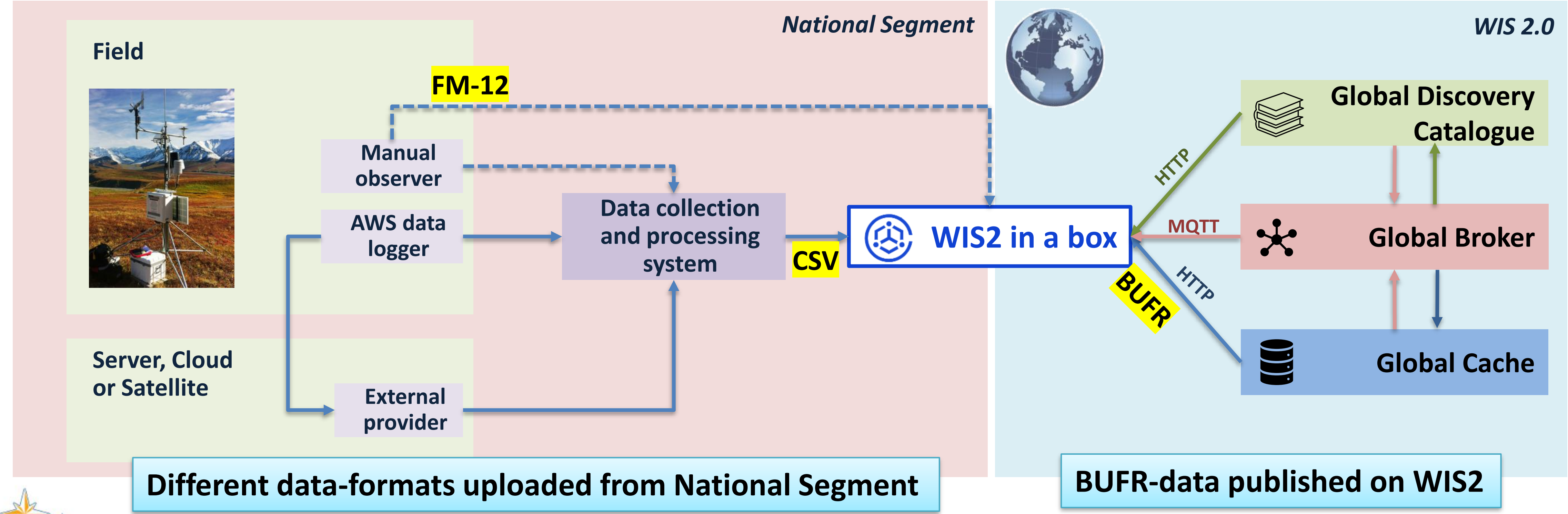
Plugins in use	File extension	File pattern
BUFR data converted to BUFR	bin	^.*\.bin$
FM-12 data converted to BUFR	txt	^.*_(\d{4})(\d{2}).*.txt$
BUFR data converted to BUFR	b	^.*\.b$
CSV data converted to BUFR	csv	^.*\.csv$
BUFR data converted to GeoJSON	bufr4	^WIGOS_(\d-\d+-\d+-\w+)_.*\.bufr4$

How wis2box works: data plugins

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

Example: some of the wis2box data plugins enable data transformation to BUFR prior to publication

- **csv2bufr** enables any system to upload a CSV-data-extract without needing local BUFR conversion tools
- ***synop2bufr*** enables direct publication of FM-12 synop reports from manual observers



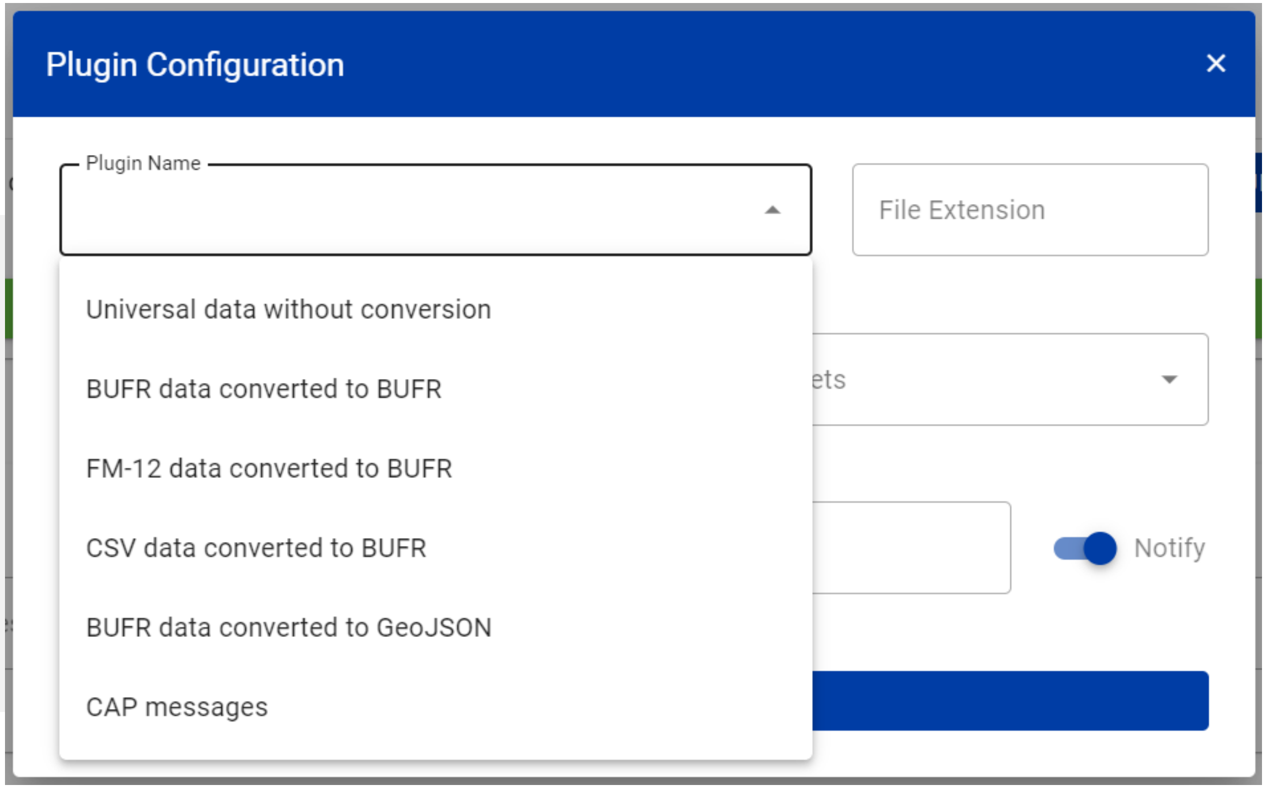
Different data-formats uploaded from National Segment

BUFR-data published on WIS2

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



Plugin Configuration

Plugin Name

File Extension

Universal data without conversion

BUFR data converted to BUFR

FM-12 data converted to BUFR

CSV data converted to BUFR

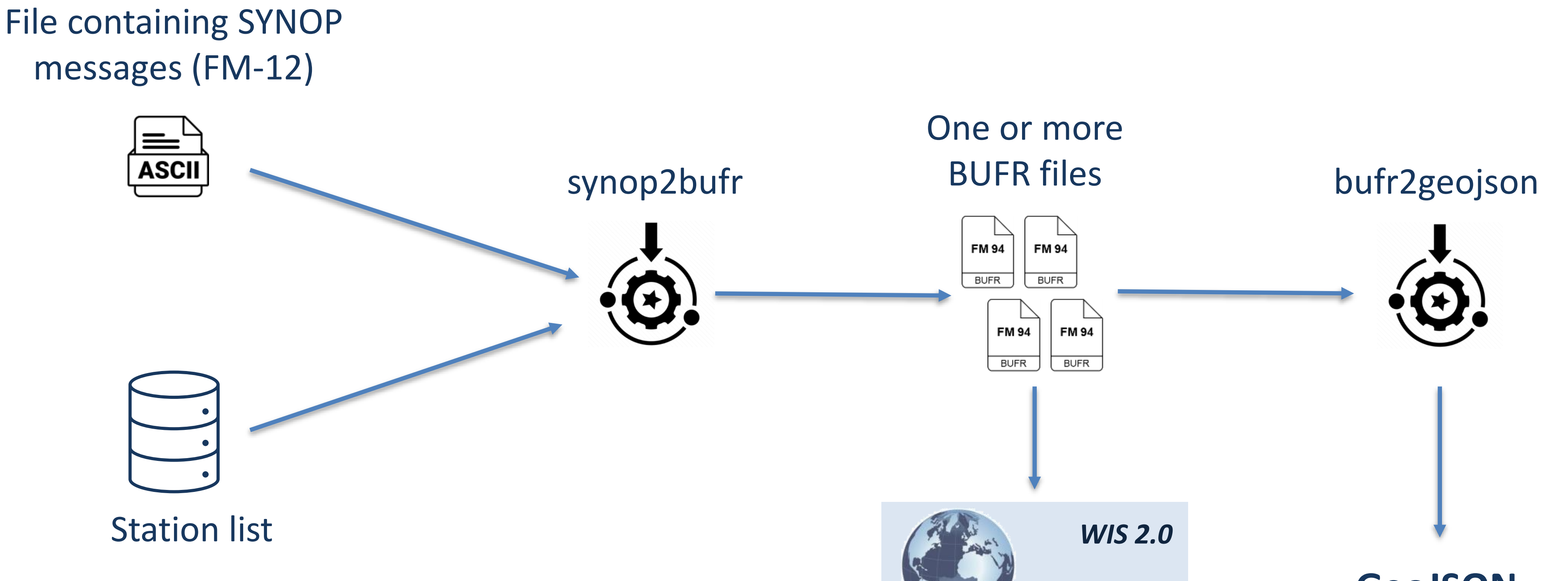
BUFR data converted to GeoJSON

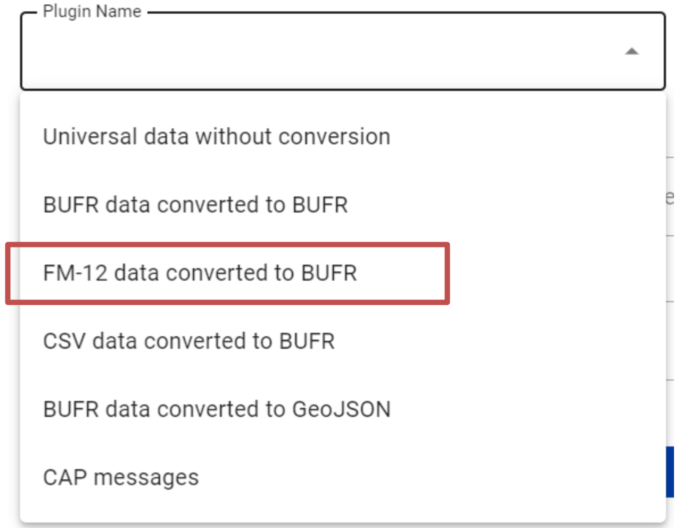
CAP messages

Developers are encouraged to contribute new data plugins to wis2box!

Data plugins use an abstract model/approach to enable extensibility and reuse, enabling developers to contribute, see github.com/wmo-im/wis2box/tree/main/wis2box-management/wis2box/data

wis2box data plugins: synop2bufr

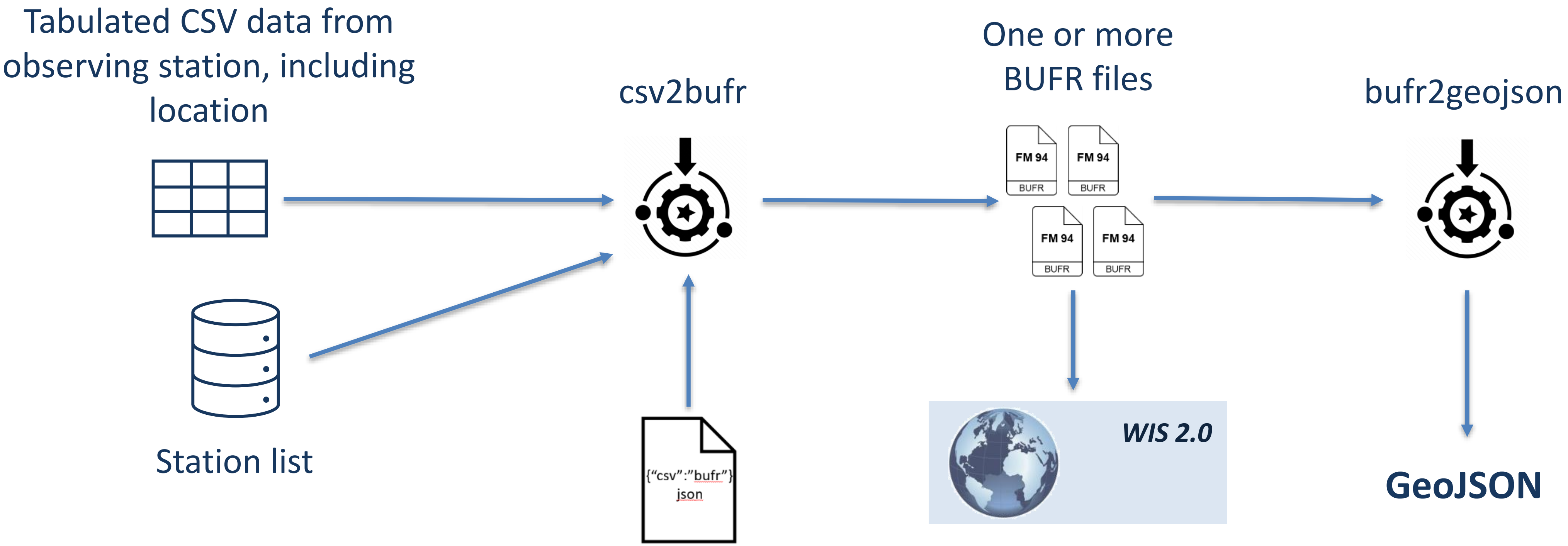




- Universal data without conversion
- BUFR data converted to BUFR
- FM-12 data converted to BUFR
- CSV data converted to BUFR
- BUFR data converted to GeoJSON
- CAP messages

```
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



Tabulated CSV data from observing station, including location

Station list

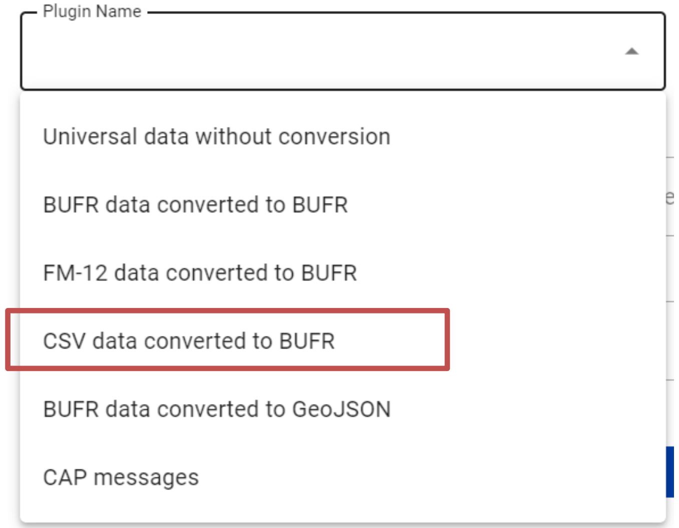
csv2bufr

mapping template

One or more BUFR files

bufr2geojson

GeoJSON

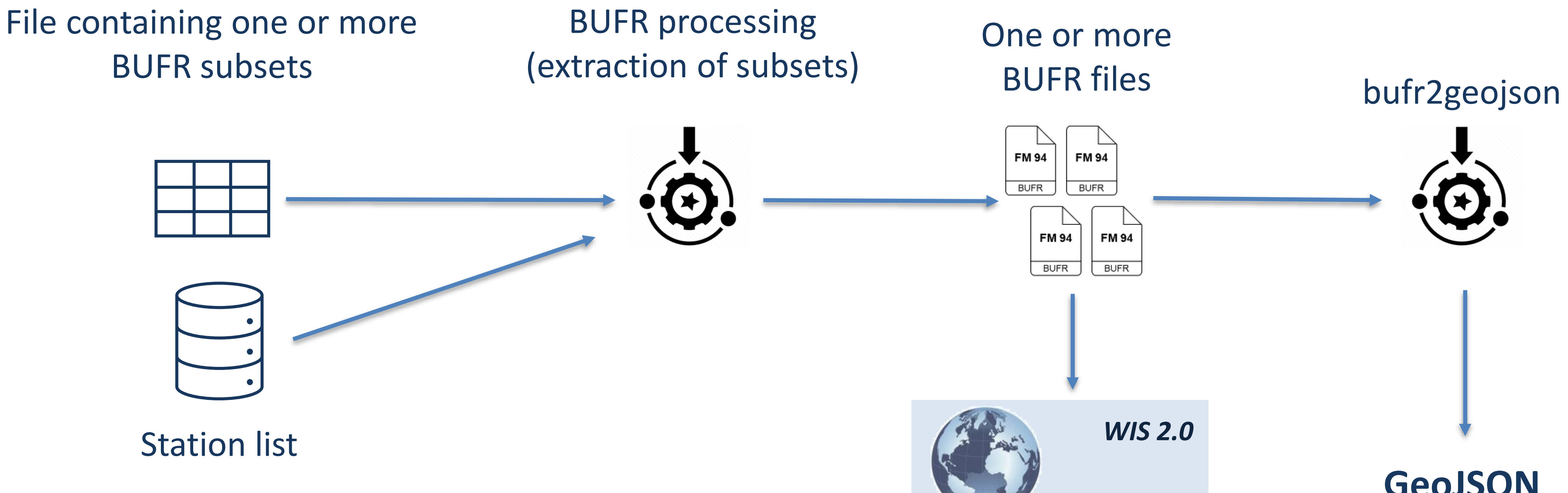


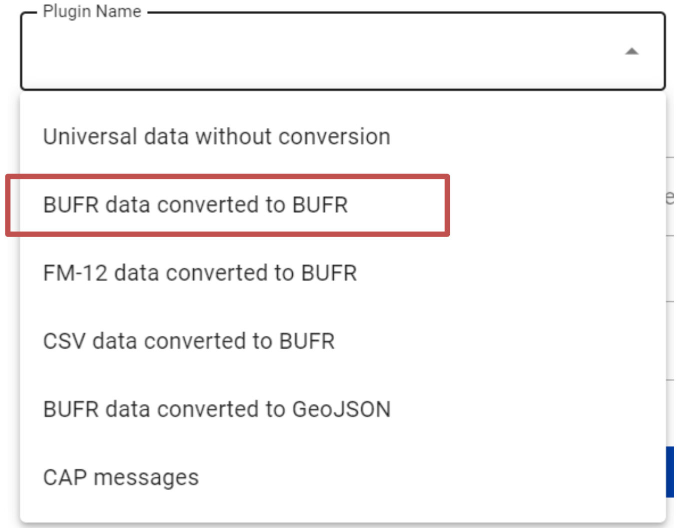
Plugin Name

- Universal data without conversion
- BUFR data converted to BUFR
- FM-12 data converted to BUFR
- CSV data converted to BUFR
- BUFR data converted to GeoJSON
- CAP messages

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

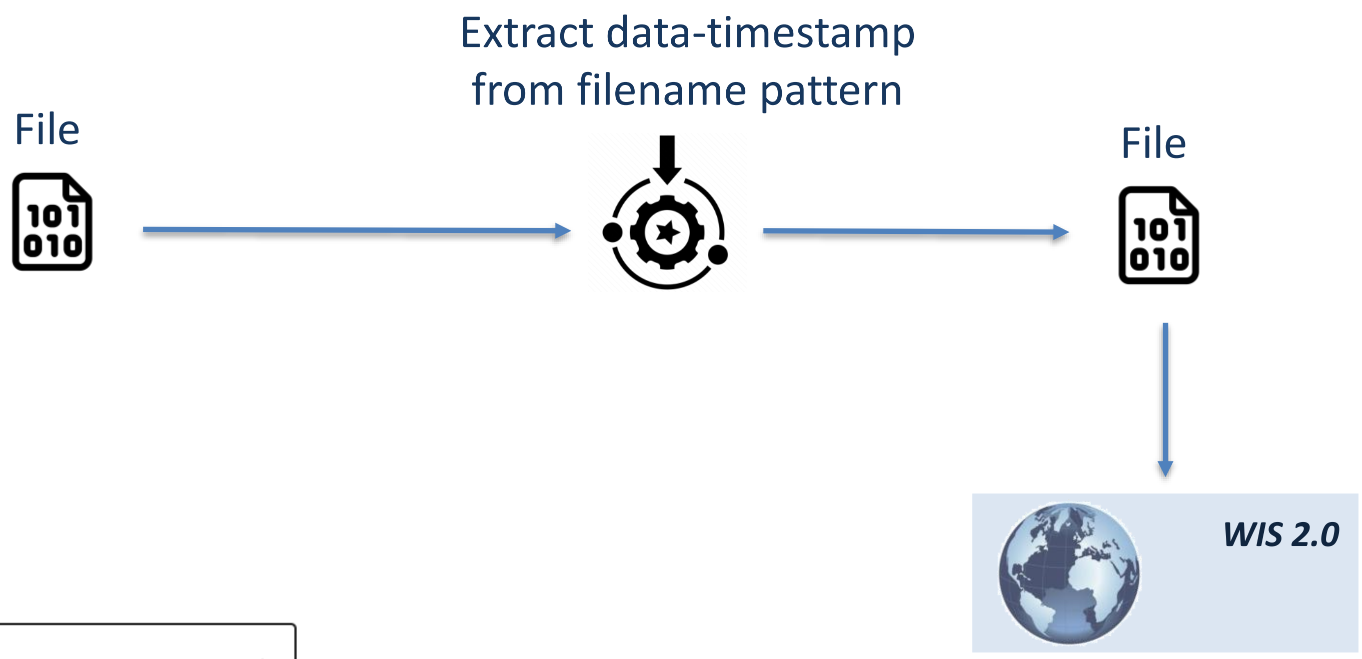
wis2box data plugins: bufr2bufr




Plugin Name
Universal data without conversion
BUFR data converted to BUFR
FM-12 data converted to BUFR
CSV data converted to BUFR
BUFR data converted to GeoJSON
CAP messages

```
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

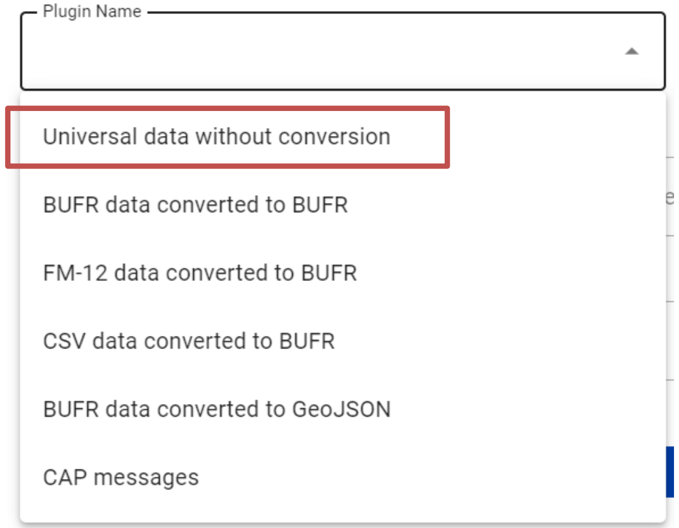


File

Extract data-timestamp from filename pattern

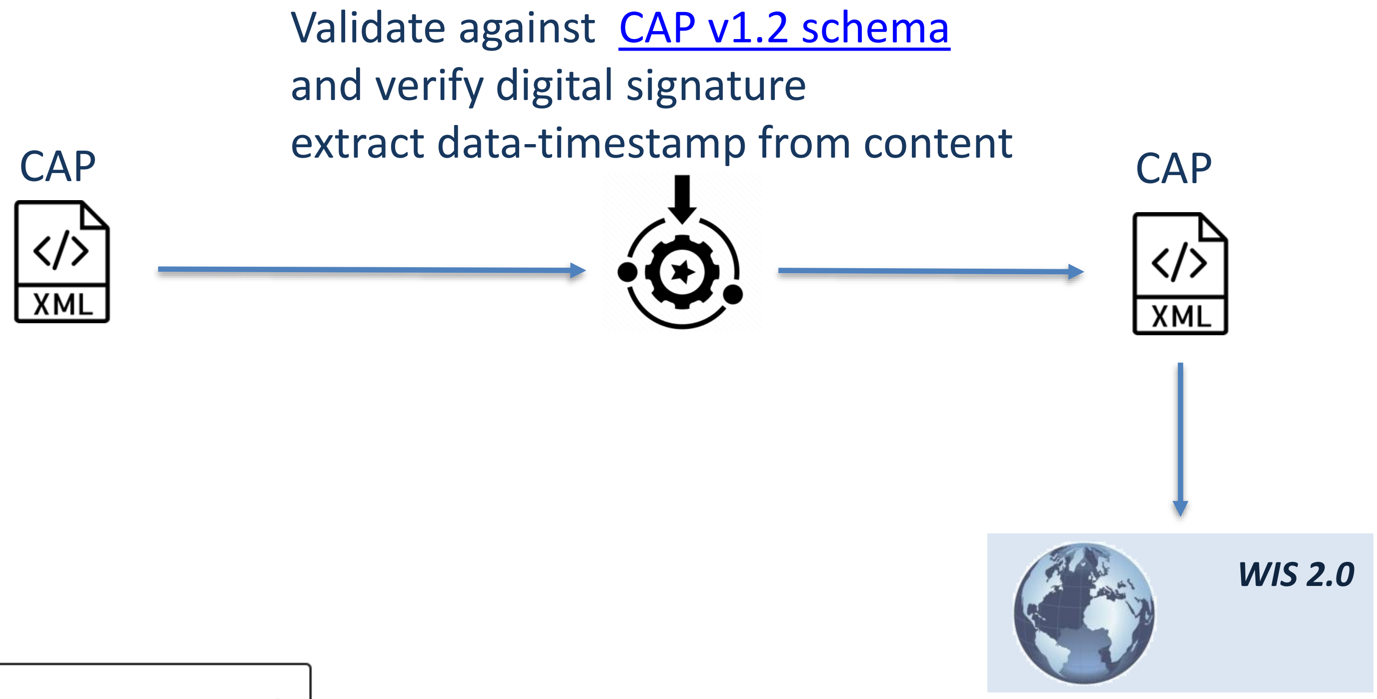
File

WIS 2.0

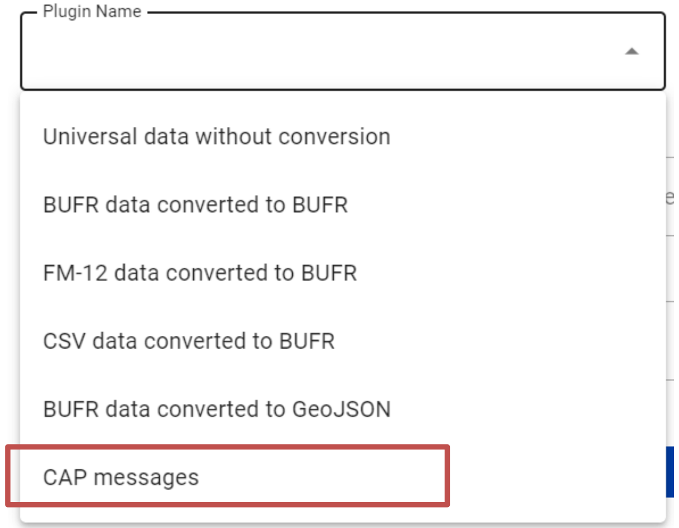

Plugin Name
Universal data without conversion
BUFR data converted to BUFR
FM-12 data converted to BUFR
CSV data converted to BUFR
BUFR data converted to GeoJSON
CAP messages

```
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)



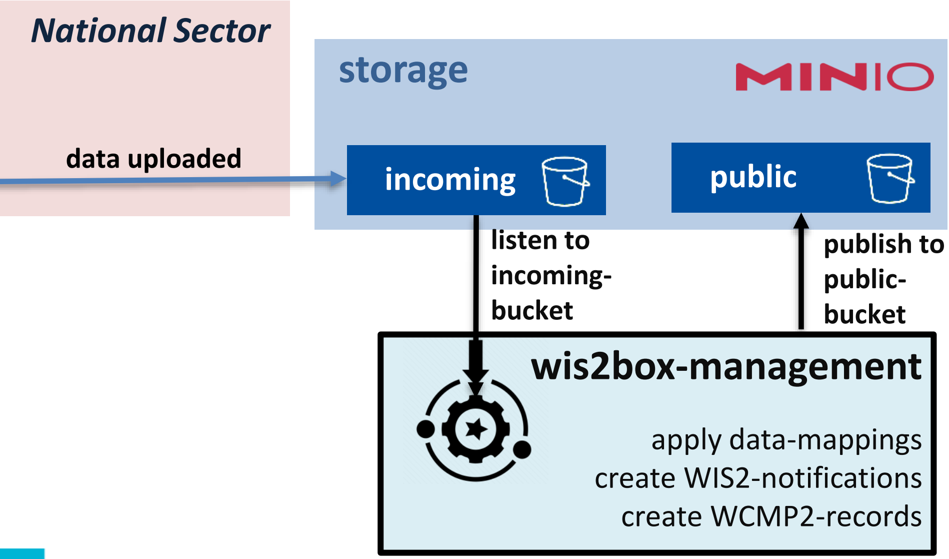
Validate against [CAP v1.2 schema]
and verify digital signature
extract data-timestamp from content



Plugin Name
Universal data without conversion
BUFR data converted to BUFR
FM-12 data converted to BUFR
CSV data converted to BUFR
BUFR data converted to GeoJSON
CAP messages

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

wis2box data mapping workflow



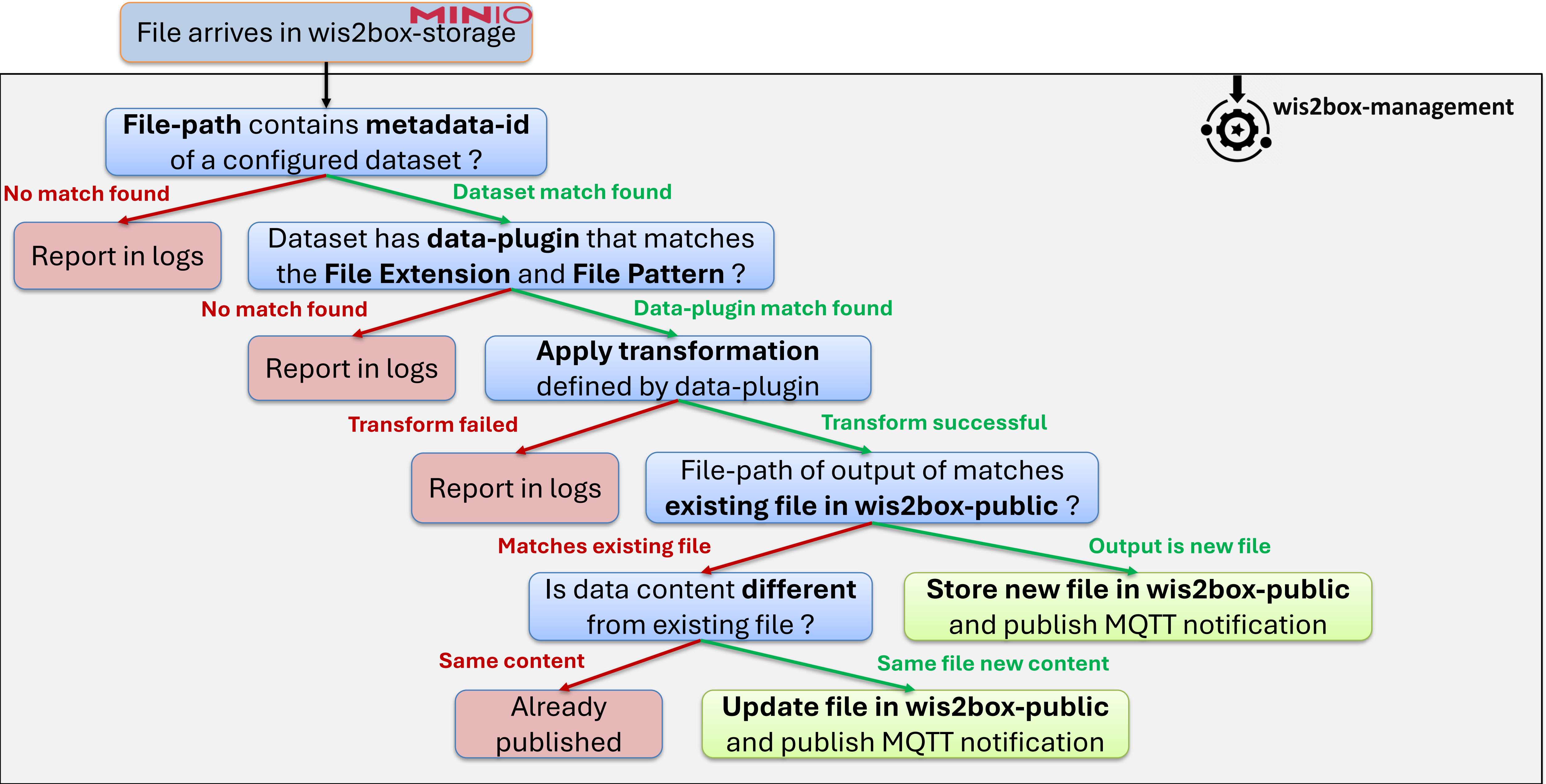
Dataset driven workflow triggered when wis2box-management receives a notification from the storage service

wis2box-management **matches incoming data** with datasets based on **file-path**

data plugins are applied to **transform & publish the data**

WORLD METEOROLOGICAL ORGANIZATION

wis2box data mapping workflow



MINIO

File arrives in wis2box-storage

wis2box-management

File-path contains **metadata-id** of a configured dataset ?

No match found

Report in logs

Dataset match found

Dataset has **data-plugin** that matches the **File Extension** and **File Pattern** ?

No match found

Report in logs

Data-plugin match found

Apply transformation defined by data-plugin

Transform failed

Report in logs

Transform successful

File-path of output of matches **existing file in wis2box-public** ?

Matches existing file

Is data content **different** from existing file ?

Same content

Already published

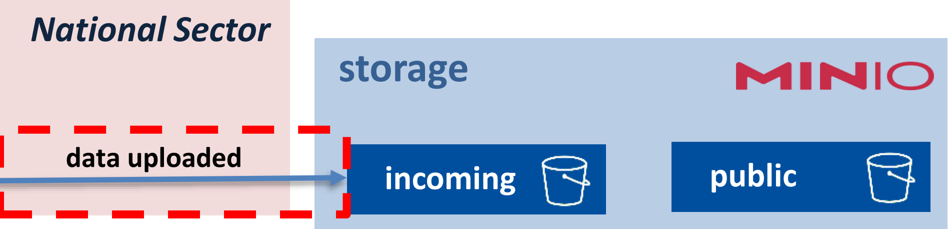
Same file new content

Update file in wis2box-public and publish MQTT notification

Output is new file

Store new file in wis2box-public and publish MQTT notification

How to upload data to wis2box



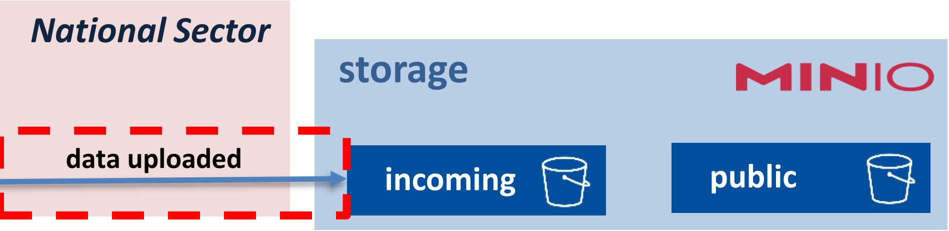
Data can be uploaded to wis2box-incoming in different ways

Optimal solution will depend on systems used in national sector

Options for ingesting data in wis2box:

- **Manual upload using MinIO UI:** manually create path and upload data, useful for testing data-workflow
- **Interfaces in wis2box-webapp:** manual submission using FM-12 form or manual file upload interface
- **Scripts:** write code to upload data using MinIO client software
- **Using SFTP protocol:** forward data to MinIO SFTP-endpoint

How to upload data to wis2box



Data can be uploaded using a minio client library

- Useful to setup custom data workflow:
  - Cronjob running Python script uploading files from local server
  - Query local database to extract CSV and upload result
- https://min.io/docs/minio/linux/developers/minio-drivers.html

Python example to upload a file using the MinIO client module:

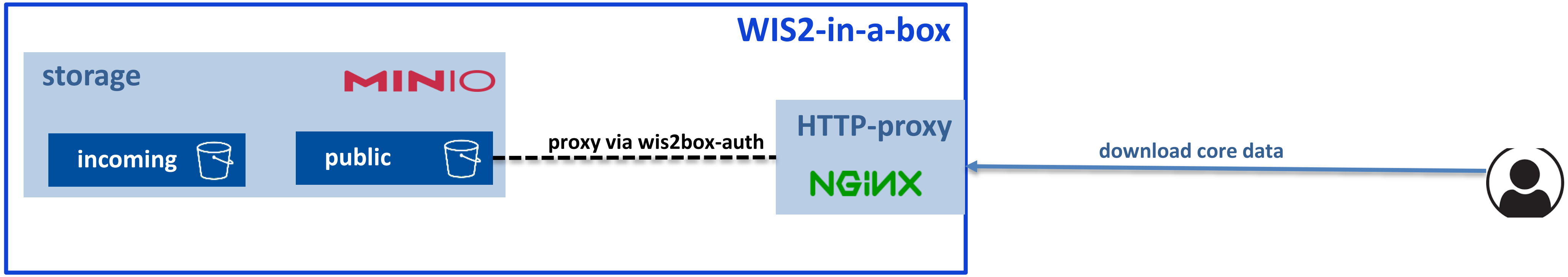
```python
import glob
import sys
from minio import Minio

filepath = '/home/wis2box-user/local-data/mydata.bin'

endpoint = 'http://wis2box-host:9000'
WIS2BOX_STORAGE_USERNAME = 'wis2box'
WIS2BOX_STORAGE_PASSWORD = 'XXXXXXXXX'

client = Minio(
    endpoint=endpoint,
    access_key=WIS2BOX_STORAGE_USERNAME,
    secret_key=WIS2BOX_STORAGE_PASSWORD,
    secure=is_secure=False)

filename = filepath.split('/')[-1]
# path should match the metadata-id for corresponding dataset
minio_path = 'urn:wmo:md:my-center-id:synop-test-dataset'
client.fput_object('wis2box-incoming', minio_path+filename, filepath)
```

WORLD
METEOROLOGICAL
ORGANIZATION

Access control in wis2box

Nginx proxies access to the content of the “wis2box-public”-bucket at **WIS2BOX_URL**/data

All objects in **“wis2box-public” are** available for download unless access control is applied by the wis2box-user



WORLD METEOROLOGICAL ORGANIZATION

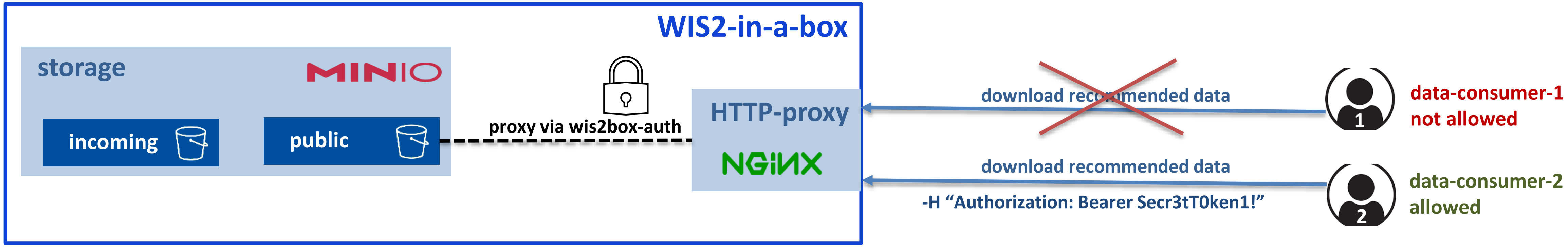
Access control in wis2box

Within wis2box datasets can be made restricted by adding a token using ‘wis2box auth’ commands:

```
wis2box auth add-token --metadata-id urn:wmo:md:mw-mw_met_centre:surface-weather-observations Secr3tT0ken1!
```

To access a restricted dataset Data Consumers should include the token in the HTTP-request-headers:

```
curl -H "Authorization: Bearer Secr3tT0ken1!" "<canonical-url>"
```



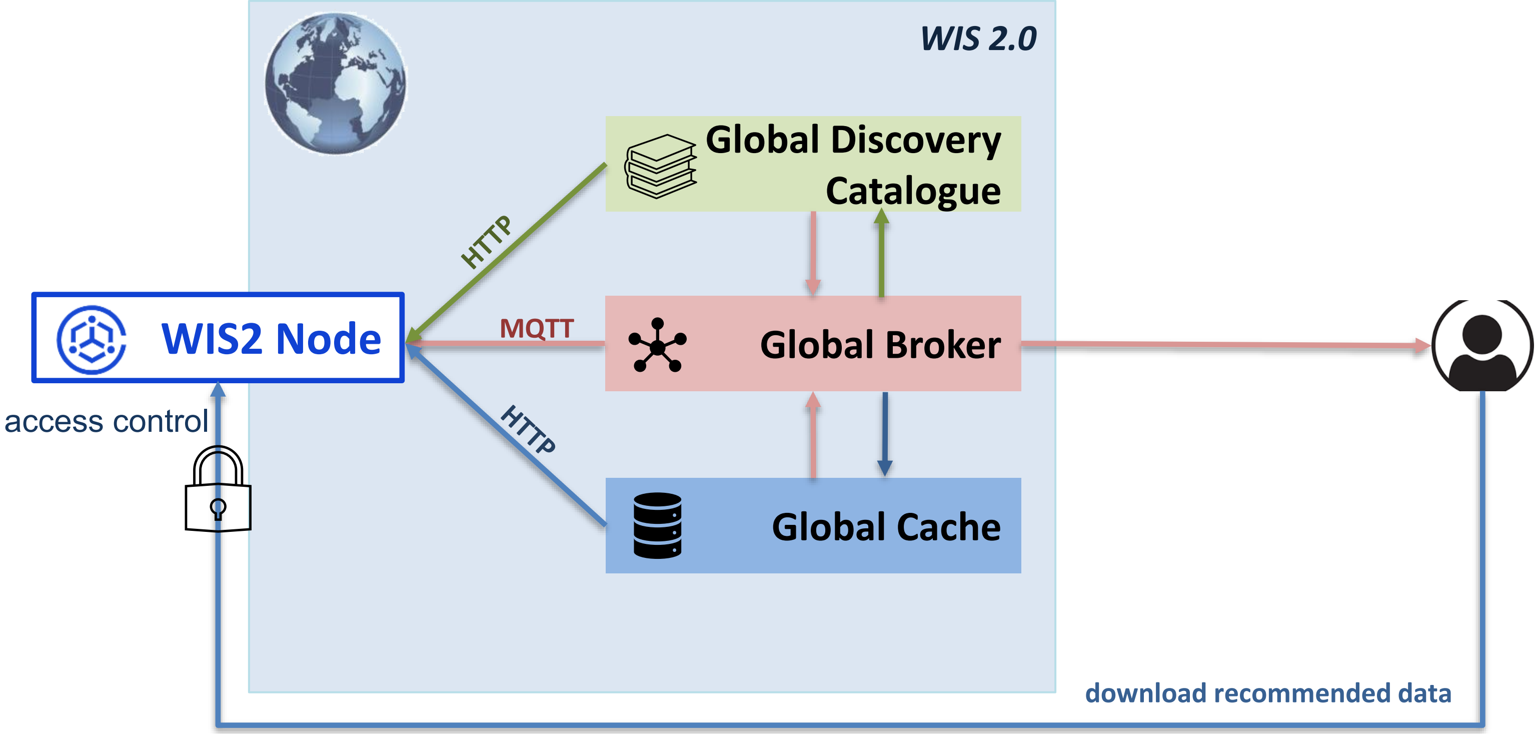
Datasets published on ***origin/a/wis2/…/data/core/…*** should ***never*** **be restricted**

Datasets published on ***origin/a/wis2/…/data/recommended/…*** ***can*** ***be restricted*** to limit data sharing to specific data consumers

Access control in wis2box

Recommended data in WIS2 **MUST include a license URL** in the metadata record
wis2box-instance can host a license-file if required (upload license to wis2box-public)

Recommended data in WIS2 **MAY be access controlled**
wis2box provides the option to add access-control using an authentication token



WORLD METEOROLOGICAL ORGANIZATION

Adding GTS properties using wis2box

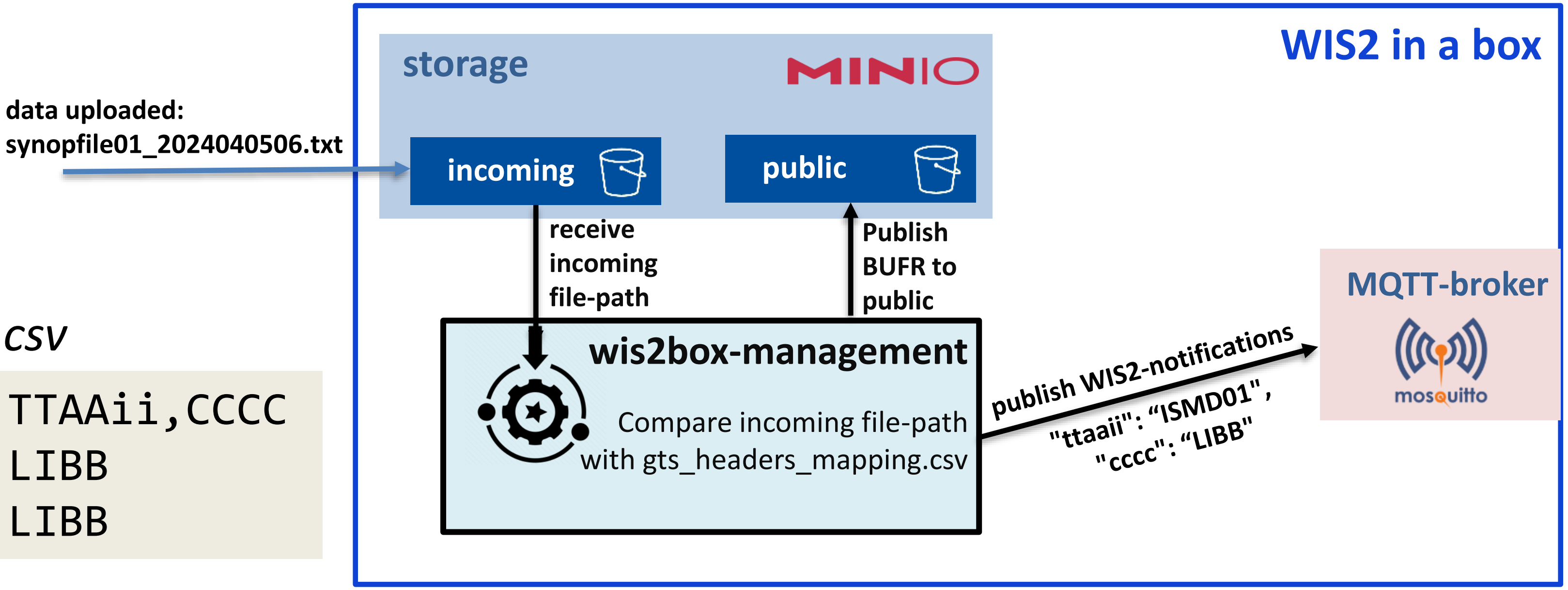
wis2box can use a CSV file to map GTS properties to incoming filenames

The CSV file should be named ***gts_headers_mapping.csv*** and placed in the directory defined by the WIS2BOX_HOST_DATADIR environment variable

https://training.wis2box.wis.wmo.int/practical-sessions/adding-gts-headers-to-wis2-notifications/

gts_headers_mapping.csv

```
string_in_filepath,TTAAii,CCCC
synopfile01,ISMD01,LIBB
synopfile02,ISMD02,LIBB
```



wis2box will <u>NOT</u> add GTS headers to the WIS2 Notification Message if:

- gts_headers_mapping.csv file is not present in **WIS2BOX_HOST_DATADIR** environment variable
- gts_headers_mapping.csv file is present but the incoming filepath did not contain any of ‘string_in_filepath’ specified

Summary

wis2box is a Free and Open Source Reference Implementation of a WIS2 Node

- **WIS2 Node core functions + data-conversions tools to adhere to WMO data standards**
- Source code: https://github.com/World-Meteorological-Organization/wis2box
- Developers can freely use wis2box-components to adapt existing systems to be WIS 2.0 compliant
- The wis2box is maintained and supported by WMO Secretariat
- **Feedback by the community is appreciated to help improve wis2box**

wis2box is software, hosting requirements similar to a public web-service (use cloud or DMZ)

- To function as a WIS2 Node, the wis2box-host needs HTTP and MQTT traffic routed to public IP
- Minimum system requirements: 2 vCPUs, 4GB Memory, 24GB storage
- Requires Python, Docker and Docker Compose

On-line resources

- Documentation: https://docs.wis2box.wis.wmo.int
- Training: https://training.wis2box.wis.wmo.int
- WIS2-in-a-box Newsletter: https://community.wmo.int/media/news/wis2box-newsletter-2026-no1

Thank you

wmo.int

WORLD
METEOROLOGICAL
ORGANIZATION

1950-2025
SCIENCE *for* ACTION