DB Group @ unimo

ECDP: a (Big Data) Energy Communities Data Platform

Sonia Bergamaschi, Luca Gagliardelli

Università degli Studi di Modena e Reggio Emilia, Modena, Italy
{name.surname}@unimore.it

https://www.linkedin.com/company/dbgroup-unimore

Local Energy Communities

- At **COP28**, nations agree to **transition away from fossil fuels** and reach **world net zero carbon emissions** by **2050**
- **Local Energy Communities (LEC)** are already widely **recognized** in the **European Union**, as evidenced by their inclusion in the **FitFor55** files
- LEC put **citizens at the center of the energy transition**
  - LEC can significantly **accelerate the transition from fossil fuels**
  - By **2050**, around **45% of renewable energy production** could be in the hands of **citizens**

The project and the partners

The aim of the project is to **develop a platform to collect and analyze big data about the energy consumption inside Local Energy Communities** (LEC), encouraging a more conscious use of energy by the users and promoting the self-consumption of the energy produced by the LEC.

The project is funded by the Italian Ministry of Economic Development as a part of the 2019-2021 National Electricity System Research Plan.

The project involves three partners:

- ENEA: **commissioned and supervised the project**. It is a public body aimed at research and technological innovation in the sectors of energy.
- DBGroup: **designed the platform**. It is the database research group of the University of Modena and Reggio Emilia, mainly works in big data integration.
- DataRiver: **developed the platform.** It is an innovative SME which also develops innovative software solutions in the field of Big Data Integration & Analytics.

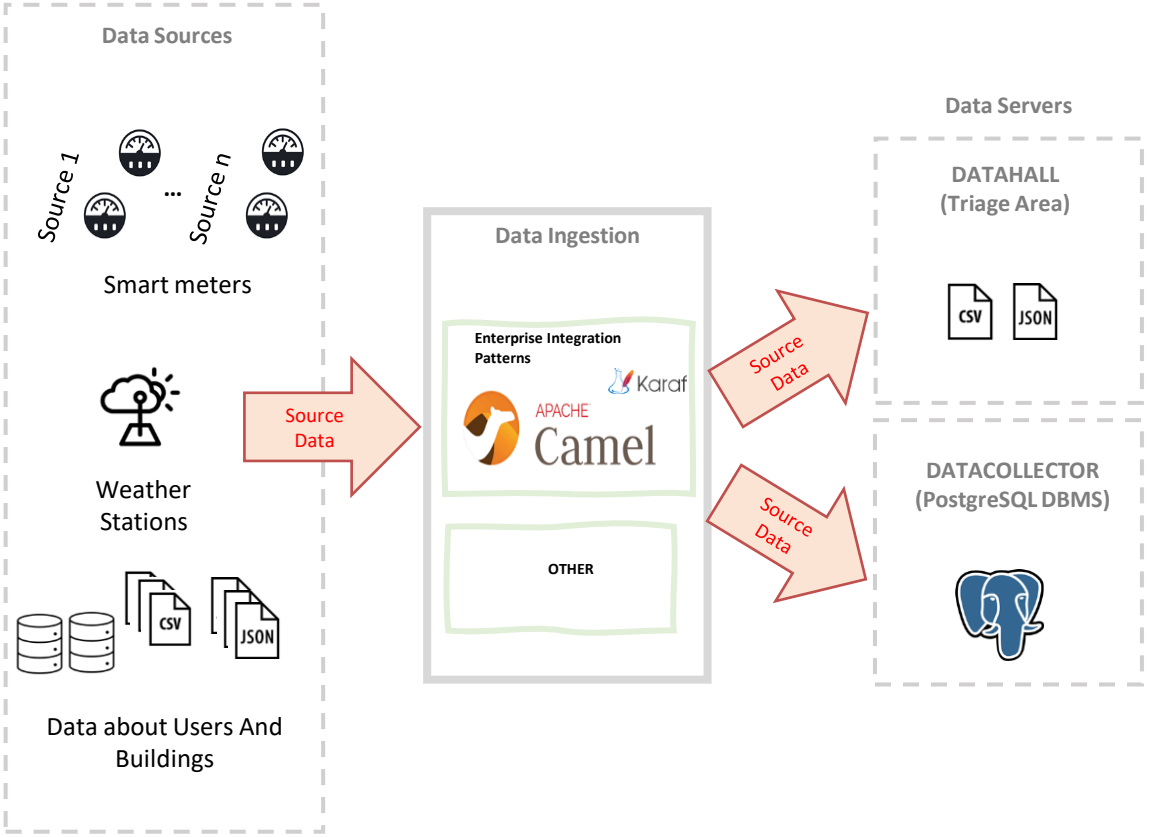
The input data

The development of the platform was guided by several use cases deriving from previous projects developed by ENEA:

- GECO: developed innovative solutions to improve the local energy communities, promoting the *prosumers*, i.e. citizens that produce and consume energy
- SelfUSER: through several sensors placed on a *smart building*, measures the energy consumed by the users and the energy produced by the photovoltaic panels placed on the building, with the aim of studying the load curves to optimize the storage and self-consumption of the produced energy
- PELL: collects information about the devices of the public lighting systems and measures their energy consumption, with the goal to analyze and optimize it.

Pre-existing Data Sources & Ingestion

- To manage the data of the previously presented projects, ENEA developed an **ingestion system**
- By using Apache Camel and Karaf the collected **raw data is stored in the so-called Triage Area**. Mainly, this data is in **JSON** and **CSV** formats.
- **Structured data** (like static information about buildings or users) **is stored in the so-called DataCollector** which is managed by a PostgreSQL DBMS.



Some Data Sources

Project	Data source	Type	Format	Granularity	Fields
SelfUser	EnelX	Unstructured	CSV	Second	Sensor_ID, Local_Time, AVG(CurrentA), AVG(VoltageV), AVG(Pow_activeW), AVG(Pow_reactiveVar)
SelfUser	Weather Sensors	Unstructured	JSON	Minute	ID, timestamp, temperature, humidity
SelfUser	ARERA	Unstructured	CSV	15 minutes	POD, timestamp, energy, reactive_power
SelfUser	SelfUser static data	Structured	PostgreSQL DB	15 minutes*	The database contains 10 tables that describe the users and the buildings of the LEC.
PELL	PELL static data	Structured	MySQL DB	15 minutes*	The database contains 3 tables that describe the devices of the public lighting system (e.g. position and technical details)
PELL	PELL dynamic data	Structured	Parquet	15 minutes	Data are stored in a parquet file

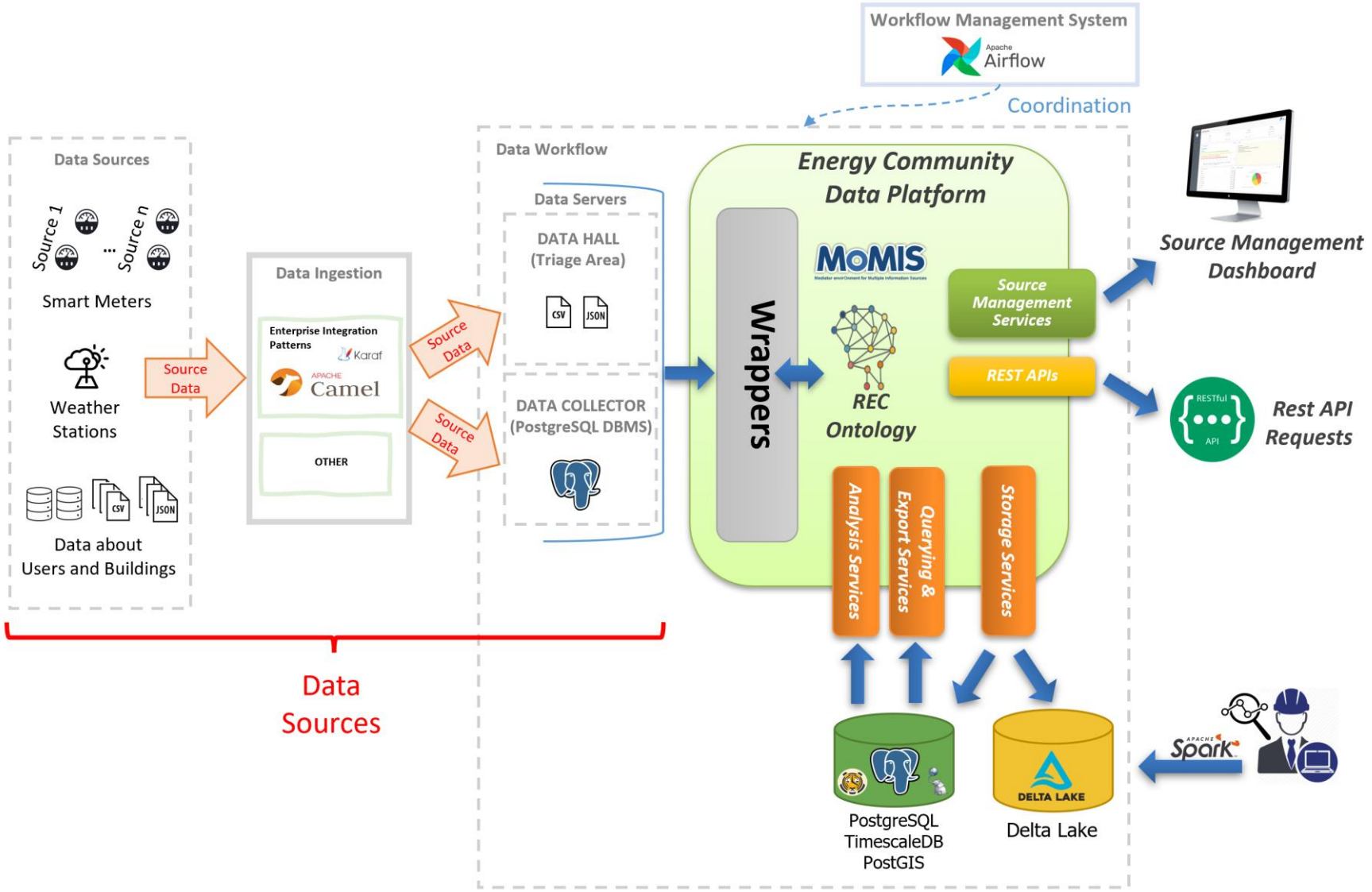
*updates from these databases are checked every 15 minutes, but data might not change so frequently

Main requirements of the ECDP Big Data Platform

The big data platform to be designed requires to:

- **Collect data from many data sources** with different formats, structured and unstructured
- **Perform online and offline** (batch) **analytics** operations
- **Manage time series and geographical data**: consumption data of a POD (Point of Delivery) are associated with a time and geographical coordinates
- **Store all the raw data** (at the original granurality) for further analysis;
- **Integrate data coming from different data sources with different time granularity**: e.g. combining weather information with the energy production of photovoltaic panels
- **Be Scalable & Adopt Open-Source solutions**

ECDP: Energy Community Data Platform

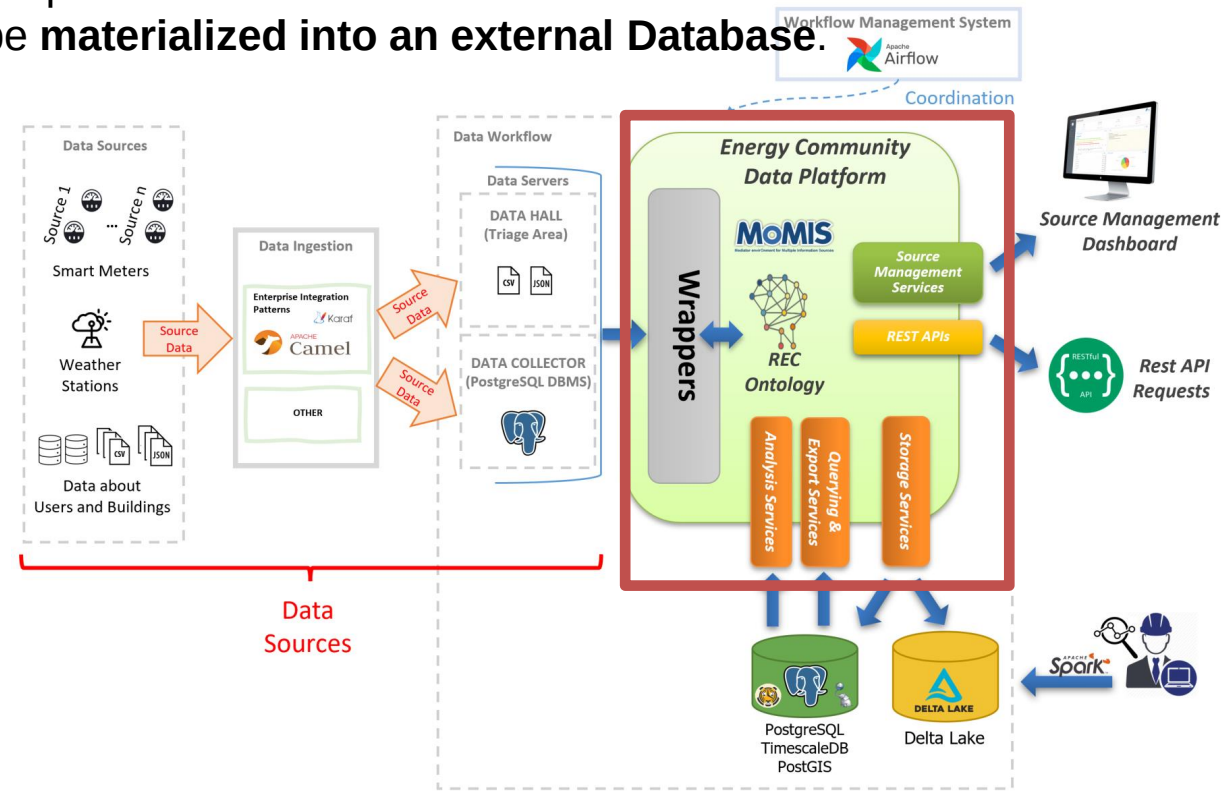


MOMIS

- MOMIS is a **data-integration system** based on a wrapper/mediator architecture:
  - **Wrappers** makes available many data formats transformers
  - **The Mediator** performs data-fusion generating a **Global Integrated Schema**
- A **Global Schema can be queried as a traditional database by using the SQL language** through the query manager or third-party applications
- MOMIS performs the so-called **virtual integration**, i.e. data are retrieved from the sources at query time, guaranteeing that the queries results always contain updated data
- To speed up frequent queries, integrated data can be **materialized into an external Database**.

MOMIS is the core of the platform, it is used to: ingest the data from different data sources, perform data integration, store data in the storage layer, and connect the storage layer with the rest of the world.

We chose to use MOMIS because it is an open-source system currently managed by DataRiver that was designed by the DBGroup and played a central role in its research activities for several years.

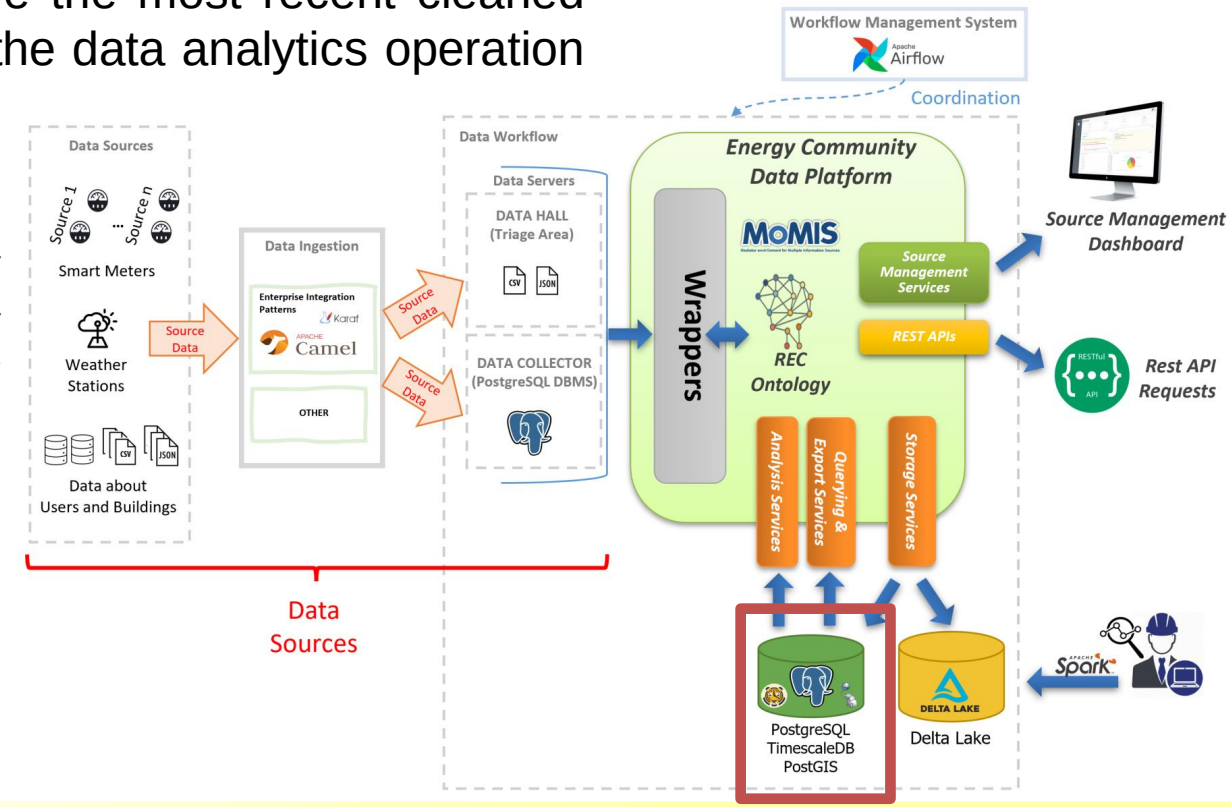


PostgreSQL + TimescaleDB + PostGIS

- **PostgreSQL** is one of the top performing open-source DBMS
- **TimescaleDB** and **PostGIS** are open-source PostgreSQL extensions which allow respectively to manage time series and to perform spatial queries

PostgreSQL with its extensions is used to store the most recent cleaned and integrated portion of the data to speed up the data analytics operation performed through the MOMIS Dashboard.

We chose this combination of tools because the data to be managed are time series: every measurement comes from a Point Of Delivery (POD) and is associated with a time and geographical position. The TimescaleDB extension allows performing aggregation queries based on time, while PostGIS allows performing spatial queries.



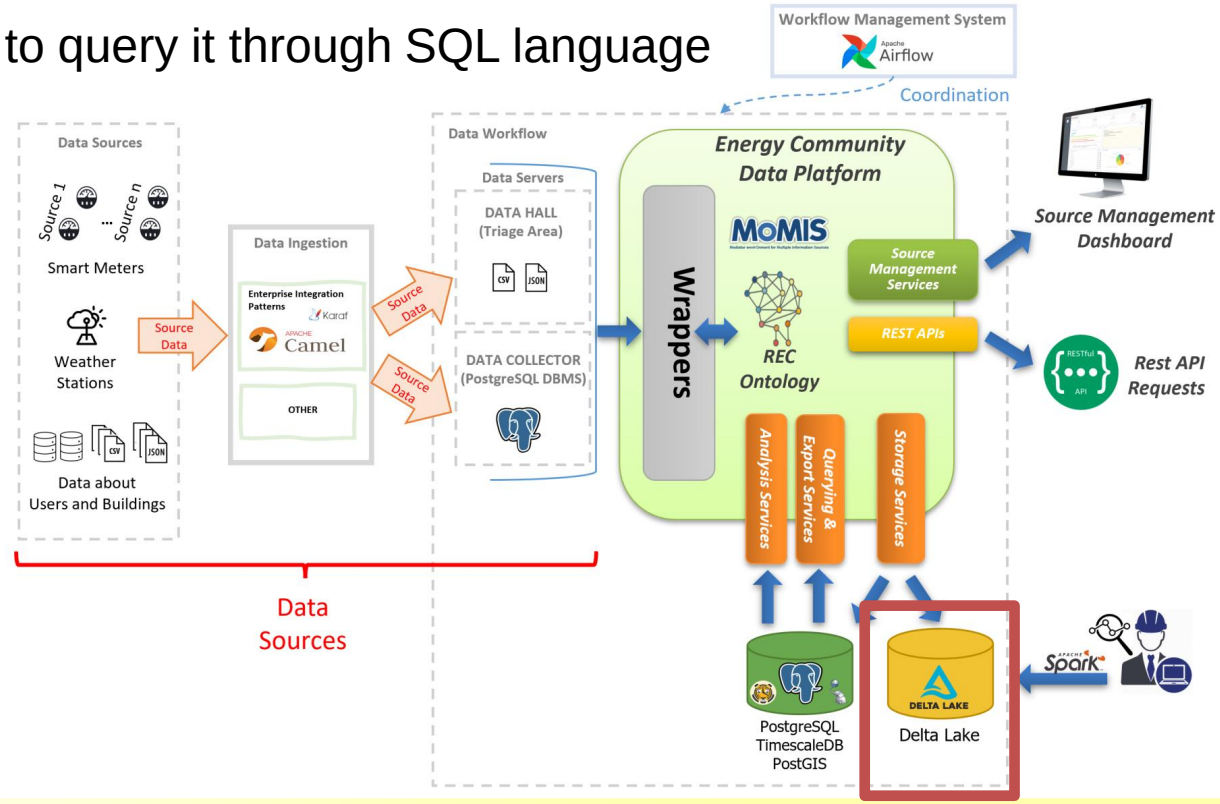
Delta Lake

Delta Lake is an open-source storage framework that provides many interesting features such as:

- ACID transactions support
- Scalability
- Data versioning: provides access to an earlier version of the data
- Log to audit all changes of the data
- Integrated with Apache Spark, that enable to query it through SQL language

In the platform, we use Delta Lake to store all the raw data that can be used for further analysis and to store the older integrated data that does not need to be accessed frequently from the MOMIS Dashboard.

The main reason why we chose Delta Lake is that it can be directly queried by using Apache Spark seamlessly since Apache Spark is already used by ENEA to perform data analysis operations.

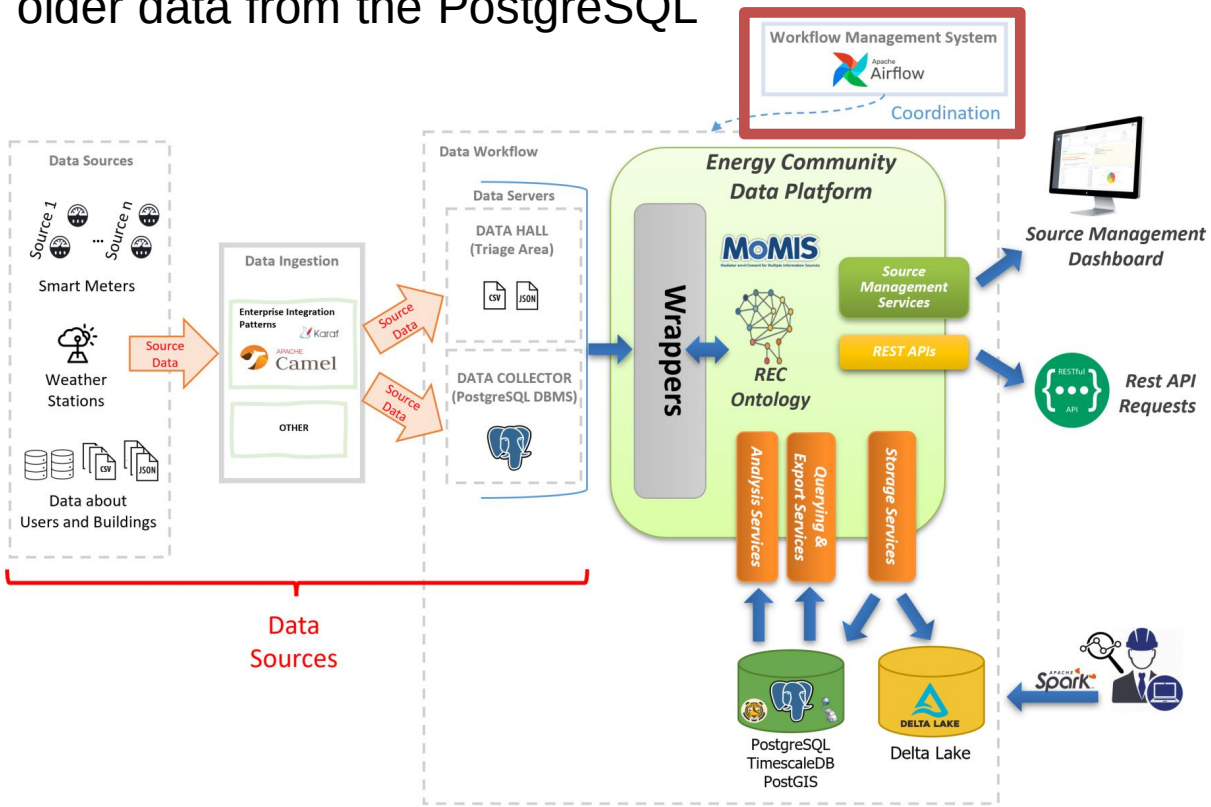


Airflow

Apache Airflow is an open-source platform to programmatically schedule and monitor workflows. Airflow lets to define Python scripts that can be scheduled and automated, moreover, it can manage errors and exceptions.

In the platform, Airflow is used to automate several tasks, such as the import of new data from the data sources, or moving older data from the PostgreSQL database to the Delta Lake database.

We chose Airflow because it is fully compatible with the other employed software components, and it is fully customizable through simple Python scripts. Moreover, Airflow can manage the execution of Spark applications, making it possible to perform operations on the Delta Lake storage framework.

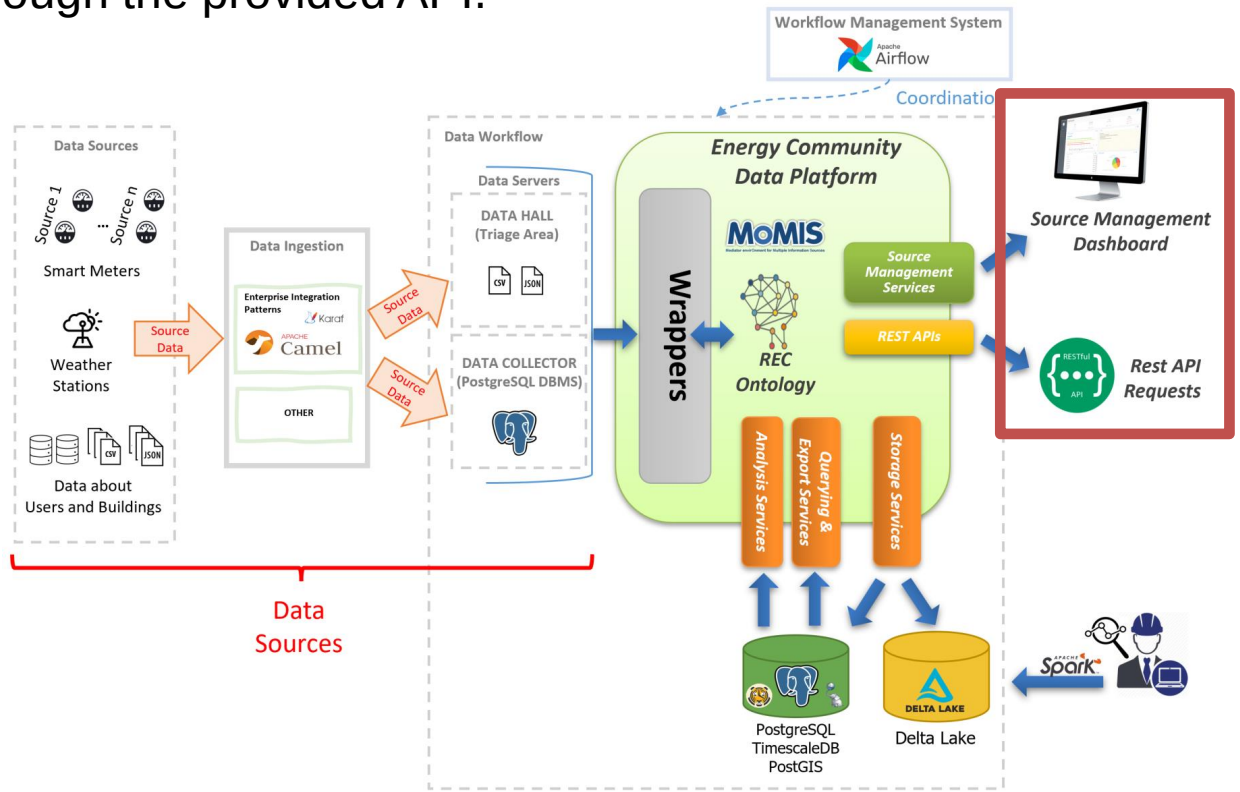


MOMIS Dashboard

The MOMIS Dashboard is a data visualization and analysis tool developed by DataRiver for MOMIS which allows you to perform the analysis of the integrated data through different visualization modes (e.g. bar charts, line charts, tables, etc).

However, other external tools can be used through the provided API.

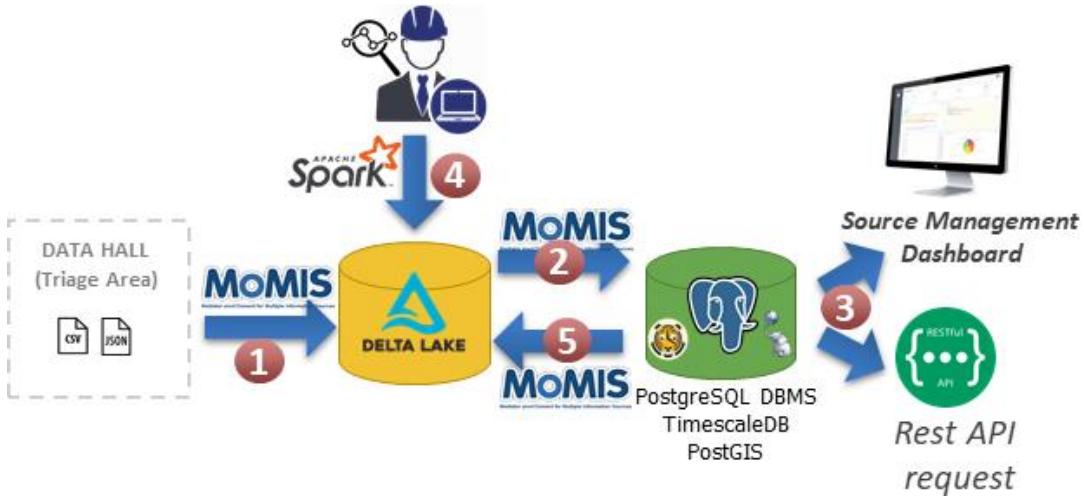
We chose the MOMIS Dashboard as a visualization tool because it is strictly integrated with MOMIS and is provided by DataRiver.
Moreover, this tool is easy to use and highly customizable, making it possible to extract useful insights from the data.



Use case: SelfUser raw data

Data about energy consumption/production and weather conditions are collected from several sensors placed in smart buildings. Energy consumption/production has a granularity of a second, while weather conditions of a minute.

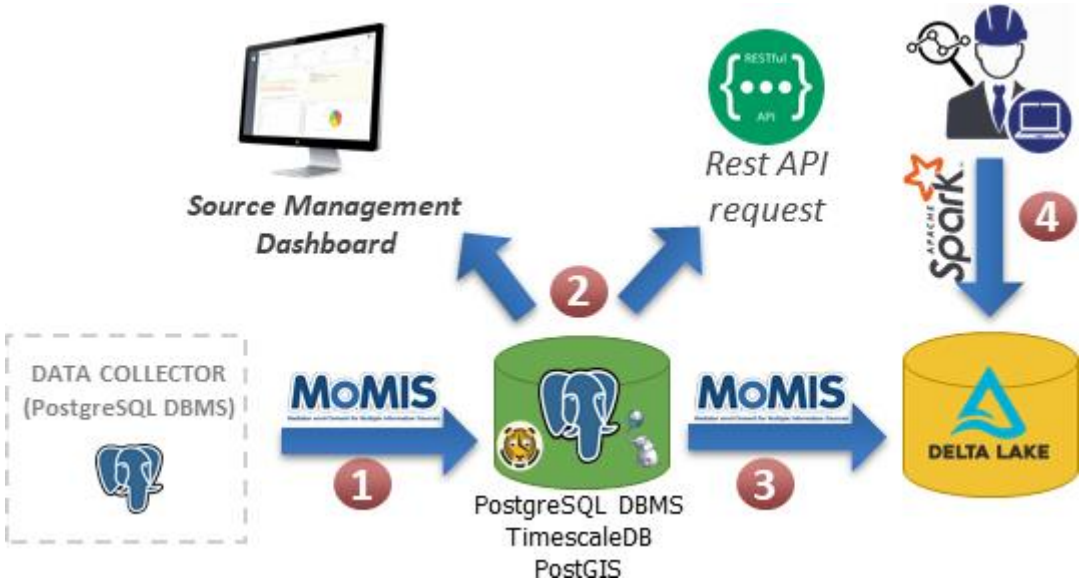
1. MOMIS is used to collect the raw data and store them into the Delta Lake;
2. MOMIS cleans and integrates the data, storing them into the PostgreSQL database with time series extension
3. Users can query these data through the MOMIS Dashboard or API
4. Data stored in the Delta Lake can be queried by using Spark
5. Periodically, to keep the query on the PostgreSQL database efficient, old data are moved from the PostgreSQL database to the Delta Lake



Use case: SelfUser structured data

The processed data with a granularity of 15 minutes are stored in a PostgreSQL DBMS on the servers managed by ENEA.

1. MOMIS is used to collect the data and store them into a PostgreSQL database, optimized with time series management extension;
2. Users can query these data through the MOMIS Dashboard or through API;
3. Periodically, to maintain PostgreSQL efficiency, old data are moved from PostgreSQL to Delta Lake;
4. Data stored in Delta Lake can be extracted by using Spark.



Use case: PELL

The PELL Project data is composed of static data about the lighting infrastructures kept in a **MySQL** database, while that regarding consumption is kept in **DataFrames** on a Hadoop File System.

1. A Spark Script is used to import the data into the Delta Lake;
2. MOMIS integrates the data and stores them into the PostgreSQL DB with time series management extension;
3. Users can query these data through the MOMIS Dashboard or through API;
4. Periodically, to maintain PostgreSQL DB interaction efficient, old data are moved from PostgreSQL to Delta Lake;
5. Data stored into Delta Lake can be queried by using Spark.

Conclusions

- We presented the **Energy Community** (Big data) **Data Platform** that:
  - **Stores** and **analyzes** data of **LECs**
  - Helps the users and administrators to **monitor energy consumption** to **optimize** the **self-consumption**
- The main **lesson learned** is that maintaining the different workflows separated through a modular architecture provides many benefits:
  - **Combine** the **strengths** of **each** adopted **tool**
  - Maximize **scalability** and **flexibility**
  - Guarantee the **data** at **different abstraction levels**
  - **Meet** the **needs** of different **users**
  - This architecture can be **reused** in other **time series management** scenarios

Thanks for your attention

Questions?