PureSphere

The Thr'IF Musketeers

OT-7: Foundation of data engineering
Projet 2023 - Presentation

Tom DELAPORTE

Kevin KANAAN

Jorick PEPIN

Presentation plan

1. **PureSphere, what is it?**
   1.1. PureSphere reason to be
   1.2. Data sources
   1.3. Questions
2. **PureSphere overview**
   2.1. Conceptual view
   2.2. Logical view
3. **Ingestion**
4. **Staging**
   4.1. Cleansing
   4.2. Transformation
   4.3. Enrichment
5. **Production**
   5.1. Graph database
   5.2. Star Schema
6. **Further improvements**

1

PureSphere

What is it?

PureSphere reason to be

Have you ever wondered if we can we believe or not the statistics on industrial pollutant emissions?

→ **This is the starting point of PureSphere**

PureSphere is a **data pipeline** designed **to provide sufficient data** to **build trustworthy analysis** regarding the **impact of the industrial sites on their surrounding environments.**

Data sources

GÉORISQUES
Mieux connaître les risques sur le territoire

List of industrial facilities releasing pollutants.

Géorisques

• • •

GEOD'AIR

Reference data and statistics on air quality in France.

Geod'air

• • •

data.gouv.fr

hub'eau

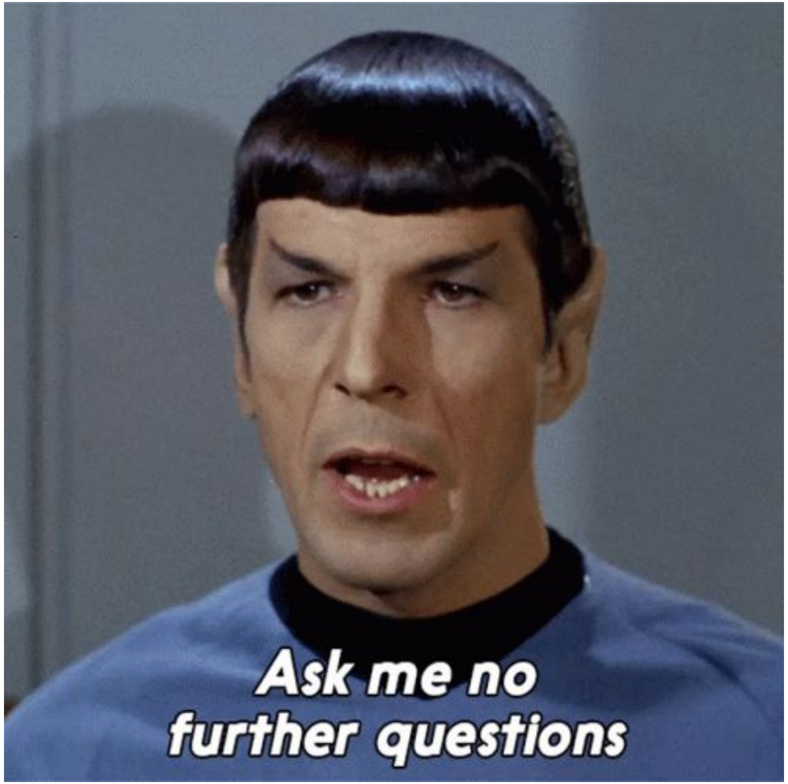
List of physico chemical analysis of water quality.

Hub'eau

• • •

API

Questions



Easy ● ● ●

What are the zones for which we have information about the air quality and the water quality?

Medium ● ● ●

Can we see the impact of industrial sites on their surrounding area in terms of air and water quality?

Hard ● ● ●

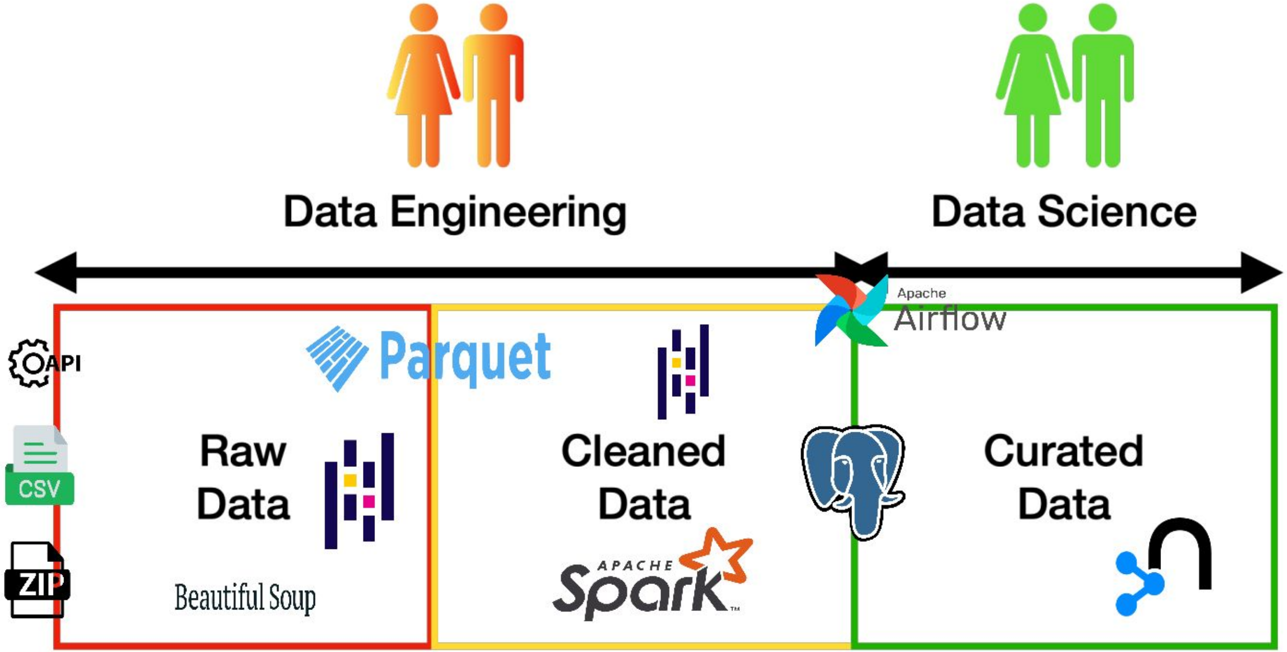
Do the data from the air and water quality sensors coincide with the pollutant discharges given by Géorisques?

2

PureSphere

Overview

Conceptual view



Data Engineering

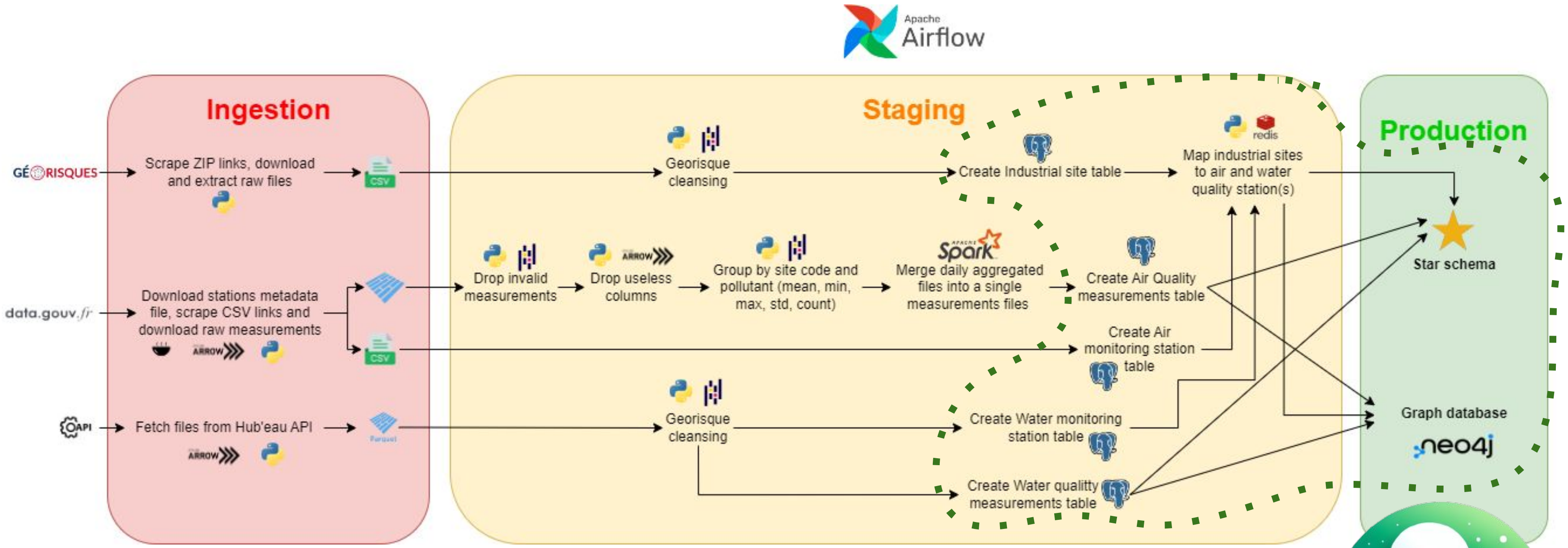
Data Science

Raw Data

Cleaned Data

Curated Data

Logical view



Apache Airflow

Ingestion

GÉORISQUES → Scrape ZIP links, download and extract raw files → CSV

data.gouv.fr → Download stations metadata file, scrape CSV links and download raw measurements → CSV

API → Fetch files from Hub'eau API → Parquet

Staging

Georisque cleansing → Create Industrial site table → Map industrial sites to air and water quality station(s)

Drop invalid measurements → Drop useless columns → Group by site code and pollutant (mean, min, max, std, count) → Merge daily aggregated files into a single measurements files → Create Air Quality measurements table

Create Air monitoring station table

Georisque cleansing → Create Water monitoring station table

Create Water qualitty measurements table

Production

Star schema

Graph database

neo4j

3. Ingestion

Several issues

- **APIs limitations**: ✖ OpenAQ → Limited by the call rate ⁓ Switch to Geod'air
  ✖ Hub'eau → Limited to 20000 rows/response ⁓ Smart requests

- **Data volume**: Geod'air example

≃3.5 GB / year CSV → Parquet ≃66 MB / year

- **Sources aren't directly accessible**: ⁓ Extract link from web pages (Beautiful Soup)

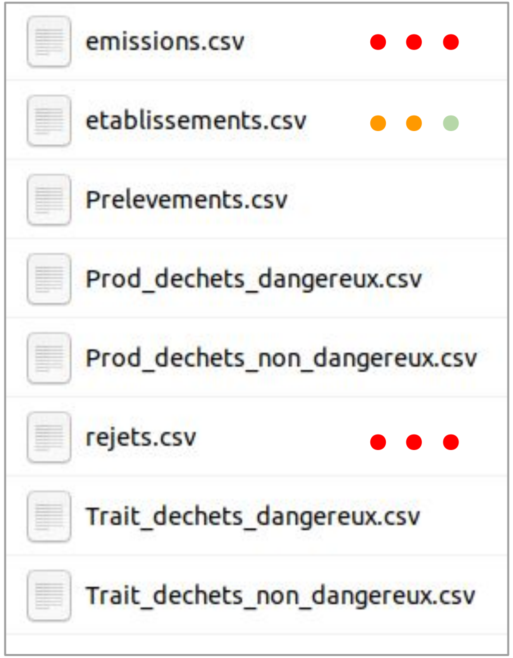
4

PureSphere

Wrangling

Cleansing

I - Georisques

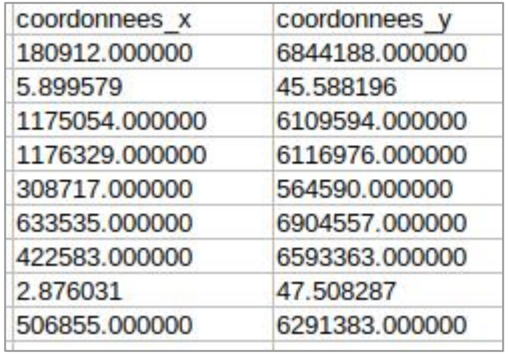


- emissions.csv ● ● ●
- etablissements.csv ● ● ●
- Prelevements.csv
- Prod_dechets_dangereux.csv
- Prod_dechets_non_dangereux.csv
- rejets.csv ● ● ●
- Trait_dechets_dangereux.csv
- Trait_dechets_non_dangereux.csv

- Remove unnecessary data (files and columns)
- Transform literal addresses into GPS coordinates (latitude, longitude)

only 0.8% of valid values

GeoPy

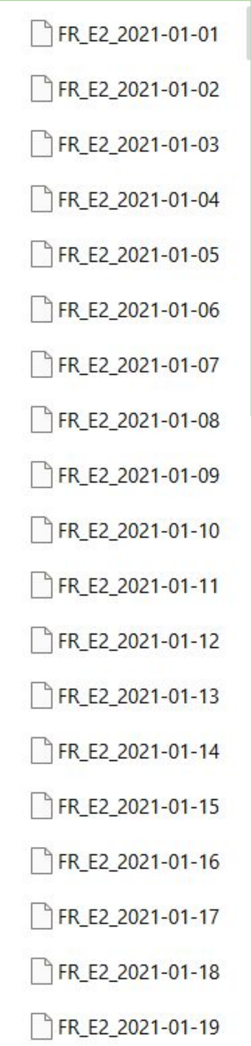


coordonnees_x	coordonnees_y
180912.000000	6844188.000000
5.899579	45.588196
1175054.000000	6109594.000000
1176329.000000	6116976.000000
308717.000000	564590.000000
633535.000000	6904557.000000
422583.000000	6593363.000000
2.876031	47.508287
506855.000000	6291383.000000

Cleansing

II - Geod’air

After running the ingestion pipeline, the landing zone contains a Parquet file per day from the 1st of January 2021 until today. We also have a file containing relevant metadata about air monitoring stations.



FR_E2_2021-01-01
FR_E2_2021-01-02
FR_E2_2021-01-03
FR_E2_2021-01-04
FR_E2_2021-01-05
FR_E2_2021-01-06
FR_E2_2021-01-07
FR_E2_2021-01-08
FR_E2_2021-01-09
FR_E2_2021-01-10
FR_E2_2021-01-11
FR_E2_2021-01-12
FR_E2_2021-01-13
FR_E2_2021-01-14
FR_E2_2021-01-15
FR_E2_2021-01-16
FR_E2_2021-01-17
FR_E2_2021-01-18
FR_E2_2021-01-19

2 majors steps

1. **Remove invalid rows**: Some measurements are tagged as falsy or incoherent, some other have missing measurement. We kept data only coming from trustworthy stations.

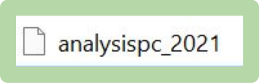
2. **Drop useless columns**: For the sake of our project, we won’t need all the columns contained by Geod’air files.

Cleansing

III - Hub’eau

In the Landing zone we also have a parquet file containing all the data that were measured during a year.

- Removed unnecessary columns
- Removed invalid rows



analysispc_2021

Transformation

Geod’air

A file per day containing hourly averaged pollutant concentrations for all the stations (a single station can monitor more than one pollutant).

↓

For each file, we group by station and pollutant type and aggregate the following way:

- Average pollutant concentrations
- Min and max hourly average concentration recorded
- Standard deviation
- # measurements

↓

Single file resulting of the concatenation of all the daily files augmented with a date column.

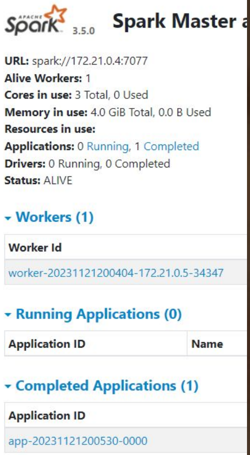
Batch processing

pandas

Transformat

Apache Spar

Single file resu d with a date column.



Spark 3.5.0 Spark Master a

URL: spark://172.21.0.4:7077
Alive Workers: 1
Cores in use: 3 Total, 0 Used
Memory in use: 4.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Resources
worker-20231121200404-172.21.0.5-34347	

Running Applications (0)

Application ID	Name	Duration

Completed Applications (1)

Application ID		Duration
app-20231121200530-0000	D	1.9 min

```
[2023-11-21, 20:05:57 UTC]                         Manager: Finished task 1.0 in st
0.0 (TID 1) in 945 ms on 172.21.0.5 (executor 0) (1/1053)
```

Transformation

Hub’eau

Adding aggregations and calculating the average concentration of a pollutant for a day.

The goal is to have a file containing averaged pollutant concentrations for all the stations. In a way that matches for sure what we have with the air quality.

Batch processing

pandas

Transformation

Map industrial sites to their surrounding air and water quality monitoring stations based on:

- Spatial distance
- Kind of pollutant released by the industrial site VS kind of pollutant monitored by the station
- Any other relevant information we might have access to

The mapping represents heavy computations.

To avoid doing the computation twice, **we might use Redis to cache the mapping.**

Key	Value (List)
Industrial_site_1	[AQ_station_1, … WQ_station_42]

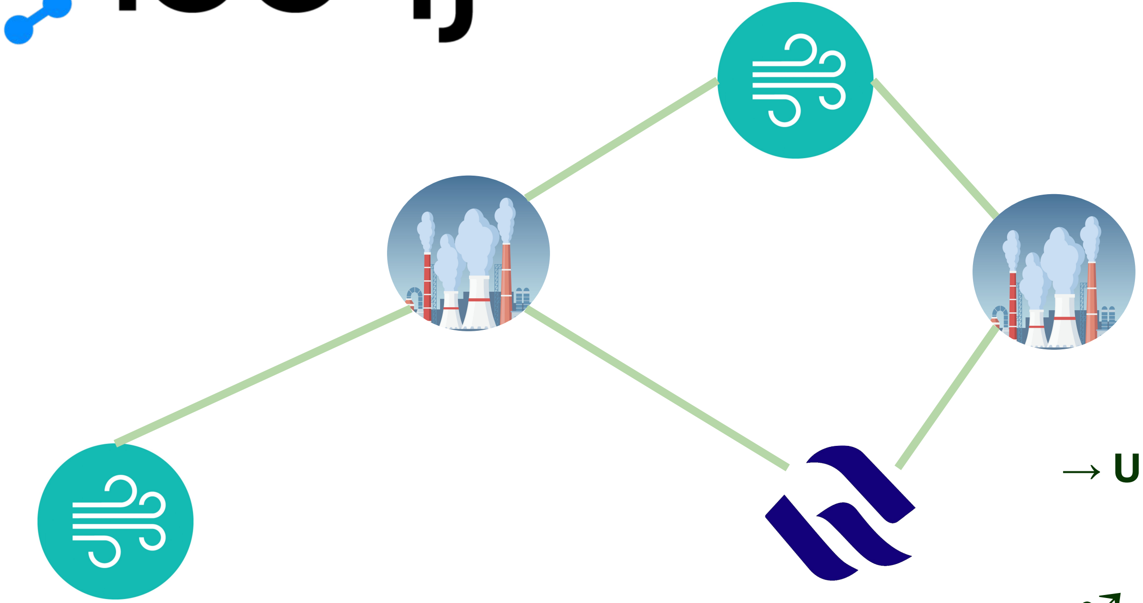
5

PureSphere

Production

Graph database

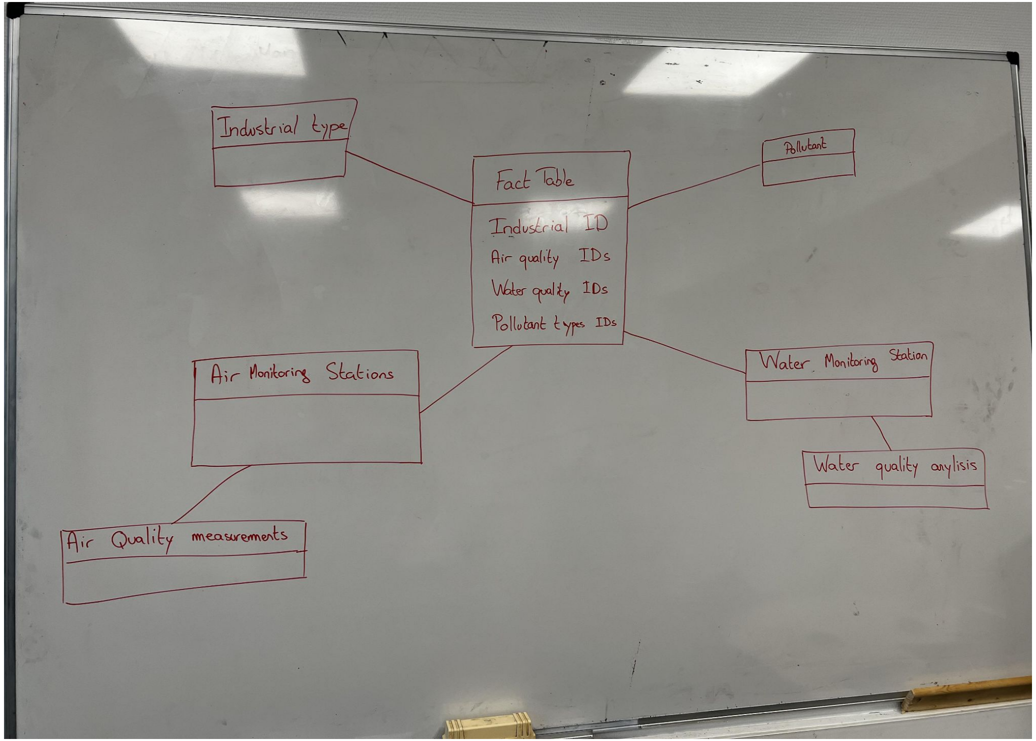
neo4j



→ **Using Neo4J spatial functions**

↝ Hard • • •

Star schema



Easy ● ● ●

Medium ● ● ●

6

PureSphere

Further improvements

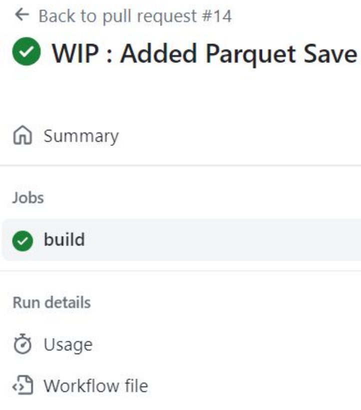
Further improvements

Use Redis for caching

Fine tune Spark to go even faster

Github Codespace

Thr’IF Muske



← Back to pull request #14

WIP : Added Parquet Save

Summary

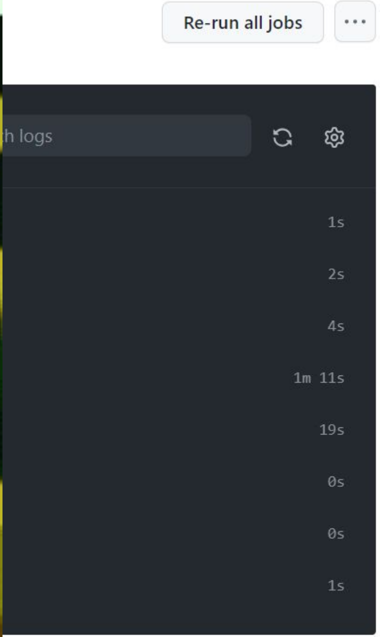
Jobs

build

Run details

Usage

Workflow file



Re-run all jobs ···

h logs

```
1s
2s
4s
1m 11s
19s
0s
0s
1s
```

Automatic PyLint c **the PEP allergic**

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

Thanks!

Do you have any questions?

The Thr'IF Musketeers

Tom DELAPORTE

Kevin KAANAN

Jorick PEPIN