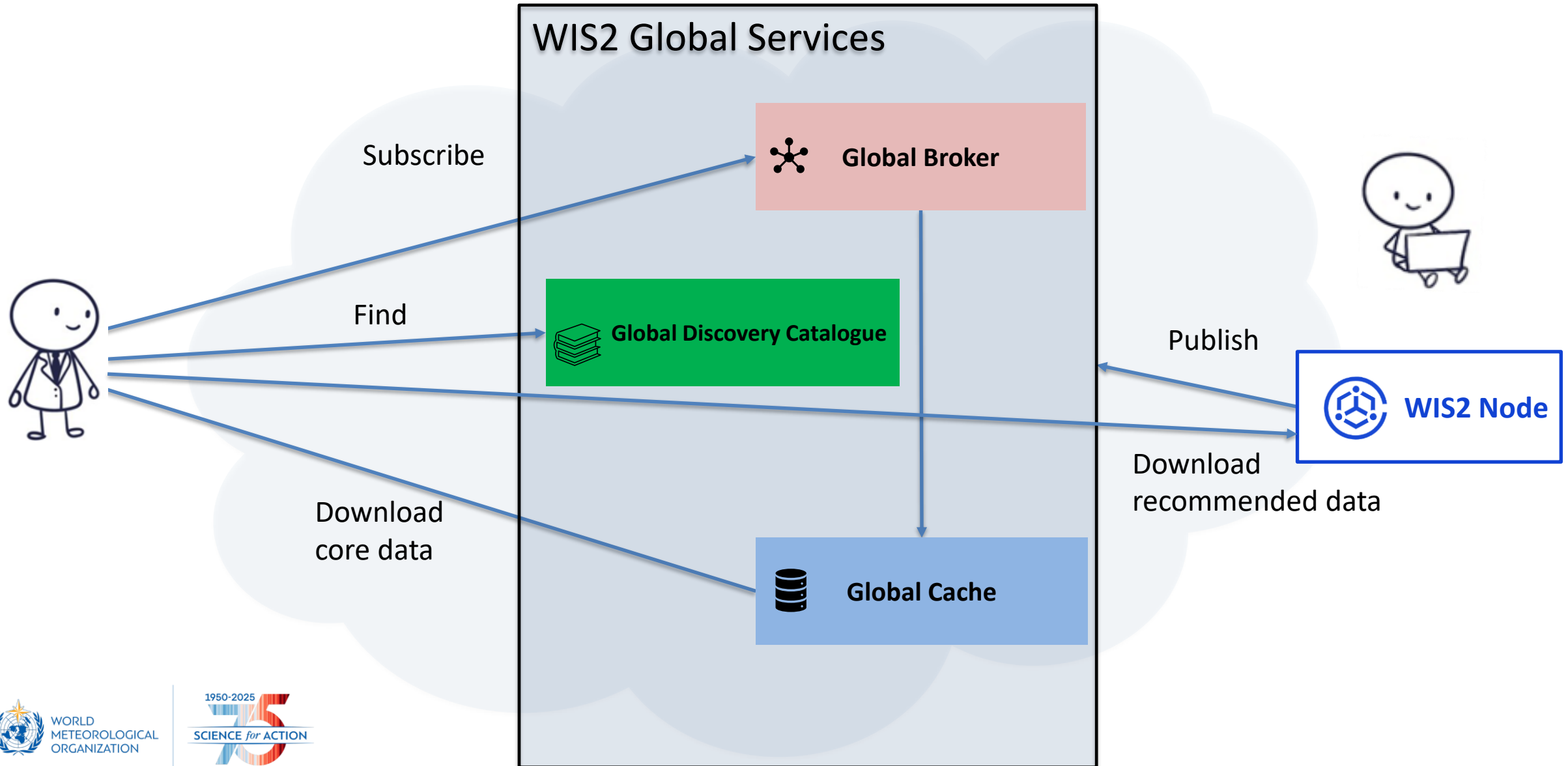


Finding and downloading data as a WIS2 data consumer

Publish, Find, Subscribe, Download

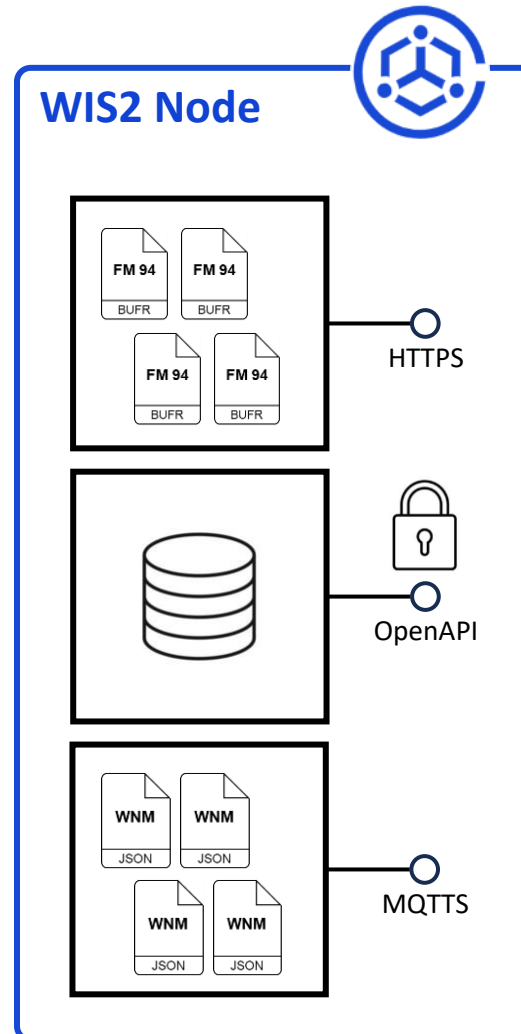


Sharing metadata in WIS 2.0

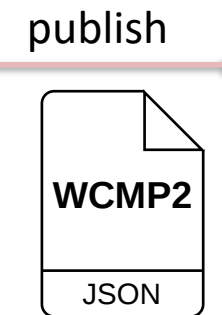
Ahh, good point! The next section in the WIS2 Guide⁵ says that I need to provide 'discovery metadata'⁶ about my Dataset



Description of the Dataset
Identifier
Title
Description
Keywords
Geometry (extent)
Time (extent)
Who to contact
Publisher contact
How to access
Data access (files)
Data access (API)
Data access (notifications)
Conditions of use
Data policy
Rights
License
How to attribute
Citation



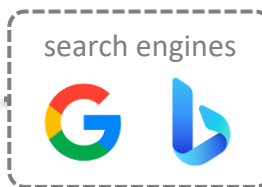
That's great. I can find the description of your Dataset. I can understand **what it contains** and **how to access** it. The **conditions of use** are clear, and I see how I can **attribute** you as the source of the data !



publish



index



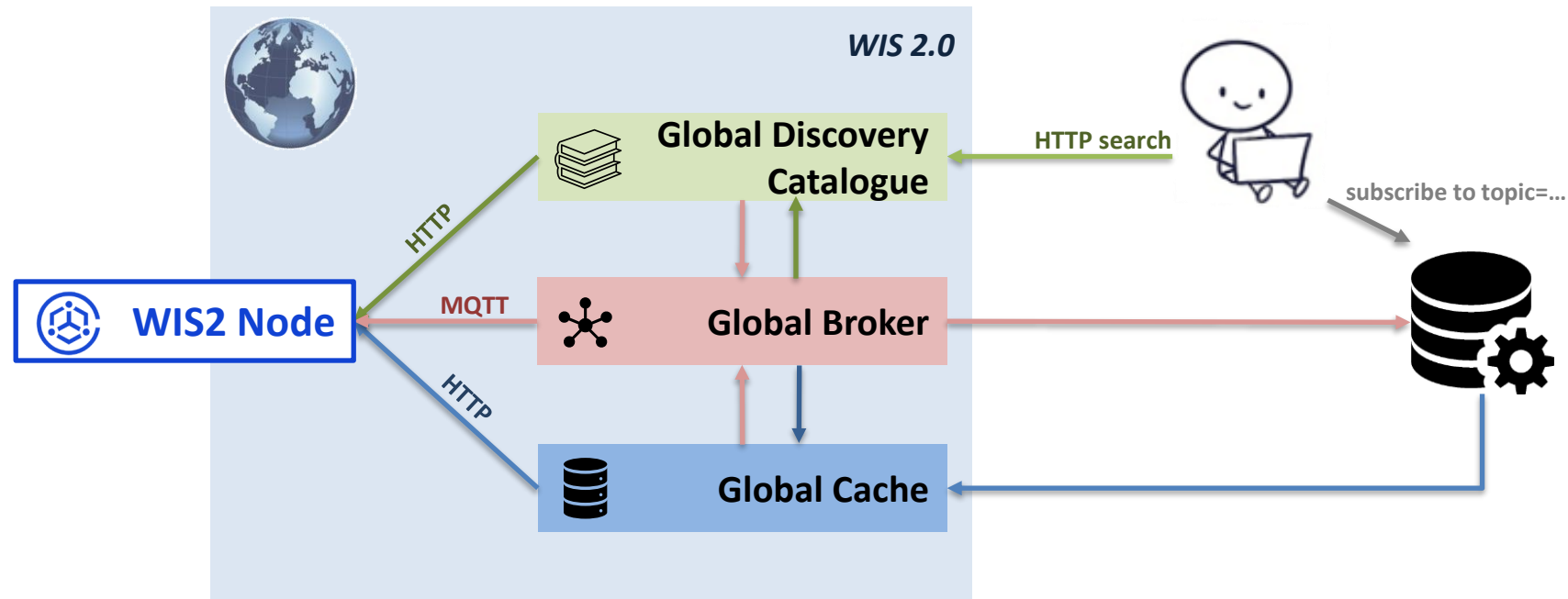
⁵ Guide to WIS (WMO No. 1061), Vol II, §1.3.2 [How to provide discovery metadata to WIS2](#)

⁶ Manual on WIS (WMO No. 1060), Vol II, Appendix F: [WMO Core Metadata Profile 2](#)

Finding data on WIS2

The Global Discovery Catalogue (GDC) provides cataloguing and discovery capability for WIS2 datasets

- Web-based API facilitating search/browse data published via WIS 2.0 (OGC API - Records)
- Harvests WIS2 discovery metadata from WIS2 Nodes (WCMP2 / OGC API - Records)
- Yellow pages (discovery metadata) gateway into WIS2 data and services
- Provides indexing capability to mass market search engines
- Provides quality assessment services of discovery metadata in support of continuous improvement in alignment with WIS2 metadata Key Performance Indicators (KPIs)



Searchable Catalogue

Predicate	Example(s)
Spatial	<code>bbox=minx,miny,maxx,maxy</code>
Temporal	• <code>datetime=t1</code> • <code>datetime=t1/t2</code> • <code>datetime=t1/..</code>
Freetext	<code>q=air+temperature</code>
Attributes	• <code>title=air+temperature</code> • <code>contacts.address.country=Oman</code>
Sorting/paging	• <code>sortBy=title&limit=100&startindex=11</code> • <code>"prev" / "next" link relations</code>
Filter returnable properties	• <code>properties=title,description</code>

Find data with pywiscat

pywiscat provides a Pythonic API atop the WMO WIS2 Catalogue in support of reporting and analysis of the WIS2 Catalogue and its associated discovery metadata

github.com/wmo-im/pywiscat

```
# fetch version
pywiscat --version
```

```
## WIS2 workflows
```

```
# search the WIS2 Global Discovery Catalogue (GDC)
pywiscat search
```

```
# search the WIS2 Global Discovery Catalogue (GDC) with a full
pywiscat search --query radar
```

```
# search the WIS2 Global Discovery Catalogue (GDC) for only recommended
pywiscat search --data-policy recommended
```

```
# search the WIS2 Global Discovery Catalogue (GDC) with a bounding box query
pywiscat search --bbox -142,42,-52,84
```

```
# get more information about a WIS2 GDC record
pywiscat get urn:x-wmo:md:can:eccc-msc:c7c9d726-c48a-49e3-98ab-78a1ab87cda8
```

```
## WIS2 workflows
```

```
from pywiscat.wis2.catalogue import search, get
```

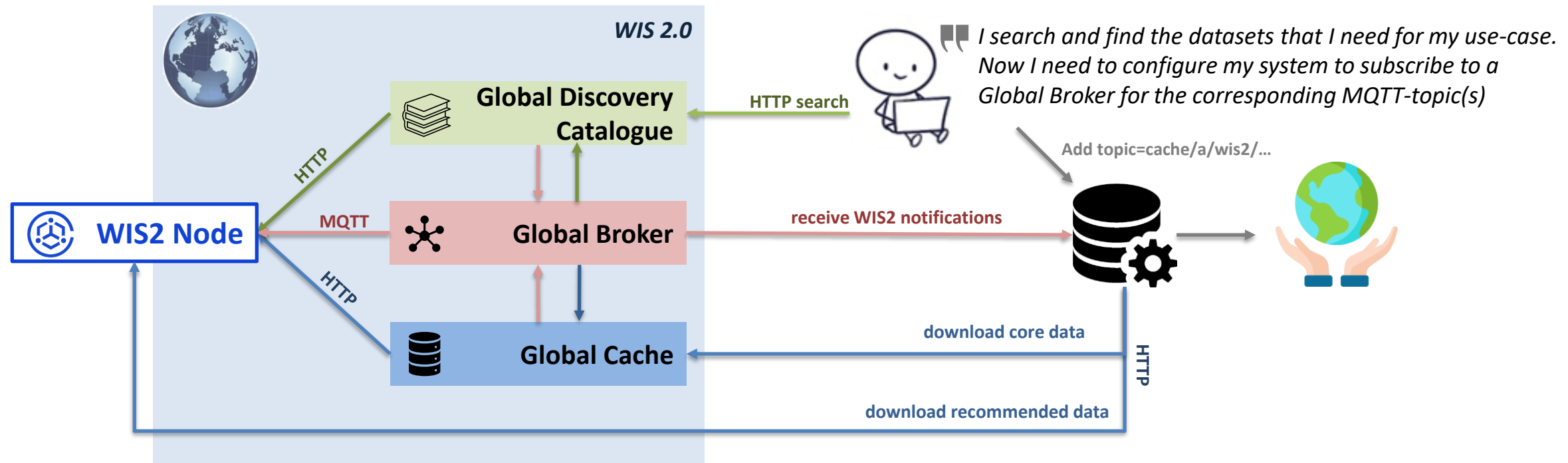
```
# search catalogue
results = search(q='radar', bbox=[-142, 42, -52, 84]))
```

```
# get a single catalogue record
results = get('urn:x-wmo:md:can:eccc-msc:c7c9d726-c48a-49e3-98ab-78a1ab87cda8')
```

Processing WIS2 notifications as a Data Consumer

Data Consumers use MQTT client-software to subscribe to specific channels

WIS2 Notifications indicate the HTTP endpoint where the data can be downloaded from



Downloading data from WIS2



- **WIS2 Notifications provide a canonical link to data**
- **The canonical link is a “one click” link to access data to your laptop/desktop/system**
- **When you subscribe to data notifications, use the canonical link to download the data**

```
{  
  "href": "https://example.org/data-file.bufr",  
  "rel": "canonical",  
  "type": "application/bufr"  
}
```

wis2downloader

wis2downloader is Free and Open Source software developed by WMO that enables subscribe/download functionality for WIS2



wis2downloader:

- Flask-based Python application including MQTT-client
- API to manage subscriptions on MQTT-topics
- Multiple download workers for more efficient data downloading
- Provides statistics via /metrics endpoint for Prometheus



Flask



wis2downloader be installed independently from wis2box: `pip3 install wis2downloader`

pypi.org/project/wis2downloader

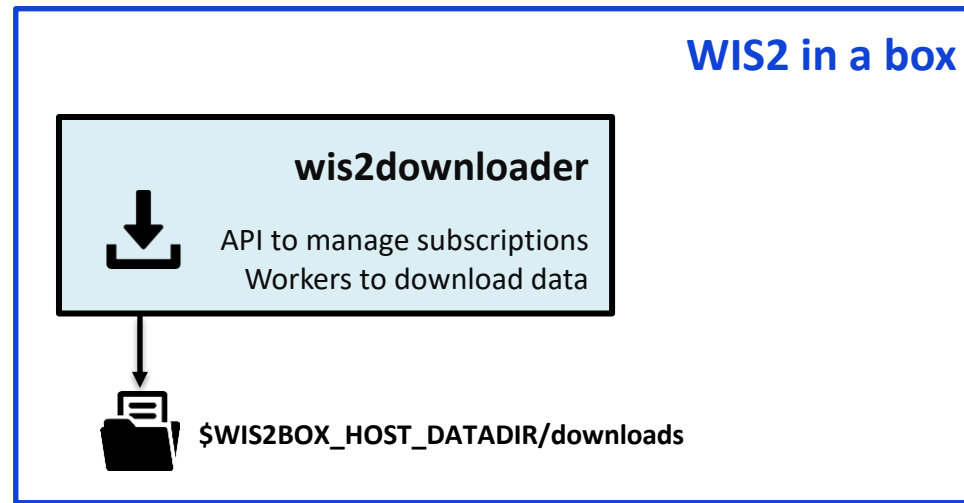
Source code:

github.com/wmo-im/wis2downloader

wis2downloader in the wis2box

Docker container running *wis2downloader* within *wis2box*

- data is downloaded in `$WIS2BOX_HOST_DATADIR/downloads`
- additional *wis2downloader* configuration using *wis2box.env*



Login to *wis2downloader* to manage subscriptions using CLI

```
python3 wis2box-ctl.py login wis2downloader
```

Commands available in *wis2downloader*-container:

```
wis2downloader list-subscriptions
```

```
wis2downloader add-subscription --topic <topic>
```

```
wis2downloader remove-subscription --topic <topic>
```

wis2downloader dashboard in the wis2box

The monitoring stack in wis2box collects and stores wis2downloader metrics in Prometheus and provides a dashboard for viewing information about the data downloaded:



Custom MQTT client based workflow



Data Consumers may prefer to create a custom MQTT-client based workflow to:

- **Integrate** WIS2 subscriber within **existing data processing system**
- **Parse and process data directly**, without writing to file-system
- **Apply authentication** for restricted datasets (data-policy=recommended)

MQTT clients are available for different software environments

One widely used client for Python is paho-mqtt

Custom MQTT client scripting



Example Python code using paho-mqtt

```
import json
import paho.mqtt.client as mqtt_client

def on_message(client, userdata, message):
    j = json.load(message.payload.decode('utf-8'))
    for link in j['links']:
        if link['rel'] == 'canonical':
            client_specific_function(link['href'])

client = mqtt_client.Client('MyDataClientID')
client.username_pw_set('everyone', 'everyone')
client.tls_set(tls_version=2)
client.connect('globalbroker.meteo.fr', port=8883)
client.on_message = on_message
client.subscribe('origin/a/wis2/int-ecmwf/data/core/weather/prediction/forecast/cyclone_tracks')
```

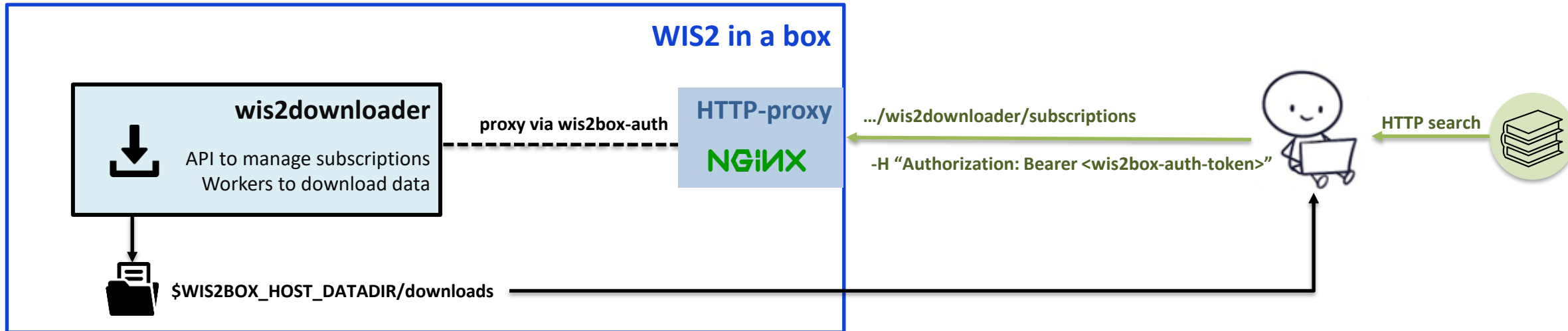
Data Consumer can customize their "on_message" code to define the actions specific for their data ingestion

wis2downloader in the wis2box

Managing subscriptions outside wis2box:

- wis2downloader API is proxied at \$WIS2BOX_URL/wis2downloader
- web-proxy in wis2box secures the API using an authentication token in the HTTP request header

```
curl http://mlimper.wis2.training/wis2-downloader/list -H "Authorization: Bearer <wis2box-auth-token>"
```



Thank you

شكراً لك

wmo.int



WORLD
METEOROLOGICAL
ORGANIZATION

