🏗️ Bronze Notebook - For Assignment 3

This notebook collects and preprocesses ERA5 reanalysis data for Norwegian electricity price areas using the Open-Meteo API. It forms the **bronze layer** of the project's data pipeline.

- Define five Norwegian price areas (NO1–NO5) with city coordinates.
- Retrieve ERA5 weather data (temperature, precipitation, wind) via API.
- Store data in a structured `Pandas DataFrame` and save to CSV.

Step 1 – Define Price Areas

We create a `Pandas DataFrame` containing five Norwegian electricity price areas: Oslo (NO1), Kristiansand (NO2), Trondheim (NO3), Tromsø (NO4), and Bergen (NO5). Each city is represented by its latitude and longitude.

In [31]:
```
import pandas as pd
import requests
from pathlib import Path
```

In [32]:
```
# Here we set the price areas with their latitude and longitude
price_areas = [
    {"price_area": "NO1", "city": "Oslo",         "latitude": 59.9139, "longitude":
    {"price_area": "NO2", "city": "Kristiansand", "latitude": 58.1467, "longitude":
    {"price_area": "NO3", "city": "Trondheim",    "latitude": 63.4305, "longitude":
    {"price_area": "NO4", "city": "Tromsø",       "latitude": 69.6492, "longitude":
    {"price_area": "NO5", "city": "Bergen",       "latitude": 60.3913, "longitude":
]

df_price_areas = pd.DataFrame(price_areas)[["price_area", "city", "longitude", "lat
df_price_areas
```

Out[32]:

	price_area	city	longitude	latitude
0	NO1	Oslo	10.7522	59.9139
1	NO2	Kristiansand	7.9956	58.1467
2	NO3	Trondheim	10.3951	63.4305
3	NO4	Tromsø	18.9553	69.6492
4	NO5	Bergen	5.3221	60.3913

Step 2 – Connect to the Open-Meteo API

We build a function `open_meteo_era5_download()` to fetch hourly ERA5 reanalysis data from the Open-Meteo API endpoint: `https://archive-api.open-meteo.com/v1/era5`. The function takes longitude, latitude, and year as inputs and returns a DataFrame.

In [33]:

```python
# We start with fetching data from the Open-Meteo API
ARCHIVE_BASE_URL = "https://archive-api.open-meteo.com/v1/era5"

# Then we set the Properties matching the CSV columns
HOURLY_VARS = [
    "temperature_2m",       # °C
    "precipitation",        # mm
    "windspeed_10m",        # m/s
    "windgusts_10m",        # m/s
    "winddirection_10m",    # °
]

# Now we create a function to download the data
def open_meteo_era5_download(longitude: float, latitude: float, year: int, timezone
    params = {
        "latitude": latitude,
        "longitude": longitude,
        "start_date": f"{year}-01-01",
        "end_date": f"{year}-12-31",
        "hourly": ",".join(HOURLY_VARS),
        "timezone": timezone,
    }
    r = requests.get(ARCHIVE_BASE_URL, params=params, timeout=60)
    r.raise_for_status()
    data = r.json()

    # Check if 'hourly' data is present
    hourly = data.get("hourly", {})
    if not hourly:
        raise ValueError("No 'hourly' data returned. Check variables or coordinates

    df = pd.DataFrame(hourly)
```

```
    df["time"] = pd.to_datetime(df["time"])
    df = df.set_index("time").sort_index()
    return df
```

Step 3 – Download ERA5 Data for Bergen (2019)

Using the API function, we download hourly data for Bergen (**NO5**) for the year 2019. The dataset includes temperature, precipitation, wind speed, and direction. The result is stored in `df_bergen_2019`.

In [34]:
```
# we can now use the function to download data for Bergen in 2019
bergen_lat, bergen_lon = 60.3913, 5.3221

df_bergen_2019 = open_meteo_era5_download(longitude=bergen_lon, latitude=bergen_lat
df_bergen_2019.head()
```

Out[34]:

time	temperature_2m	precipitation	windspeed_10m	windgusts_10m	winddirection_10
2019-01-01 00:00:00	5.7	0.7	37.0	99.7	2
2019-01-01 01:00:00	5.8	0.2	41.0	107.3	2
2019-01-01 02:00:00	6.1	0.7	42.0	112.0	2
2019-01-01 03:00:00	6.3	0.5	40.9	105.8	2
2019-01-01 04:00:00	5.8	1.1	41.2	110.2	3

Step 4 – Store Data in Bronze Layer

The DataFrame is exported to a CSV file at:
`../Data_Assignment_3/bronze/era5_bergen_60.3913N_5.3221E_2019.csv`.
This dataset is used in the Silver Notebook for outlier detection and trend analysis.

In [35]:
```
outdir = Path("../Data_Assignment_3/bronze")
outdir.mkdir(parents=True, exist_ok=True)

stem = f"era5_bergen_{bergen_lat:.4f}N_{bergen_lon:.4f}E_2019"
df_bergen_2019.to_csv(outdir / f"{stem}.csv", index=True)

outdir.resolve()
```

Out[35]:
```
WindowsPath('C:/NMBU/IND320/Data_Assignment_3/bronze')
```