



Università
Ca' Foscari
Venezia



Double Degree in Econometrics and Data Science

Internship report

Development of a Scalable Data Infrastructure for a Medium-Sized Energy Company

Supervisor

Professor Pierre Michel

Student

Gabriele Trevisan

Academic Year

2024-2025

NON-PLAGIARISM UNDERTAKING

I, (full name of the undersigned) Gabriele Trevisan

declare being fully aware that plagiarism by copying documents or a portion of a document published in all forms and media, including online publications, constitutes copyright infringement and related rights, as well as outright fraud.

Consequently, I hereby undertake to quote all sources and authors that I've used to write my internship report and its appendices.

Date: 07/09/2025

Signature:

Gabriele Trevisan

Contents

- 1 Introduction** **1**

- 2 Company background** **3**
 - 2.1 Company overview 3
 - 2.2 Business operations 4
 - 2.3 Digital and data ecosystem 8

- 3 Integration of data flows + ticket closure automation** **10**
 - 3.1 Context and needs 10
 - 3.2 Some implementation details 12
 - 3.3 Project outcome 14

- 4 Data warehouse** **15**
 - 4.1 Context and needs 15
 - 4.2 Some implementation details 16
 - 4.2.1 Requirements analysis and data sources 16
 - 4.2.2 System design 16
 - 4.2.3 Data warehouse architecture 19
 - 4.3 Project outcomes 22

- 5 Renovation of the natural gas forecasting system** **23**
 - 5.1 Context and needs 23
 - 5.2 Some implementation details 25
 - 5.2.1 Pipeline design 25
 - 5.2.2 Algorithm improvement 26
 - 5.3 Project outcome 26

- 6 Conclusions** **27**

- 7 Tools Used** **31**

Chapter 1

Introduction

Passuello Fratelli is an Italian company specialized in the supply of wood pellets, diesel fuel, LPG, electricity and natural gas, with the latter two generating more than 80% of the revenue (See Chapter 2 for the company background). To date, the company has a workforce of 74, 18 branches in three different regions (Veneto, Trentino Alto-Adige and Friuli Venezia-Giulia) and a total revenue of about 124 million euros in 2024. In the last three years it has experienced a quick growth due to a change in Italian regulations aimed at completing the liberalization of the energy market (gas and electricity). This has led to an increase of the contracts of 126% (from 38.000 to 87.000) between 2022 and 2024, making Passuello passing from being a small company to a medium and continuously growing one.

To sustain its strong growth, the company has encountered new technological demands: on the one hand, enhancing business processes through automation, and on the other, harnessing the large volumes of data generated by both longstanding and newly acquired clients.

This report outlines my internship experience at the company, which took place between April 2025 and September 2025. During this period, I contributed to three projects aimed at addressing these emerging needs.

The first project (See Chapter 3)involved the development of a system to store and utilize readings data coming from electricity meters, with the goal of automating the closure of *Customer Relationship Management (CRM)*¹ tickets and making consumption data available for analytical purposes. This project lasted one month and significantly improved both the CRM and invoicing processes, improving their reliability and efficiency. Moreover, it has paved the way for the development of other projects to leverage customer consumption data.

¹*Customer Relationship Management* software denotes a category of information systems designed to support the systematic management of customer interactions and relationships, typically by integrating data related to sales, marketing, and customer service processes.

The second project (See Chapter 4) concerned the creation of the first company data warehouse. The development took about three months, which can be divided in two parts. The project began with a preliminary analysis of the technological infrastructure, data sources, requirements, system design (pipeline and database), and the selection of software tools. Subsequently, the code was developed and tested, eventually leading to the deployment of the pipeline. The data warehouse has streamlined reporting, significantly reducing the time needed to generate existing reports and making it easier to create new ones that better leverage company data. Furthermore, thanks to the adoption of a *dimensional model* and tools like *data build tool (dbt)*, extending the data warehouse with additional views or tables is straightforward, allowing even users with limited SQL knowledge the possibility of improving business intelligence.

For the third (See Chapter 5) and last project, I worked on redesigning and optimizing the system used to forecast customers' natural gas consumption. The need arose because the recent growth made it impossible to continue using the old approach, which relied on manually downloading the numerous datasets required for forecasting and loading them into Excel for analysis. Simply put, Excel could no longer handle the required data, since it is limited to about one million rows. Moreover, manually downloading the datasets did not always allow the use of the most up-to-date data, which was crucial for the task. Therefore, the project involved developing two pipelines to clean and store the required data in the company databases, ensuring that forecasts always use the most up-to-date information. The old forecasting algorithm was re-implemented in Python, with the advantages of being significantly faster and capable of running on always-current data.

Chapter 2

Company background

2.1 Company overview

Passuello Fratelli srl is an Italian company active in the energy sector. Founded in 1921 as a grain supplier in Calalzo di Cadore (Italy), in 1937 it entered the fuel sales market by signing a deal with Shell for the sale of gasoline and lubricants. Following the liberalization of the Italian energy market in 2003, in 2005 the company began selling natural gas, and some years later, electricity. Today, it is a retail seller of energy products such as wood pellets, LPG and diesel fuel, but it owes 80% of its revenue to natural gas and electricity. As of June 2025 the company has a workforce of 74 employees, 18 branches across three different regions (Veneto, Friuli Venezia-Giulia and Trentino Alto-Adige) and more than 100.000 active connections among electricity and natural gas. In 2024, the company's total revenue was 124 million euros (+29.45% YoY) and a profit of about 5 million (+ 22% YoY) (see table 2.1).

Following the implementation of a law aimed at completing the liberalization of the energy market¹, by January 1, 2024 for natural gas and by July 1, 2024 for electricity, non-vulnerable household customers² were obligated to switch to an energy supplier in the *free market*³. This shift represented a major opportunity, which Passuello Fratelli successfully seized. Table 2.1 presents key figures of the past three years, including revenue, profit, branches, employees, and active connections.

¹Law No. 124 of 4 August 2017, more information [here](#)

²Non-vulnerable customers are those outside the vulnerable category, whose characteristics are defined by law based on age, economic conditions or disabilities. For more information: [ARERA](#)

³Private companies operating in the energy sector

Table 2.1: Company Performance Overview (2022-2025)

	2022 (31-12)	2023 (31-12)	2024 (31-12)	2025 (30-06)
Revenue (k€)	137,505	96,059	124,366	—
Profit (k€)	856	3,996	4,872	—
Branches	8	9	15	18
Employees	47	59	70	74
Active contracts	38,642	56,414	87,522	100,084

Between 2022 and 2024, Passuello Fratelli increased its profit more than fivefold, expanded its network with 10 new branches, and more than doubled number of active contracts. According to the management, the forecasted revenue for 2025 is 150 million euro. It is worth noting the revenue recorded in 2022. As shown in Table 2.1, it represents the highest value of the three-year period, yet it is associated with the lowest profit. This outcome was driven by the sharp increase in energy prices following the Russian invasion of Ukraine, which was directly reflected in customer bills.

2.2 Business operations

From this paragraph onward, we focus exclusively on the electricity and natural gas department, given that it represents both the company’s core business and the area where the internship was carried out.

To understand how Passuello Fratelli generates value, it is necessary to provide context on the functioning of the energy production chain, as it features characteristics that are markedly different from those of other sectors. For clarity, the mechanisms of the sector are presented separately for electricity and natural gas in the following two paragraphs.

Energy production chain

The electrical system constitutes a network-based system in which the energy withdrawn by end consumers is collectively produced and fed into the grid by generation plants distributed throughout the territory. The network thus functions as a system of communicating vessels, wherein all energy is injected and from which all energy is extracted, without the possibility of determining the specific plant from which the consumed energy originates. Various actors participate within this system: on the one hand, there are producers, distributors, vendors, and

consumers; on the other, the institutions that ensure the proper functioning of both the network and the market (see Figure 2.1).

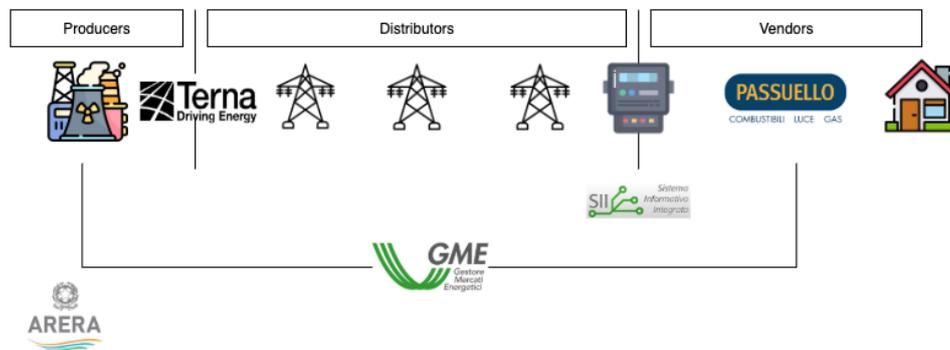


Figure 2.1: Energy production chain

The following section provides a brief description of the entities of interest:

- **Producers** are power plants supplying energy to the grid. Given that the system must be always balanced, it is the *grid operator* that decides who and when they are turned on and off.
- **Grid operator:** in Italy, the role is carried out by *Terna Spa*, which manages the transmission and dispatching of energy, ensuring that production always matches consumption and that frequency and voltage remain within optimal ranges.
- **Distributors** are companies responsible for managing and ensuring the proper functioning of the power grid. In practice, they control the transmission towers, the cabinets, up to the electric meters. In Italy there are about 120 distribution companies (ARERA, 2024) [1].
- **Vendors:** companies such as Passuello Fratelli, which operate in the final phase of the chain by selling energy to end customers. As of 2023, there were 765 vendors in Italy (ARERA, 2024)[1]
- **SII**⁴: Since there are multiple distributors and vendors, and their common point of contact is the meter, an independent entity has been established to manage the communications between vendors and distributions. In practice, whenever a customer requests an operation involving a distributor, such as activations, deactivations, switch-in, switch out, Passuello forwards the request to the SII, which then transmits it to the appropriate distribution company.

⁴Unified integrated system

- **Gestore dei Mercati Energetici (GME)**⁵ is the entity that manages Italy's electricity market to ensure transparent, competitive, and efficient energy trading.
- **Autorità di Regolazione per Energia Reti e Ambiente (ARERA)**⁶ is the independent public authority that carries out regulatory and supervisory activities in the sectors of electricity, natural gas, water services, waste management, and district heating.
- **Customer:** Customers, whether households or industries, are the end-users of electricity.

The electricity price as "raw material" emerges from the GME electricity market, and it is identified by the PUN (*Unique National Price*), and represents the electricity price expressed in kilowatt-hours (kWh). The mechanism behind PUN pricing is complex and beyond the scope of this work. Nonetheless, to provide a straightforward explanation, it is sufficient to know that every morning electricity retailers must communicate to GME their clients' estimated hourly consumption for the following day. GME manages the negotiations between electricity producers and vendors, and in the afternoon makes available the hourly PUN of the following day. It is important to note that the PUN is the same regardless of the sources (for example, green power does not cost less), it varies only by the time of the day and the area⁷.

Passuello Fratelli's business model is based exclusively on variable price contracts, reselling electricity at the PUN plus a *spread* that depends on the contract type.

Given the technical necessity of maintaining a constant balance between production and consumption of electricity, the system is designed to penalize incorrect consumption estimates. In practice, if Passuello Fratelli's customers consume more kWh than purchased for a given day, the company is required to buy the excess at a higher price. Conversely, if consumption is lower than estimated, the company can only sell the surplus at a lower price. This highlights the importance of having the most accurate possible consumption forecasting system: the more precise the forecasts, the higher the potential profits.

Natural gas

The structure of the natural gas production chain (Figure 2.2) is very similar to that of electricity, as both follow a comparable sequence of phases: production, transmission, distribution, and retail. In the case of electricity, power plants generate energy that is transmitted through high-voltage lines, distributed locally, and finally sold to end customers by vendors. For natural gas,

⁵Energy Market Operator

⁶Regulatory Authority for Energy, Networks and Environment

⁷Different parts of Italy have different hourly PUN.

the chain is organized in the same way: gas is extracted from fields (domestic or imported), transported through high-pressure pipelines, delivered via local distribution networks, and then marketed to customers by vendors such as Passuello Fratelli. This parallel structure makes it possible to analyze the two sectors using a similar framework. Nevertheless, two important differences must be highlighted. The first is the grid operator: in electricity it is Terna, while for natural gas the role is carried out by Snam, though the responsibilities are essentially the same, as both are tasked with maintaining network balance and ensuring reliability of supply. The second, and more substantial, difference lies in the functioning of the market. Unlike electricity, which must be consumed at the exact moment of production, natural gas can be stored, and this characteristic shapes its trading mechanisms, pricing structures, and contractual arrangements.

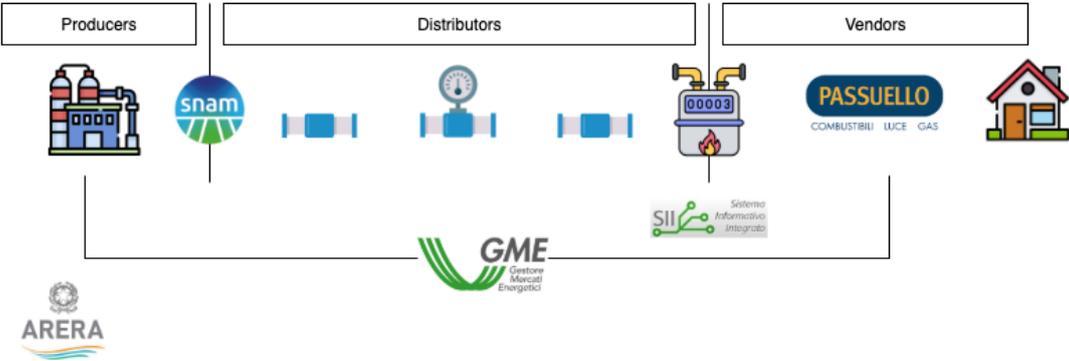


Figure 2.2: Natural gas production chain

The price of natural gas is determined in the same way as any other commodity: through purchase from a seller on the market. In practice, natural gas vendors are required each day to buy on the exchange the quantity of gas their customers are expected to consume the following day. The purchase price depends entirely on market conditions and on the vendor's ability to buy at the most favorable time. In this case, the role of GME is solely to manage and regulate the exchange. Accordingly, every day Passuello Fratelli forecasts the daily consumption of its clients and purchases the corresponding amount of gas on the market. As in the electricity sector, a penalty mechanism applies in the event of inaccurate forecasts. If actual consumption exceeds the forecast, the vendor must purchase the additional volume at a higher price. Conversely, if actual consumption is lower than forecast, the surplus can only be resold at a lower price. This asymmetry creates a direct financial risk, which makes accurate consumption forecasting essential for protecting margins and maximizing profitability.

2.3 Digital and data ecosystem

Technological infrastructure

The company technological infrastructure is entirely on-premise, with basically no reliance on cloud-base platform. The only exception is the new user portal, which runs on *Amazon Web Services (AWS)* to ensure the necessary scalability.

Software and tools

From an operational standpoint, Passuello Fratelli employs four main software applications:

- **Mexal:** Mexal is the accounting software product. It contains all the master data of customers, suppliers, and branches, detailed invoices by item, customer payment due dates and payments. Mexal uses a custom NoSQL system as its operational database, to which direct access is not granted; instead, a replica on a relational database updated hourly is made available. It is important to highlight that the replica database is heavily denormalized and difficult to understand, as tables and columns are labeled with unclear names.
- **VTE Next:** VTE is the company's CRM product. It is a highly customizable open-source software to which specific functionalities for the business type have been added by an external software house. All customer relationship operations are managed through VTE, including tasks such as opening or closing utilities, generating quotes, registering new customers, and handling transfers of ownership. The associated data is stored in an Microsoft SQL database, which is accessible in read-only mode.

- **MGE:** it's a custom software that serves as the engine for the electricity/gas division. Within MGE, all data related to utilities are stored, including technical data of meters, *PDRs*⁸, *PODs*⁹, substations, readings, etc. MGE primarily handles invoice generation, which are then entered into Mexal through APIs. In this case as well, a relational database (MS SQL) is used, which can be accessed.
- **Superba:** is a custom software that handles the commercial management of all non-gas/electricity divisions. This software will not be addressed within this report, as all projects have been developed for gas and energy sectors.

Data and Business Intelligence

The *data governance* is problematic. Data is often dirty and replicated across multiple databases, making it difficult -if not impossible- to have a single source of truth. With data distributed across three different databases (Mexal, VTE Next and MGE), performing data extractions for analytical or operational purposes requires more time, complexity, and effort than necessary, while also increasing the risk of errors. Furthermore, a large amount of valuable data, such as natural gas and electricity consumption, along with information provided by distributors, was not stored in the company's databases, despite being made available through distribution platforms.

In the absence of a robust data management infrastructure, this information was essentially left unused, and no real value was extracted from it. This was problematic because it prevented the company from exploiting data for strategic purposes such as performance monitoring and business intelligence.

⁸Redelivery point for natural gas meters.

⁹Point of delivery for electric meters

Chapter 3

Integration of data flows + ticket closure automation

3.1 Context and needs

Every time a customer asks for an operations that involves the grid, such as activations, deactivations, switch-in, switch out or a takeover, Passuello Fratelli must forward the request to the relevant distribution company. Since there are hundreds of distribution companies and hundreds of vendors, there is an online platform, called *SII*, that acts as a bridge between the two parts of the system. Through SII, Passuello Fratelli submits the operation request specifying the customers (identified by their POD/PDR). Once the distribution performs the operation, it uploads the outcome on the platform as a data feed.

It is important to note that all data exchanges between vendors and distribution companies are standardized, following specifications set by regulatory institutions, and pass through SII, which ensures that vendors' requests are routed to the correct distribution company and that the responses comply with the data standards. In addition to operation outcomes, **consumption readings** coming from the meters are made available to the vendors through the SII. These readings are used to calculate the total amount of the invoices during the billing process.

In this context, as my first project, I was requested to develop a system to optimize the way the results of the operations were recorded in the CRM software, since this task was entirely performed manually.

More specifically, up to that moment, the workflow of operations was performed as follows (see figure 3.1):

1. When a customer asked an operation, an operator in Passuello Fratelli manually created a ticket in the CRM software, in order to store its status.
2. A operator uploaded all the requests received on the SII
3. Every day, an operator needed to log into the SII to check whether the request's outcomes had been received.
4. For requests with a positive outcome, a operator had to manually update and close the corresponding CRM ticket.

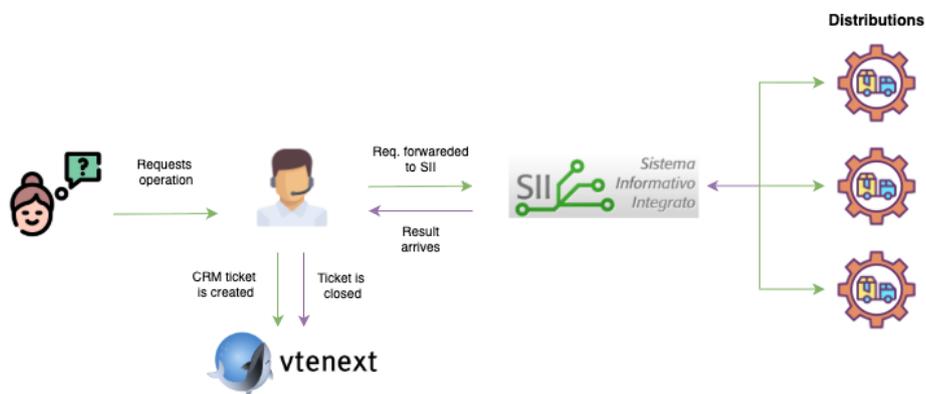


Figure 3.1: Operation request workflow.

This workflow was leading to several problems. First, it was highly time-consuming. Operators had to check the results and update the CRM tickets for each request manually, but with the rapid growth of the company, there was a large number of operations to review. In addition, the mechanism used to associate operations with their corresponding readings was cumbersome and did not always capture the correct value.

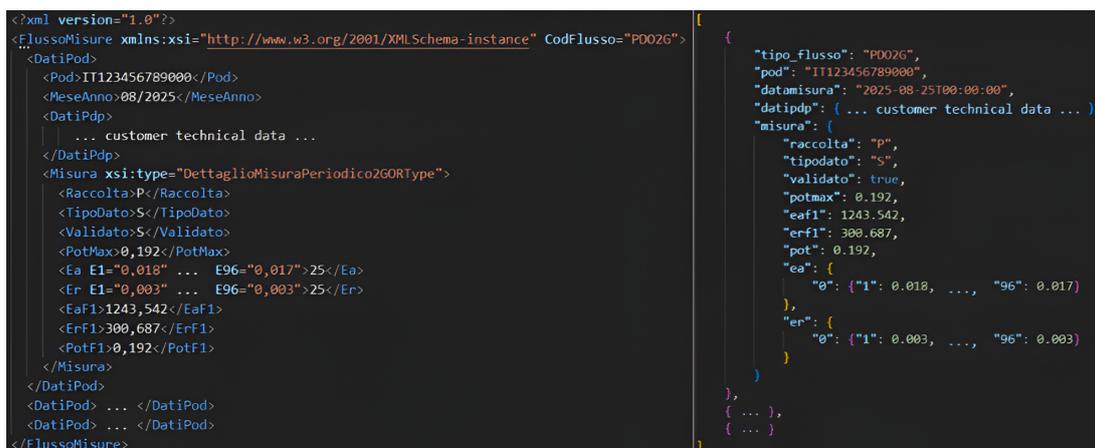
The starting point in addressing the problem was obtaining access to the raw data files provided by the distribution companies. This is made possible by the SII, which not only allows data access through the website but also enables vendors to connect to a cloud folder where all data feeds are regularly uploaded as XML files.

Accessing and storing these data would have also contributed to improving the billing process. At that time, Passuello Fratelli relied on a licensed software tool that merely served as an interface to the cloud folder containing these readings. The project, therefore, not only aimed to improve the efficiency of the CRM process, but also had the potential to eliminate the need for such a license, thereby generating cost savings for the company.

3.2 Some implementation details

The first part and most important part of the project regarded the realization of a pipeline capable of download, process and store the data feeds into Passuello Fratelli's databases, to make them accessible and easy to use. The *data feeds* are delivered as XML files, uploaded to the cloud at irregular times during the day. To date, there are 18 different types of XML files, each representing a specific operation and with a slightly different structure from each other. Each file contains a set of PODs on which the operation was performed, and for each POD it provides the technical data, the outcome of the operation, and an associated meter reading. The only common part between the XML types is the block regarding the reading. Given the differences between the structures and the fact that it is very likely that in the future new types of feed will be added, there was the necessity to build a system capable of adapting to changes without breaking.

The first idea was to transform the XMLs to obtain a tabular structure in order to store them in a relational database. However, we quickly realized that this approach was not suitable, as it would have resulted in either a single large table full of NULLs, leading to excessive memory usage, or the creation of a large number of tables, essentially one per XML type, making the development long and delicate and the maintenance very difficult. Since we needed elasticity on the data structure, we decided to write a parser that splits the XML files into single measurements and converts them into JSON documents, which are then stored in a document database. Using JSON allowed us to build a parser that could adapt to all different types of data feeds, while also giving us the possibility to structure the data according to our needs. This approach enabled us to develop a system that is resilient to changes, efficient, and modern.



```
<?xml version="1.0"?>
<FlussoMisura xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" CodFlusso="P002G">
  <DatiPod>
    <Pod>IT123456789000</Pod>
    <MeseAnno>08/2025</MeseAnno>
    <DatiPdp>
      ... customer technical data ...
    </DatiPdp>
    <Misura xsi:type="DettaglioMisuraPeriodico2GORType">
      <Raccolta>P</Raccolta>
      <TipoData>S</TipoData>
      <Validato>S</Validato>
      <PotMax>0,192</PotMax>
      <Ea E1="0,018" ... E96="0,017">25</Ea>
      <Er E1="0,003" ... E96="0,003">25</Er>
      <Eaf1>1243,542</Eaf1>
      <ErF1>300,687</ErF1>
      <PotF1>0,192</PotF1>
    </Misura>
  </DatiPod>
  <DatiPod> ... </DatiPod>
  <DatiPod> ... </DatiPod>
</FlussoMisura>
```

```
{
  "tipo_flusso": "P002G",
  "pod": "IT123456789000",
  "datamisura": "2025-08-25T00:00:00",
  "datipdp": (... customer technical data ...),
  "misura": {
    "raccolta": "p",
    "tipodato": "S",
    "validato": true,
    "potmax": 0.192,
    "eaf1": 1243.542,
    "erf1": 300.687,
    "pot": 0.192,
    "ea": {
      "0": {"1": 0.018, ..., "96": 0.017}
    },
    "er": {
      "0": {"1": 0.003, ..., "96": 0.003}
    }
  }
},
{ ... },
{ ... }
```

Figure 3.2: XML-JSON parsing example. Each block represent a couple POD-day.

Once the ingestion part was achieved, we built the workflow to connect to the CRM software

for automatically closing the tickets. At high level, the mechanism is simple: for each document, using the CRM APIs, we check if there is a corresponding open ticket. If so, we update the ticket information and set it as closed. To make the system more resistant we implemented two additional features:

- **Document hashing:** since distributions sometimes send duplicate data feeds, we implemented a system that uses hashing to check whether a document has been already used to close a ticket.
- **Closure retry:** if the process fails and the system cannot close a ticket, it retries up to five times over the following five days

The main tools used for the project are MongoDB for database, Python for the logic, and Airflow for the orchestration.

The following diagram better illustrates the new workflow:

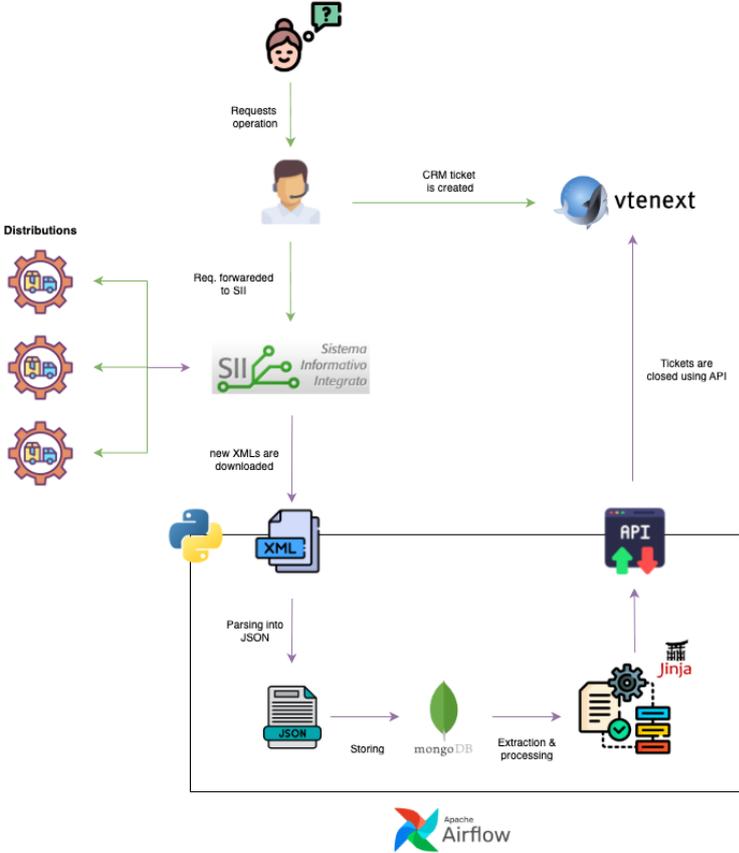


Figure 3.3: CRM ticket closure automation workflow

3.3 Project outcome

The implementation of the project has significantly improved both the management of operation results and invoicing. Regarding the operations, the automation of ticket closure allowed the Passuello Fratelli operators to manually check only the tickets that the system didn't close, reducing the time required for the process from few days to some hours, and the number of workers from two/three to one. Between June and July, the system handled, on average, about 2992 data feeds and 148 ticket closures per week. By reducing manual intervention, the risk of incorrect data entry by operators has also been eliminated. Most importantly, the creation of this system has made the process scalable, enabling it to support the company's ongoing growth.

The creation of the database also had positive effects on the invoicing process, making it more efficient: previously, readings were extracted using a software that wasn't always up-to-date with the new data feed types, failing to process the new ones and consequently causing data loss. On top of that, their servers were slow, introducing a bottleneck in the process. Since implementing the new system, this software is being used progressively less, and it is likely to lead to complete abandonment, with consequent license cost savings of approximately €12,000 per year. With the database, data extraction is faster, and it allows access to information that wasn't available before.

Finally, the internalization of data opens the door to other projects, such as developing the model for forecasting electricity consumption, which currently relies on a very costly software, and the implementation of a new section in the user portal where customers can view their daily consumption and, if desired, download the data.

Chapter 4

Data warehouse

4.1 Context and needs

A data warehouse is a centralized system for collecting, storing and managing data from heterogeneous sources. Unlike operational databases, it is designed specifically for supporting analytics and the decision-making process: data is cleaned, organized, stored and consolidated in order to create a single source of truth, easily accessible by Business Intelligence tools. Before the implementation of the project, no such system was in place within the company. Data extraction for analytical purposes was performed by querying directly the operational database of the various software in use when needed. As outlined in Section 2.3, Passuello Fratelli relies on three distinct applications for accounting, billing, and customer relationship management (CRM). This fragmentation introduced the risk of inconsistency, since identical data was often duplicated across multiple systems.

To mitigate this, nightly synchronization pipelines were implemented to align key records, such as customer information and accounts, across the databases. Nonetheless, the absence of a unified single source of truth remained a significant limitation. This motivated the development of a data warehouse, designed to consolidate, clean, and prepare data specifically for analytical purposes. Moreover, extracting data from operational databases presented additional challenges. These systems are typically highly normalized, with table structures and column names that are neither intuitive nor descriptive. As a result, the process of querying them for analysis was both time-consuming and error-prone.

The data warehouse development took about three months. First, a requirements analysis was carried out, with the goal of defining the characteristics and features that are useful to stakeholders. Afterwards, we designed the system, focusing on the technical implementation and the selection of tools.

Finally, we wrote the code and performed the tests, leading to the deployment of the data warehouse.

4.2 Some implementation details

4.2.1 Requirements analysis and data sources

Given the large amount of data, source and business processes, a series of interviews with key stakeholders was conducted to gain a clearer understanding of objectives and priorities and to outline a preliminary roadmap. These meetings revealed that the foremost priority was to improve access to billing and payment data. Furthermore, it was agreed that the data warehouse should be updated incrementally, with a refresh scheduled at least once per day.

After the requirements had been defined, we proceeded by analyzing each operational database used by the software. For each data source we created a diagram showing the tables and their relationships, data types and, when necessary, the explanation of the fields.

Subsequently, based on the diagrams, the tables and fields useful for analytical purposes were selected.

The product of the analysis was a document in which we reported, for each database, a description of the usage, the type of database management system, the diagrams, tables and fields relevant for the data warehouse as well as other technical information useful for both developers and stakeholders.

4.2.2 System design

Before presenting the technical choices and delving into implementation details, it is necessary to briefly introduce the concept of *dimensional modelling* and the fundamental components of a data warehousing system. In the following paragraphs, we provide some theoretical background on the topics to give the readers a better understanding of the work we have done.

Dimensional modelling

Dimensional modeling is a data modeling technique used for analytical purposes. It is considered the standard approach because it ensures simplicity, comprehensibility, and fast query performance[2].

Dimensional model introduces the concepts of *facts* and *dimensions*. A fact is a quantitative and measurable value belonging to a business process, on which aggregation operations such as

sum, average, minimum, and maximum can be performed. The characteristics that describe a fact, and through which measurements can be aggregated, are called dimensions.

For a practical example, let's consider a retail company with multiple stores and focus on sales. Every day, transactions occur at the points of sale. We can model each transaction as a fact, as follows:

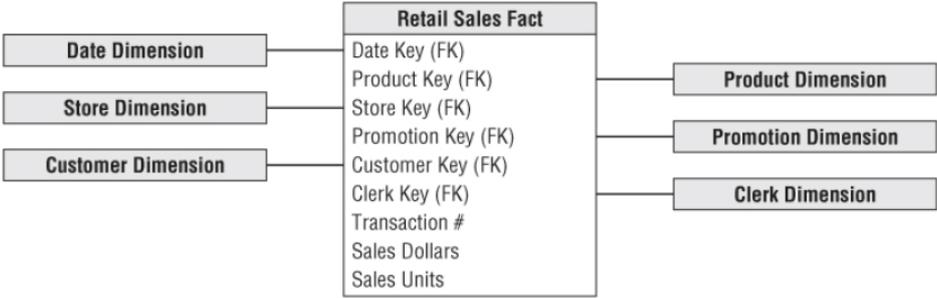


Figure 4.1: An example of a *fact*. Figure adapted from Kimball et al. (2013)[2]

For each transaction there are the sales amount and the sold units as quantitative measures. Moreover, we have the *dimensions* (characteristics) that describe them: date, product, store, customer. Each dimension is itself a table, containing the attributes that define it.

For example, the product dimension could be something like this:

Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Abrasive Indicator
Weight
Weight Unit of Measure
Storage Type

Table 4.1: An example of a *dimension*. Table adapted from Kimball et al. (2013)[2]

In the dimensional model there are two ways, called *schemas*, to structure the relationship between facts and dimensions: the *star schema* and the *snowflake schema*.

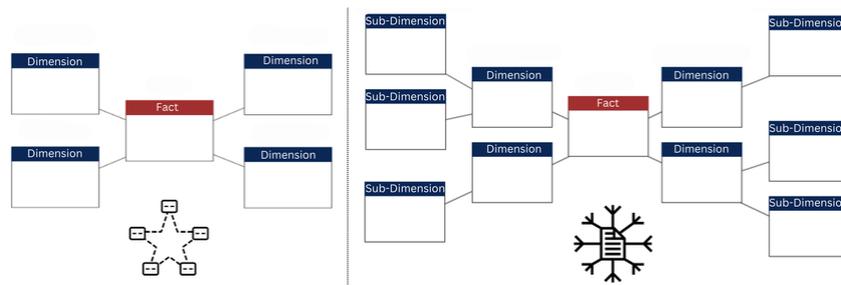


Figure 4.2: Graphical representation of star schema (left) and snowflake schema (right)[3].

The star schema is the simplest and most widely used model, consisting of only a single hierarchical level between facts and dimensions. Its main characteristics are simplicity, ease of understanding, and fast query performance. Dimension tables are not normalized, which results in some data redundancy. In the snowflake schema, dimension tables are normalized, eliminating redundancy but increasing complexity in both usage and querying.

Data warehousing concepts

There are different ways and tools for building a data warehouse, but we decided to introduce only the architecture proposed by *Kimball et al.* [2], from which we took inspiration for our project.

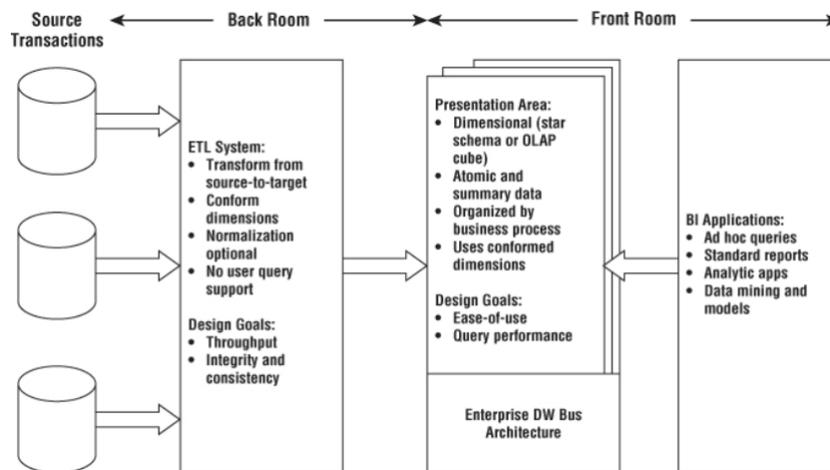


Figure 4.3: Figure adapted from: Kimball et al. (2013)[2]

A data warehousing system is made of three main components: data sources, the *Extract, Transformation and Load (ETL)* system, and the presentation area.

- **Data sources:** These are all the data sources useful for Business Intelligence. They normally consist of the operational databases of the software used within the company, but they can also be individual or multiple files in JSON, CSV, XLSX, XML, ... formats.

- **ETL/ELT systems:** systems that extracts data from the sources, transforms it, and loads it into the presentation area, making it available to business users for analysis. During the transformation phase, the data is cleaned, integrated, enriched, and consolidated, ultimately generating the *fact* and *dimension* tables.
- **Presentation area:** The presentation area is where data is organized as facts and dimensions, and it is available for direct querying by users, report writers, and other analytical business intelligence applications.

4.2.3 Data warehouse architecture

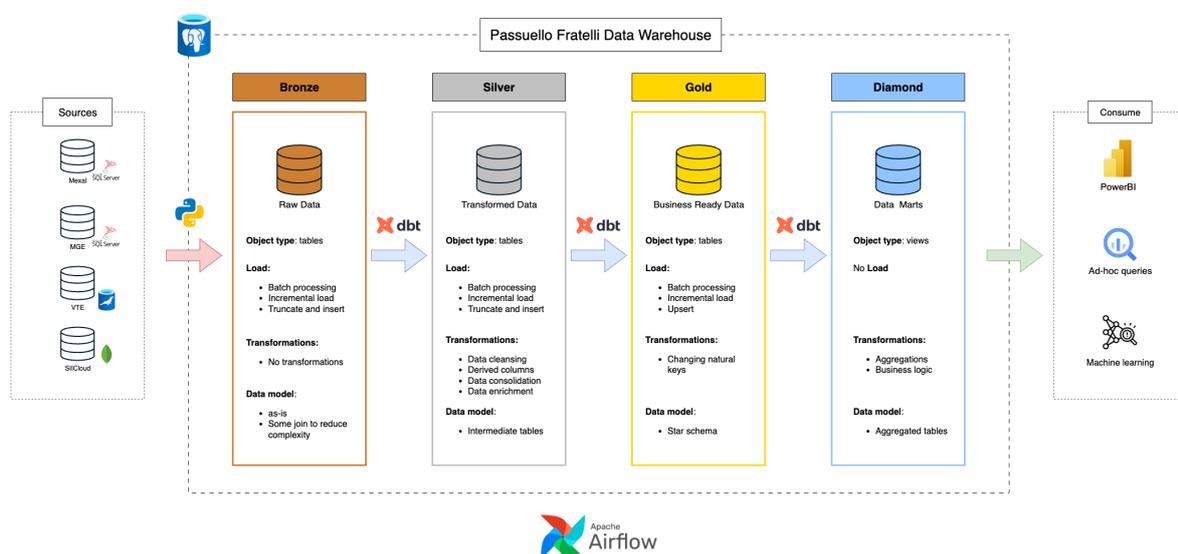


Figure 4.4: Passuello Fratelli DWH architecture diagram

The data warehouse is implemented through a **medallion architecture**¹ in which each layer has a specific responsibility:

- The **Bronze layer** stores the raw data incrementally extracted from the sources at the end of each *ingestion*. To simplify, we can say that for each table of interest in the operational databases there is a corresponding table with the same schema in the Bronze layer.
- In the **Silver layer**, data from the Bronze layer is cleaned, aggregated, and consolidated to create the dimensional model in a star schema.

¹Medallion architecture is a layered data approach that organizes raw, cleaned, and curated data to ensure quality, usability, and scalability.

- In the **Gold layer**, once the transformation in the Silver layer is completed, surrogate keys² are calculated, and the data is archived in the Gold layer. From this layer onward, read-only access is granted to business intelligence users. The data is clean and ready to be queried in the form of a dimensional model using a star schema.
- In the **Diamond layer** are stored views and tables for specific purposes requested by the business intelligence users.

Extraction, Load and Transformation phase

For the project, we decided to use an *Extraction, Load and Transformation* (ELT) approach: raw data is extracted from the sources and loaded directly into the data warehouse, where all transformations take place.

For the *ingestion* phase³, we developed a custom system (Figure 4.5) using *Jinja*⁴ as a templating engine to implement incremental updates. The process works as follows: first, DDL queries⁵ are executed to set up the Bronze layer tables. Then, for each table requiring an incremental update, the system queries the data warehouse to retrieve reference parameters (for example, the ID or date of the last record loaded in the previous ingestion). These parameters are then used to populate *SQL query templates*, which are not stored as plain SQL but as template files where the WHERE clauses are dynamically filled. In this way, the system extracts only new or modified records compared to the previous load, ensuring both efficiency and traceability.

²A surrogate key is an artificial unique identifier for a database record, typically a number, used instead of a natural key.

³(Extraction + Loading)

⁴Jinja is a fast, expressive, extensible templating engine that allows writing code similar to Python syntax. For more information, see Jinja's website [4]

⁵SQL commands used to define, modify, or remove database structures, such as tables, indexes, and schemas. Examples include CREATE, ALTER, and DROP statements.

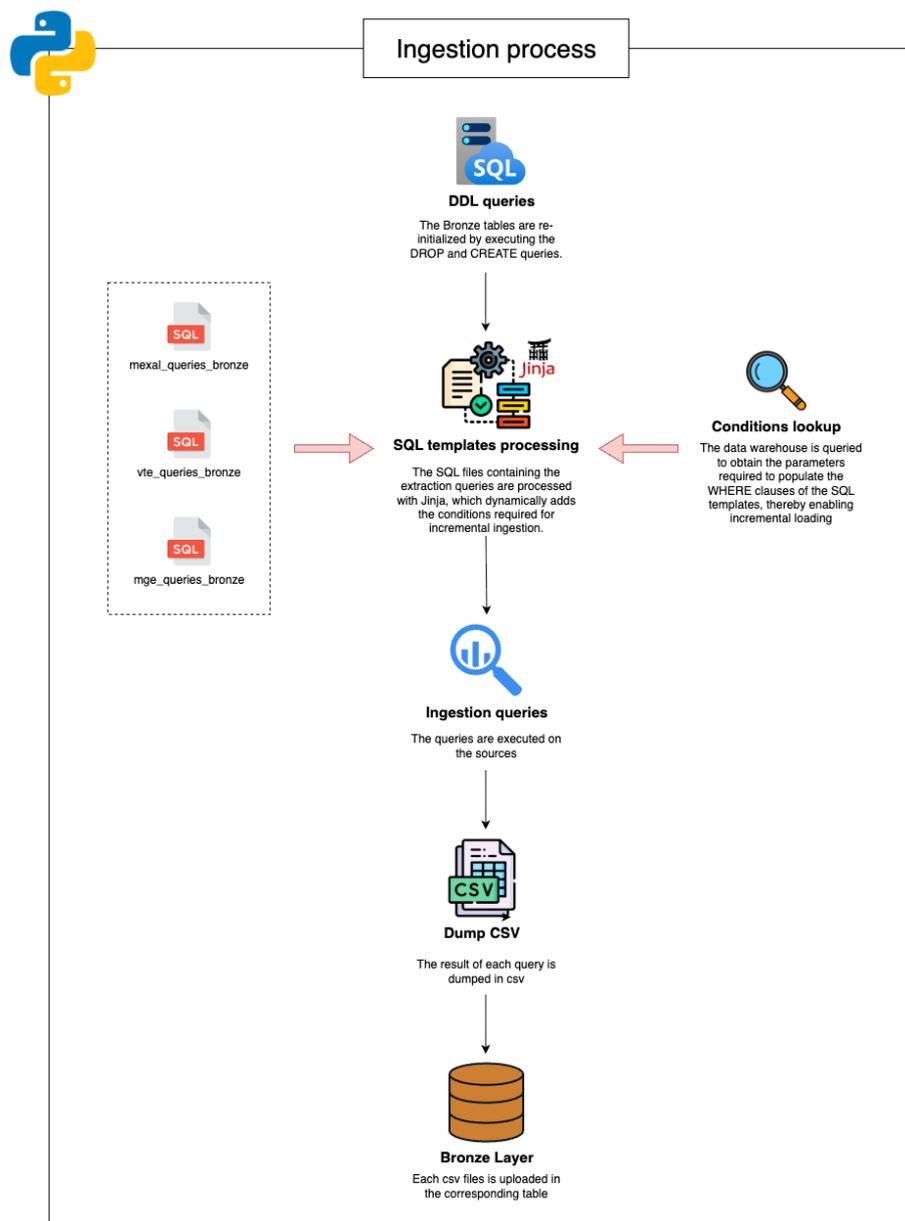


Figure 4.5: Representation of the ingestion process

For managing transformations, we used data build tool (dbt), an open-source tool that simplifies and standardizes the data transformation process by introducing software development best practices into the analytics world, such as testing, code reuse, versioning, and documentation. Specifically, *dbt* allows the definition of reusable SQL models, automatically manages the execution flow through a dependency graph, implements tests to ensure transformation correctness, and automatically generates project documentation.

Finally, we used *Apache Airflow* to orchestrate the ELT process. Airflow is a widely adopted open source tool that manages data processing workflows, allowing the definition and schedul-

ing of pipelines in Python. It creates chains of tasks that can include, among others, both Python function calls and bash commands.

The data warehouse relied on PostgreSQL, running on an Amazon Web Services EC2 ⁶ instance, to ensure operability and scalability.

4.3 Project outcomes

The data warehousing system meets all the requirements: it supports incremental updates, and the ingestion mechanism allows multiple ingestions per day without compromising data consistency. The data warehouse is currently used to generate specific reports and to speed up the preparation of documentation regularly required by energy market regulators. More broadly, the project has enabled access to large volumes of data that were once difficult and time-consuming to obtain. Thanks to the ingestion mechanism and *dbt*, new data sources can be integrated quickly and efficiently, making the system dynamic and easily scalable.

⁶Amazon EC2 (Elastic Compute Cloud) is a web service that provides resizable virtual servers in the cloud.

Chapter 5

Renovation of the natural gas forecasting system

5.1 Context and needs

Every day, Passuello Fratelli must forecast their clients' natural gas consumption for the upcoming days. As explained in Chapter 2.3, producing accurate forecasts is key to the Company: the more precise the prediction, the lower the penalties incurred and, the greater the profits. In addition, for budgeting purposes, consumption must be forecast for the entire thermic year¹, with monthly updates at the beginning of each month covering the subsequent four weeks.

The mechanism used to calculate the prediction for a given meter and date is pretty simple, but it requires a lot of data. Before presenting the calculation, it is important to introduce the concept of *consumption profile*.

Each user is profiled by the grid operator according to their type of consumption, for example whether it is residential, industrial, or more or less sensitive to temperature.

There are 16 different consumption profile types, and for each type, SNAM provides the following data: ²

- **daily share of annual consumption (DAC)**: It's an estimate, expressed as a percentage, of the daily consumption with respect to the annual total. For example: "On April 16th, 2024, the consumption amounts to 0.0000001% of the annual total."
- **Weather Sensitivity Share (WSS)**: It's an estimate, expressed as a percentage, of the share of daily consumption that is sensitive to temperature. For example: "On April 16th, 2024, 75% of my natural gas consumption is temperature-dependent".

¹The term *thermic year* refers to the period from October 1st to September 30th

²The grid operator for natural gas

Moreover, twice a day, SNAM publishes the *WKR coefficients*³ for the following days. WRKs are weather correction factors, used to correct the share of consumption sensitive to the temperature.

Now that we have presented all the parts involved, we can see the formula

$$C_{meter,date,profile} = WSS_{profile} \cdot \frac{C_{meter,date[year-1],profile}}{wkr_{date[year-1]}} \cdot wkr_{date} + (1 - WSS_{profile}) \cdot C_{meter,date[year-1],profile} \quad (5.1)$$

In practice, to predict the consumption of a meter for a specific day, we take the consumption of the most similar⁴ day of the previous year (*date[year-1]*), we separate the temperature sensitive and non sensitive using parts using the $WSS_{profile}$, we correct the former using the corresponding WKR, and then we sum up the two parts.

The last point to clarify is the source of historical consumption values ($C_{meter,date[year-1],profile}$). There are two ways to obtain them, depending on the type of *treatment*⁵ of the user. If it is daily, the value for the corresponding day is used. If only the monthly value is available, the DAC coefficients are used to estimate the daily consumption. All these data are supplied by *SNAM* and *SII*, which publish and updates them with different frequencies. For example, even historical consumptions have different publications, each one giving more reliable values. In order to make the most accurate prediction, it is important to be up to date with the publications.

Until now, forecasts have been performed using Microsoft Excel, manually downloading the data files, loading them into the software, and calculating the results. The main problem that highlighted the need to change this approach was that, given the company's growth, Excel did not support enough rows to store all the data required for computation. Furthermore, since predictions needed to be calculated daily and the datasets were large, the most up-to-date data was not always used.

In this context, I was asked to redesign the system to address all the issues it had and, more importantly, to make it scalable.

³For more information, see *Article 6.1 of this ARERA publication*[5]

⁴Similarity in terms of calendar. For example, for 25/12/2025 we consider 25/12/2024, but for Monday, 08/09/2025 it is used Monday 09/09/2024

⁵The treatment indicates the frequency at which consumption readings are received. It can be daily or monthly

5.2 Some implementation details

The redesign of the system was divided into two phases. In the first phase, we designed and implemented new pipelines to store all the data required for the prediction in the company databases. We then reimplemented the existing algorithm in Python to ensure the system continued to function as before. In the second phase, the focus shifted to redesigning the algorithm itself in order to improve the accuracy of the forecasts. The next sections discuss the two phases in greater detail.

5.2.1 Pipeline design

Given the large volume of data, the different sources from which they originate, and the need to build a system that always uses the most up-to-date information, it was natural to start by creating data pipelines that store and process the information, making it suitable for forecasting purposes. In practice, three distinct workflows were implemented and executed daily via Apache Airflow to handle the storage of:

- The WKR coefficients, updated twice a day and accessible via API
- The cleaned historical consumption data, provided by the SII every two weeks and accessible through cloud folders
- The registry of active users, provided by SII and updated twice a month

For each category, the data was historized, using the upload timestamp as a reference to always retrieve the latest available record. For the remaining data (WSS and DAC coefficients), since they are updated by SNAM only at the beginning of the thermic year, we created a script to take the raw files as input, process them, and load them into the database, thus streamlining that process as well.

Once the pipelines were fully operational, the old forecasting algorithm was reimplemented in Python, querying the databases at each execution to ensure the use of the freshest data. In this way, within a relatively short time, the prediction process was both improved and accelerated.

5.2.2 Algorithm improvement

The newly created database makes it possible to easily experiment with different predictive approaches and models. In the first phase, we decided to follow the approach suggested by SNAM, which in its publications describes the techniques they use for their predictions. At the time of writing, this phase has only just commenced, and no results are yet available. The plan is to evaluate Machine Learning models to determine potential improvements in predictive performance.

5.3 Project outcome

The implemented solution has immediately demonstrated its advantages. First of all, the issues arising from the previous methodology have been resolved through the development of a more effective and scalable system, capable of supporting the company's growth. The reimplementation in Python has made the forecasting process significantly faster and opens the way to further analyses and the development of dedicated tools. Moreover, the creation of the database supporting the forecasting system has introduced a new and valuable source of data for the company, from which meaningful insights can be extracted. Another important benefit lies in having a clean and well-structured database, which makes it easier to experiment with new models aimed at improving forecast accuracy.

Chapter 6

Conclusions

During my internship at Passuello Fratelli, I worked on three main projects. The first on acquiring data from the distribution network and integrating it into the company's databases, with the primary objective of automating a process related to CRM ticket closure.

The need of a new system arose because, with the rapid growth that the company is experiencing, operators were struggling to perform certain operations manually, due to the increasing workload. The adopted solution was the development of a pipeline that, every thirty minutes, connects to the cloud service where the data are uploaded, downloads and processes them, and then stores the results in a MongoDB database hosted on an AWS EC2 instance.

Since the raw data consisted of 18 similar XML files with slightly different structures, the main technical challenge was deciding on a format to store the data that would, on one hand, be easily usable, and, on the other, allow for a not overly complex infrastructure that could be maintained efficiently. The solution consisted of developing a custom XML-to-JSON parser, which made it possible to leverage the flexibility of JSON to design a more generic approach. At the same time, it allowed the definition of specific structures for certain parts of the data flows, thereby simplifying their use in subsequent processing steps. Once the ingestion mechanism was operational, integration with the CRM system was implemented to automate ticket closure.

The technical solution adopted proved highly effective, as it not only addressed the original problem but also leveraged the created database to streamline the invoicing process. Regarding tickets, taking into consideration June and July, the pipeline processes an average of 2,992 files per week, corresponding to approximately 500,000 meter readings and 148 tickets automatically closed. From an operational perspective, the solution reduced the required personnel from 2–3 people to just one, allowing tasks that previously took up to three days to be completed within a single morning. Additionally, the creation of the database

accelerated the billing process by enabling faster extraction of more complete consumption readings. A further positive outcome of the project was the near-complete elimination of a previously used software application, which had an annual licensing cost of nearly €12,000.

The second project involved the creation of a data warehouse. The need arose because the company uses three different software systems for accounting, invoicing, and CRM. Each system operates on a separate database, and since some data, such as customer master records, needs to be shared across the three systems, duplication and synchronization mechanisms had been implemented between the databases. The logical and physical separation of these three business processes meant that there was no **single source of truth**, requiring direct interaction with operational databases to generate reports, making the process long, complex, and prone to errors.

The development lasted approximately three months and was roughly divided into three phases. The first phase involved requirements analysis, during which were conducted several interviews with stakeholders and potential users of the future data warehouse to define its features and priorities. In addition, an in-depth analysis of the data sources to be used was carried out, examining the structure and characteristics of the various databases. Finally, an initial set of documentation was produced, summarizing the project objectives, desired features, and the findings from the data source analysis.

Subsequently, the design and implementation phases took place, during which the database architecture was defined, the functioning of the ETL/ELT system was established, and the tools to be used were selected.. Firstly, it was decided to run the database on an AWS EC2 instance for efficiency and scalability reasons. PostgreSQL was chosen as the RDBMS due to its rich feature set, open-source nature, security, and reliability. Regarding the ETL/ELT system, an Extraction, Loading, and Transformation approach was adopted. For the *ingestion* phase (Extraction + Loading), a custom Python system was developed to meet the required incremental updates and frequency of data refresh. For transformations was used *DBT*, an open-source tool for data transformation, as it greatly facilitates the creation of transformation pipelines by providing useful features such as code reuse and testing tool. Within the database, a *medallion architecture* with four layers (bronze, silver, gold, and diamond) was implemented to logically separate the different stages of data transformation. The bronze layer serves as a repository for raw data, while the gold and diamond layers contain cleaned and structured data following the dimensional model, making them easily queryable. Apache Airflow was used to orchestrate the pipeline execution.

Since its deployment, the data warehouse has already demonstrated its reliability and improvement over the previous data extraction methods. It is currently used to generate certain reports and to accelerate the production of documentation regularly required by energy market regulatory authorities. Overall, the project has opened the door to extracting large amounts of information that were previously difficult and time-consuming to obtain. The ingestion mechanism combined with DBT makes it simple and fast to add new data sources, making the system dynamic and easily extensible. The main potential improvement, aside from adding more sources, is the creation of dashboards based on the data in the data warehouse.

The third project focused on redesigning the system for predicting clients' natural gas consumption. For Passuello Fratelli, it is crucial to produce accurate daily forecasts for their clients' portfolio in order to purchase the right amount of gas on the market. Additionally, monthly and annual consumption estimates are required for budgeting purposes. Although the forecasting formula itself is relatively simple, it relies on a wide variety of data sources. The old system was no longer viable because it relied on Excel, which required manual downloading and loading of all necessary datasets, making the process slow and inefficient. Moreover, with the company's recent growth, Excel was unable to handle the increasing volume of data anymore.

The solution involved creating a database, fed by pipelines that daily download, reorganize, and store all the data needed for forecasting. This approach allowed, on one hand, to generate predictions using always up-to-date data, and on the other, to simplify data access for analytical purposes. Building this database also enabled the reimplementation of the old forecasting mechanism in Python, making it faster, more efficient, and more reliable. Moreover, the creation of the database paved the way for further improvements to the forecasting system, as data can now be extracted much more easily for testing and validation purposes.

During my internship, I observed how, although the company had access to a large amount of data, much of it remained unused or underutilized. This was true both analytically, due to the lack of business intelligence tools, and operationally, where much of the available information could have been leveraged to optimize and increase the efficiency of certain business processes. Against this backdrop, the projects I worked on contributed to initiating a path toward better utilization of the available data. On one hand, the development of solutions such as the data warehouse and the ticket closure system allowed the company to begin transforming data into tangible operational and decision-making tools, while also making certain business operations more efficient and supporting growth. On the other hand, the creation of various pipelines and databases laid the groundwork for a more systematic and valuable use of information, providing

an organized and scalable infrastructure upon which the company can rely for its development.

During my internship experience, I focused primarily on the "data engineering" side of the data domain, implementing pipelines for extracting, cleaning and organizing data to make them more accessible and useful. In doing so, I had the opportunity to learn several tools commonly used in this field, such as Apache Airflow and DBT, that are currently in high demand in the job market ¹, MongoDB and PostgreSQL. This experience allowed me to deepen my understanding of database management, including tasks such as designing schemas, managing users, defining access permissions and creating indexes to improve query performance. Moreover, since some projects were hosted on AWS EC2 instances, I had the opportunity to learn shell commands to connect to the servers, install packages, start and stop services and manage security by restricting access to specific connections. The only aspect I felt was missing was the opportunity to explore topics more closely related to data science, such as model development or general data analysis tasks. Nonetheless, I believe that this internship experience has been highly formative and has allowed me to carry out full-stack projects of significant value for my future professional career.

¹Reddit discussion on Airflow and DBT[6]

Chapter 7

Tools Used

The following tools were used during the development of the projects:

- **Python** — <https://www.python.org/>
- **PostgreSQL** — <https://www.postgresql.org/>
- **MongoDB** — <https://www.mongodb.com/>
- **dbt** — <https://www.getdbt.com/>
- **Apache Airflow** — <https://airflow.apache.org/>
- **Amazon Web Services** — <https://aws.amazon.com/>
- **Jinja** — <https://jinja.palletsprojects.com/en/stable/>

Bibliography

- [1] ARERA. “I numeri dei servizi pubblici.” (2024), [Online]. Available: <https://www.arera.it/comunicati-stampa/dettaglio/arera-i-numeri-dei-servizi-pubblici>.
- [2] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd. Wiley Publishing, 2013, ISBN: 1118530802.
- [3] A. Jaiswar, *Data warehouse: Architecture overview*, Jun. 2024. DOI: 10.36227/techrxiv.171779354.46288271/v2.
- [4] Jinja. “Jinja.” (2025), [Online]. Available: <https://jinja.palletsprojects.com/en/stable/>.
- [5] Arera. “Testo integrato delle disposizioni per la regolazione delle partite fisiche ed economiche del servizio di bilanciamento del gas naturale (tiscg).” (2019), [Online]. Available: <https://www.arera.it/fileadmin/allegati/docs/19/148-19a11.pdf>.
- [6] Reddit. “Airbyte, snowflake, dbt and airflow still a decent stack for newbies?” (2025), [Online]. Available: https://www.reddit.com/r/dataengineering/comments/112qmw1/airbyte_snowflake_dbt_and_airflow_still_a_decent/.