



POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

**Automating Payroll Paper Generation
in the Healthcare Industry: A
Technical Solution for Hublo's
Customer Plants**

Supervisor

Prof. Paolo GARZA

Candidate

Francesca PAOLETTI

Internship Tutor

Jules Balland

ACADEMIC YEAR 2022-2023

Abstract

As the healthcare industry continues to evolve, automation has become increasingly important to improve the overall efficiency and productivity of businesses.

With this in mind, Hublo, a medical technology solutions provider, sought to automate the process of generating payroll papers for its customer factories. This procedure proved to be challenging since many various requirements and standards had to be considered while also ensuring efficacy and generalization.

The technical solution provided involves several steps, including data collecting and collation, data cleansing and construction, and combining plant features to automatically generate payroll papers. The project's ultimate goal was to provide a global solution that was not only efficient and adaptable, but also intuitive, answering the particular demands of individual clients while also meeting regulatory standards.

This project was a great achievement, with the technical solution proving to be extremely efficient and allowing Hublo to automate what was previously a laborious and time-consuming operation. Since the introduction of this solution, numerous additional clients have subscribed to the module, demonstrating its efficacy and adaptability. The automated pipeline is safe, efficient, and GDPR-compliant, increasing safety, productivity, and accuracy while decreasing the number of manual procedures necessary.

This project exemplifies the benefits of automation in the healthcare business and demonstrates the possibilities for future innovation in this field.

Acknowledgements

Come conclusione di questo elaborato e di questo lungo percorso, ci tengo a ringraziare dal profondo del cuore tutte le persone che anche solo con un sorriso, un messaggio, un abbraccio hanno condiviso con me qualche momento di questo meraviglioso cammino.

Ringrazio il mio relatore Paolo Garza, per la sua disponibilità e i preziosi consigli nella stesura di tale elaborato e tutto il Politecnico di Torino, che mi ha permesso di crescere professionalmente e personalmente, dandomi gli strumenti necessari per affrontare tutto ciò che verrà.

Grazie a Hublo, che mi ha accompagnato nella mia prima esperienza lavorativa con amicizia, rispetto e pazienza e in particolare a Jules, che ha creduto in me fin dal primo giorno e mi ha seguita e accompagnata lungo tutto questo percorso.

Un grazie, che neanche basterebbe, alla mia famiglia, per il supporto e il sostegno infinito, per avermi preso la mano nei momenti più duri, aiutandomi ad affrontare le sfide e celebrando con me ogni conquista, anche la più modesta.

Grazie agli amici di sempre, a chi ho incontrato durante il percorso all'università, condividendo sudori, fatiche, lunghissime ore in aula studio,

traguardi e successi, e chi ha condiviso con me una delle migliori esperienze che abbia mai vissuto in Francia. Grazie per essere stati dei compagni insostituibili, da vicino e lontano.

Table of Contents

List of Tables	8
List of Figures	9
Acronyms	11
1 Introduction	13
2 Context	17
3 Project	23
3.1 Company	23
3.1.1 Hublo Match	25
3.1.2 Tools	28
3.1.3 Database	30
3.2 Data collection	37
3.3 Data cleaning	42
3.3.1 Data cleaning process	43
3.4 Automation	45
3.5 Data Processing	47
3.5.1 Payroll codes and formats	49
3.5.2 Clients parameterization	51
3.5.3 Document generation	53

3.5.4 Process automation	55
4 Analysis of results	57
5 Conclusion	67
Bibliography	71

List of Tables

2.1	Percentage of absenteeism in French health care facilities. . .	19
3.1	List of basic payroll codes.	51

List of Figures

2.1	Statistics of the French medical sector.	18
2.2	Geographical distribution of healthcare workers in Europe with respect to the population.	20
3.1	Create a shift.	27
3.2	Select a candidate.	27
3.3	Generate and sign the contract.	27
3.4	Visualise dashboards.	27
3.5	Main steps of Hublo Match product.	27
3.6	Hubspot overview.	28
3.7	Planhat overview.	29
3.8	DBT overview.	30
3.9	Database system and input sources.	31
3.10	Data replication using Stitch.	32
3.11	Extract, Prepare, Load phases.	33
3.12	ERD.	36
3.13	RESTful API model.	38
3.14	Plan of the project.	40
3.15	Example of code to import external data in the database.	41
3.16	Documentation of HubSpot for the extraction of data.	42
3.17	Data Cleaning process.	43
3.18	Employees' work activities.	46

3.19 DBT Layers.	49
3.20 Data pipeline on DBT.	52
3.21 Project folder for document generation.	54
3.22 Heroku Scheduler.	56

Acronyms

HTTP

Hypertext Transfer Protocol

ETL

Extract, Transform, Load

API

Application Programming Interface

CRM

Customer Relationship Management

WHO

World Health Organization

FHF

Fédération hospitalière de France

Chapter 1

Introduction

In today's digital age, the importance of automation has been widely recognized across industries. Companies are increasingly looking for ways to automate their core activities to increase efficiency, productivity, and safety while minimizing the number of manual procedures required. This is particularly true in the healthcare industry, where technology has the potential to transform patient care and improve the day-to-day lives of medical professionals and where accuracy and timeliness are crucial for the well-being of patients.

Hublo is a scale-up that focuses on developing technology solutions for medical establishments. Its goal is to improve operators' day-to-day lives by promoting the use of technology and effective tools for carrying out numerous daily tasks. However, Hublo still faces challenges in automating some of its core activities, such as generating payroll papers for all operators working in customer plants. Currently, the process is manual, resulting in a significant loss of time and labor.

The purpose of this Master's thesis is to create a new technical method to streamline the process of generating payroll papers for Hublo's customer

plants, from data collection and collation to the development of an entire pipeline. The proposed approach entails gathering data from various platforms, transmitting all data sources to a single destination, cleaning and constructing new, clean, and usable data, and merging each plant's characteristics to generate the payroll papers automatically on a monthly basis. The difficulty in designing a template that is generic enough to match the diverse criteria, characteristics, and specifications of each customer is the reason for the lack of an automatic pipeline. Thus, the goal of this project is to create a flexible and adaptable global solution that is efficient and intuitive enough to allow for immediate modifications in the future.

In addition, this thesis considers the legal issues surrounding the documentation, which contains personal data of each facility's operators. This data must be secured and used in compliance with the existing GDPR legislation. Therefore, the project aims to draw up a secure and efficient pipeline for the automatic creation of such documentation, which includes studying the database, identifying the main tools to be integrated into the pipeline, understanding all the specifications to be taken into account, studying the different software used by the facilities, and identifying a secure and efficient procedure for automatically sending this documentation to the contact persons of each facility.

Overall, this thesis aims to provide a technical solution for automating payroll paper generation that will boost efficiency, safety, and minimize the number of manual procedures required, while also considering legal requirements and individual customer needs.

The document outlines the steps taken to complete the project, beginning with a detailed study of the company and the various input sources and tools, as well as the technical description of the proposed solution, an analysis of the results, advantages provided, and potential future developments.

The technical aspect of the project involves several steps to generate these documents automatically at the end of each month. The first step is data engineering, which involves collecting data from various sources and using ETL development to prepare it. The next step is data cleaning, which involves creating an efficient and usable work base for subsequent steps. The third step involves collecting the specifics of each institution, analyzing all possible variables that may be required, integrating this data with the previously processed information, and developing the code to generate the documents. The final step involves automating the code so that the generation of the document is automatic each time a new customer subscribes to the module, once the relevant parameters have been entered.

After this introduction, the following chapters are organized as follows. Chapter 2 offers an overview of the current state of the French and global healthcare systems, highlighting the challenges they are encountering and examining the potential benefits of automating certain internal processes. Chapter 3 describes in detail the methodology used in this project, including the data collection and cleaning procedures, the selection of tools and software, the technical architecture and implementation of the pipeline, and the evaluation of the results. Chapter 4 presents the results of the project, including a quantitative and qualitative analysis of the time and cost savings, the quality of the generated documents, and the feedback from the end-users. Chapter 5 discusses the implications and limitations of the project, the ethical and legal considerations, and the potential future developments of the pipeline and concludes the thesis by summarizing the main findings, the contributions to the field, and the recommendations for further research and practice.

Chapter 2

Context

In this chapter, we will describe in detail the context of the French and global health care systems, which have recently been plagued by absenteeism and recruitment difficulties, in order to understand the context of the project, which will be covered in the following chapter.

Indeed, the great shortage of healthcare workers is a reality throughout Europe and in the whole world.

The World Health Organization (WHO) predicts a shortage of 15 million health professionals by 2030 [1]. While low-income countries are particularly hard hit, all countries, regardless of development level, face difficulties in training, recruiting, and distributing their workforce. The trend has been known for several years, and the causes are well understood: the world's population is aging, which will necessitate more and more care, as will the rise in chronic diseases, while the health workforce is not being replenished sufficiently due to retirements, and training capacity is limited.

The World Health Organization (WHO) report, released in collaboration with the International Council of Nurses and Nursing, reveals that there are

now fewer than 28 million nurses worldwide [1]. Between 2013 and 2018, 4.7 million people joined their ranks, but there is still a 5.9 million-person global shortage, which is predicted to only grow by the years.

In France, the shortage of nurses, which increased significantly following the Covid-19 crisis due to numerous hours of overtime and deterioration of working conditions, can be explained by two main factors:

1. an ever increasing absenteeism,
2. recurrent recruitment problems.

In Figure 2.1, the statistics about exits and entries in the medical sector from 2015 and 2030, according to the European project EDJNet statistics [2].

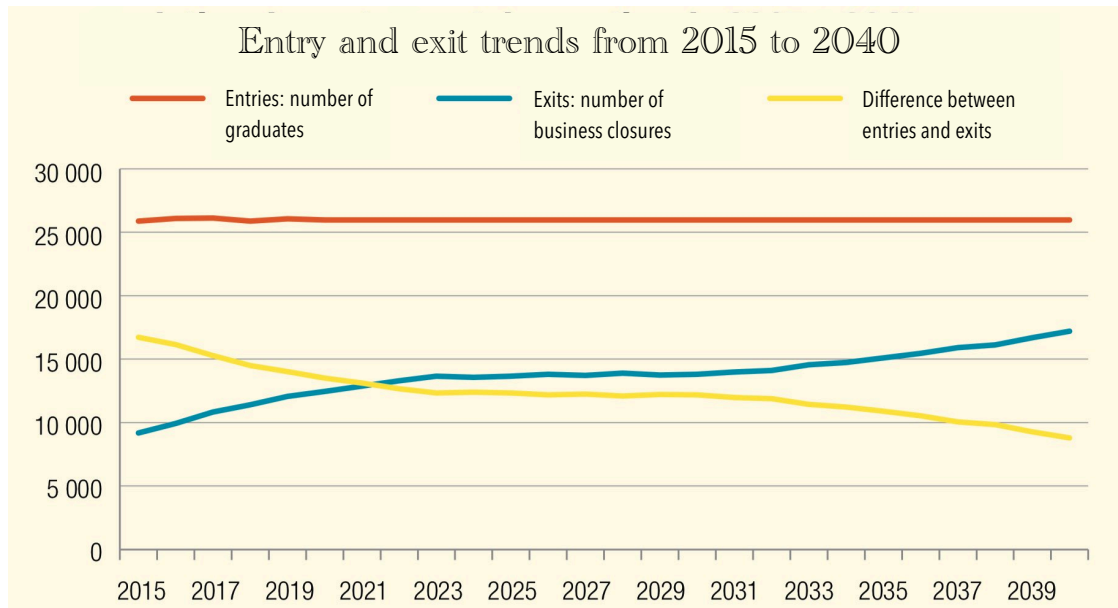


Figure 2.1: Statistics of the French medical sector.

As the graph shows, the gap between entries and exits in the French medical sector is expected to considerably grow over time. Hence, according to the FHF, Fédération hospitalière de France, the rate of absenteeism has

increased over the past year to an average of 10% [3]. The percentages of absenteeism for the previous three years are shown in the table below. From 2019 to 2021, there was an average increase of 1%, with a marginal increase in 2020.

Average absenteeism rate observed over the years	2019	2020	2021
Social or medico-social establishment (ESMS)	11.2%	12.1%	11.8%
Health care facilities	8.8%	10.1%	9.8%
University Hospital Cen- ter/Regional Hospital Center (CHU/CHR)	8.8%	9.7%	9.9%
Total	8.9%	10.0%	9.9%

Table 2.1: Percentage of absenteeism in French health care facilities.

As healthcare teams work on a just-in-time basis, as soon as one person is absent from the service, the additional workload falls directly on those present.

As far as recruitment problems are concerned, health care facilities are struggling to meet their needs. According to the FHF, health care institutions are facing an absenteeism of around 10%, and up to 5% of unfilled care positions in public hospitals and medical-social centers.

In Figure 2.2, the distribution of healthcare workers in Europe [4]. As previously stated and as easily noticeable in the figure below, the ratio of healthcare workers with respect to population in France, with the exception of major cities, is extremely low.

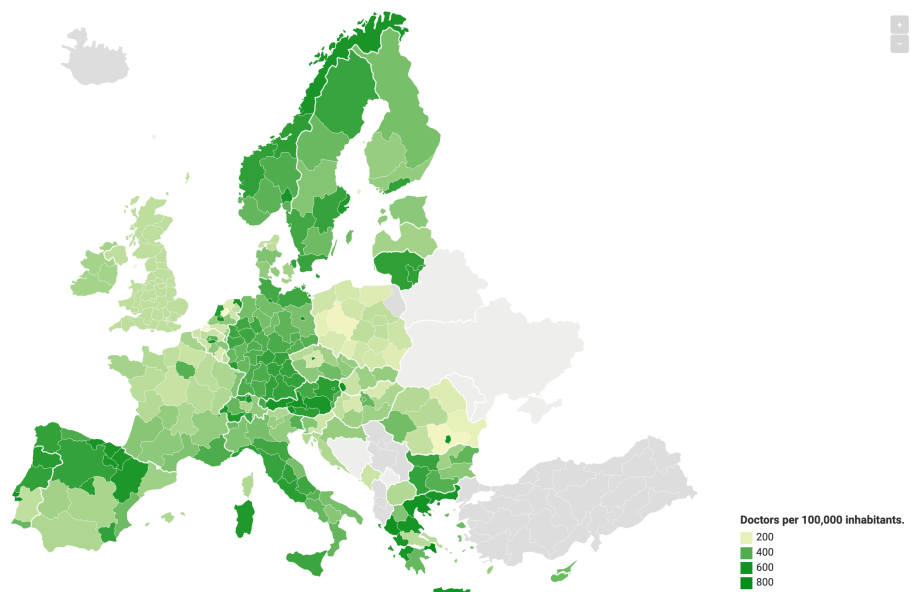


Figure 2.2: Geographical distribution of healthcare workers in Europe with respect to the population.

The primary effect of the nursing shortage is the closure of hospitalization beds in public health facilities. According to the DREES, Direction de la recherche des études, d'évaluation et des statistiques, the public sector had lost 5,758 hospitalization beds in a year by September 2021 [2]. Furthermore, according to FHF, 100000 positions go unfilled each year, and this number is expected to rise further in the aftermath of the pandemic. Indeed, health-care workers have been subjected to unprecedented physical and psychological strains, as well as being forced to work understaffed and many hours of overtime. For these reasons, 40% of health care workers declared their willingness to change jobs following the crisis [3].

Because absenteeism is expected to rise due to inactivity as well as recruitment difficulties, some actions must be planned to assist the medical sector in dealing with this massive shortage. The primary goals are to improve working conditions, particularly the quality of life at work and career management, to revalue salaries, to allow for career diversification, and to assist medical facilities in recruiting.

Hublo is a start-up which currently works in the French and German healthcare industries with the goal of addressing the severe lack of healthcare professionals by creating new technology.

The company aims to be able to positively and successfully improve the working circumstances for all medical personnel by showing them more respect for their way of life and their jobs, which in turn fosters better relations between them and the administrative staff. Its objective is to simultaneously make contributions in the area of recruitment, which is currently the second biggest problem for healthcare institutions, as explained before.

Hublo seeks to create technological solutions to address the aforementioned issues, aiming to increase the caliber of work performed by medical facility employees while also aiding the facility in attracting new hires by streamlining manual processes. For this reason, it has developed two primary products, *Hublo Match* and *Mstaff*. The former is a tool for managing staff replacement and absenteeism that makes use of a wider network of employees than the institution in question has access to. Therefore, the objective is to create a product that is straightforward, efficient, and simple to use while also saving a lot of time in administrative and personnel organization and making easier to handle uncovered tasks. The second product is a recruitment software, which aims to simplify the recruitment process by automatically conducting a selection phase based on uploaded documents and a match based on the institution's and candidates' requirements.

In this thesis, we will study, comprehend, and integrate modules from the first product, *Hublo Match*.

Chapter 3

Project

This chapter will begin by introducing the company, the main product, and the tools. Second, a detailed presentation of the database and the replication procedure will be provided, followed by a description of the project, beginning with its goal, moving on to the data collection and data cleaning process, and finally the implementation and automation details.

3.1 Company

Hublo is a medical-focused technology company that was founded as medGo in 2016 and later merged with Whoog in 2019. The startup's vision is to develop digital solutions that will allow caregivers to work in the best possible conditions. Hence, caregivers gain valuable time to focus on patient care by digitizing time-consuming manual processes and making them fast and intuitive.

As expressed in the previous chapter, staff management is one of the most difficult challenges for healthcare systems, such as scheduling, recruitment and retention tasks. Indeed, they must rethink how they attract, recruit, manage, motivate, involve and interact with their teams.

Therefore, Hublo's goal is to support all European health institutions by putting digital technology at the service of people.

In terms of product catalog, Hublo developed four main products, to face the major challenges of the healthcare system nowadays.

Hublo Match is the company's first and most important product, which was launched in 2016. It is a staff replacement management solution that allows healthcare providers to save time, cut recruitment costs, and improve employer-employee relationships.

The goal of this product is to minimize the need for a healthcare facility's HR manager or team leaders to make multiple phone calls in order to find a replacement [5].

Mstaff, the company's second flagship product, is a straightforward and collaborative HR technology created with hospitals to effectively build their employer brands and manage their hiring processes from posting job offers to validating candidates. Mstaff enables any business to create a completely unique web space. It enables the centralization of all applications with a dematerialized CV library, the automation of applicant response emails, and the scheduling of interviews. Additionally, it enables multiple posting of the offers on the most pertinent health employment boards to reach the greatest number of prospects.

Hublo Sourcing, the third product, is described as a tool for finding alternatives. Indeed, this solution increases the likelihood of finding profiles that meet each institution's search criteria and provides the establishment with access to a steady stream of nearby and readily available prospects. It allows temporary workers to be found based on facility requirements such as job, distance, location, service, and schedule. Businesses can use Hublo Sourcing

to find temporary staff that they can keep on board in order to plan for the future, reduce hiring costs, and lower employee turnover.

PlanBlanc is a concise and user-friendly platform that allows for quick staff communication and crisis management. After importing a file containing the staff database onto a secure server, it enables the creation of distribution groups in accordance with the various requirements. The facility can construct an alert message in a crisis and choose which groups it should be sent to. Additionally, it enables the communication of necessary adjustments at the appropriate times while allowing for the monitoring of mobilization progress using real-time dashboards. The facility can employ the statistics to adjust its communication strategy when the crisis has been resolved.

Hublo is currently used by over 500,000 healthcare professionals and the solution has managed 55 million replacement hours in over 2,500 healthcare facilities across France. Furthermore, in an analysis conducted by the business, 94% of clients claimed to have saved time by utilizing Hublo solutions, 93% have enhanced their employees' quality of life at work, and 90% have decreased the need of temporary labor. On a daily basis, more than 7000 missions are posted, with a fill rate of over 76%.

Concerning the international expansion, in 2020 Hublo opened a new section in Germany and signed the first clients. At the beginning of 2022, the company opened its horizons also in Spain and Netherland, with the goal of launching the product with the first establishments by the end of the year.

3.1.1 Hublo Match

Before diving into the project, it is necessary to understand the functionalities of the company's main product, Hublo Match, in order to better understand

the project's goal and utility.

As explained in the section above, Hublo Match is a replacement management solution, whose clients are medical institutions. This product is used by the healthcare institutions' managers to handle the whole network of workers, with the aim of fighting absenteeism and reducing the overall administrative time spent to call workers and find replacements.

When an institution subscribes to the product, the first step is to distribute the establishment codes to employees and temporary workers so that they can connect to the network via the application. Once their integration is validated, as well as the jobs and skills they fill, they can receive notifications from the establishment and be notified if a replacement is required.

Depending on the configuration, the institution can publish an offer, and the network will be notified via the application or via SMS. Each employee can decide to apply for the mission by simply clicking on the notification.

Hublo notifies the selected temporary employee and sends a reminder the day before the mission, once the recruiter has made a selection among all the applications received.

Among all the services provided by Hublo Match, some additional modules can be added to the base offer. The first option is the possibility of automatically generate the contract to be signed by the worker, i.e. the person is a temporary worker and not an employee of the institution, based on the institutions' specifications, which is automatically sent to the healthcare professional chosen for the replacement, who just needs to signed it electronically. Moreover, in order to increase the likelihood of finding competent and available professionals, if the facility is part of or forms a cluster, it is possible to establish a communal replacement staff network. The third options consists in the data module, which allows each establishment to consult several dashboards at any time, in order to easily track the evolution of the network

and the activity on the platform. It also provides detailed monthly reports to forecast replacement needs and manage absenteeism. Lastly, Hublo Match offers full integration with the softwares used by each establishment: the HR software to always have an up-to-date version of employees' documents, the payroll software to handle workers' monthly payments, and the ATM software to directly view in Hublo the caregivers' schedules and track the number of hours already completed.

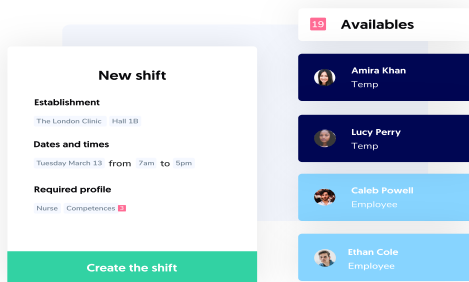


Figure 3.1: Create a shift.

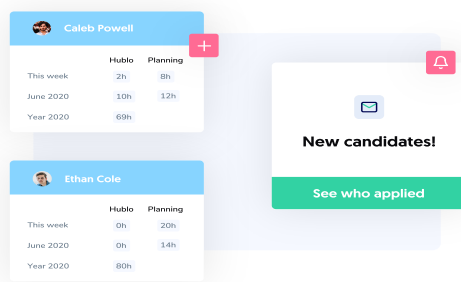


Figure 3.2: Select a candidate.

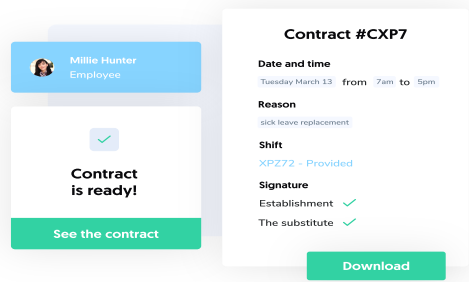


Figure 3.3: Generate and sign the contract.

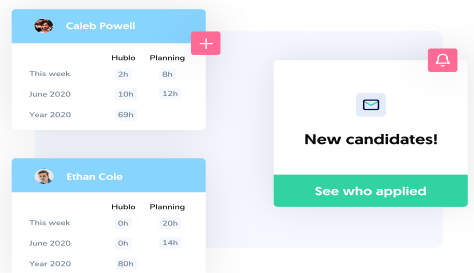


Figure 3.4: Visualise dashboards.

Figure 3.5: Main steps of Hublo Match product.

3.1.2 Tools

This section describes in detail the main toolset for storing, analyzing, customizing, and exporting data.

HubSpot CRM is an inbound marketing software platform used by the sales team to record establishment information, contract details and stage, and all necessary follow-up steps. It allows to get a real-time view of the sales pipeline on a visual dashboard, to access detailed reports on sales activity, productivity and performance (Figure 3.6). Indeed, CRM stands for “customer relationship management.” Customer relationship management software is a powerful tool that helps businesses organize and manage their customer relationships on a centralized and easy-to-use platform. By tracking leads and building a full database of customer activity, businesses have clear insight into where they stand with each customer in the buying process.

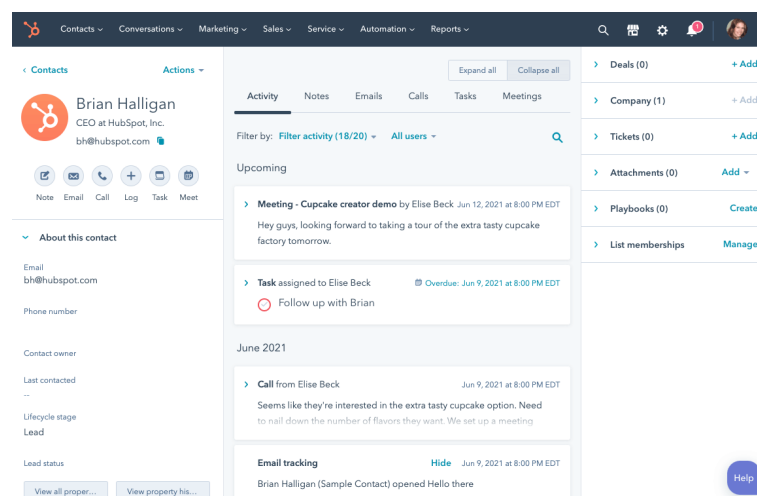


Figure 3.6: Hubspot overview.

Planhat is a client platform built horizontally for organizations to manage and grow their customers, whether for customer success, product growth and channel management (Figure 3.7). After the sales team signs a contract with

an institution, the Customer Success team handles follow-up, training, and parameterization of the platform’s use based on the institution’s specificities and needs. The team can save all useful information, activities, and customer experience on this platform.



COMPANIES	END USERS	PLAYBOOKS	OPPORTUNITIES	LICENSES	SALES	INVOICES	CHURNS	PROJECTS	ASSETS	ISSUES
HLTH	COMPANY	ARR (USD)	PHASE	CSM SCORE	RENEWAL (DATE)	LAST TOUCH (DATE)	LAST SEEN (TIME AGO)			
R & F	575k	Renewal	●●●●●	Mar 01, 2023	May 04, 2022	-				
Zithel's Zippers	500k	Adoption	●●●●●	Jun 03, 2022	May 31, 2022	-				
Daimler Group	200.2k	Onboarding	●●●●●	Jun 25, 2022	May 07, 2022	4 hours ago				
Reliance Bank	162k	Adoption	●●●●●	Oct 16, 2022	Feb 01, 2023	-				
Apple	159k	Success	●●●●●	May 07, 2023	May 29, 2022	an hour ago				
MongoDB	155.1k	Adoption	●●●●●	Jul 25, 2022	Feb 06, 2022	4 years ago				
Concur	147k	Adoption	●●●●●	Mar 10, 2023	Feb 06, 2022	4 years ago				
Zendesk	140k	Success	●●●●●	Feb 14, 2024	May 02, 2022	a year ago				
Tesco	125.1k	Adoption	●●●●●	Jun 27, 2022	May 07, 2022	2 hours ago				
ITC Ltd.	117.8k	Adoption	●●●●●	Jan 31, 2023	-	-				
United Oil & Gas, UK	112k	Success	●●●●●	Jul 06, 2022	May 02, 2022	9 hours ago				
Daimler APAC	110k	Success	●●●●●	Oct 18, 2022	May 07, 2022	an hour ago				
Toyota	108k	Success	●●●●●	May 16, 2022	Jun 03, 2022	an hour ago				

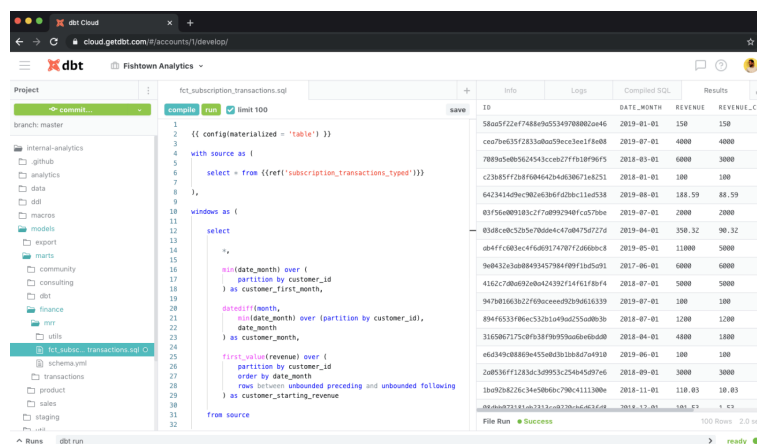
Figure 3.7: Planhat overview.

DBT is a transformation workflow that allows teams to quickly and collaboratively deploy analysis code by following software engineering best practices such as modularity, portability, CI/CD, and documentation (Figure 3.8). Once the data has been extracted from the tools, DBT allows to proceed with the data processing phase, making the raw data usable, cleaner, and more accessible. DBT also enables the creation of new transformations from existing data, which can then be used to develop any study or analysis for the customer or the company itself, which are called transformation layers.

More in details, transformation layers come in three different categories:

- A clearing layer: it is specifically used to format a table’s columns. For instance, we frequently need to convert dates that we have from CRM software with Singer because they are frequently in the unix13 format. Cleaning is also used, for instance, to standardize data in a single column or to get rid of rows that are inconsistent.

- A computation layer is used to derive useful indicators from unremarkable or incomparable raw data. You might want to change an absolute figure into a relative rate, for instance.
- A data reorganization layer is used to show a table in a way that is easy to understand for a user who is unfamiliar with databases.



The screenshot shows the DBT Cloud interface for a project named 'FishTown Analytics'. The left sidebar displays a file explorer with folders like 'internal-analytics', 'data', 'dbt', 'macros', 'models', 'reports', 'community', 'consulting', 'dbt', 'finance', 'mer', 'utils', and 'schemas'. The main area shows a SQL query in the 'fct_subscription_transactions.sql' file. The query is a window function that calculates the first revenue for each customer by month. The right sidebar shows the 'Results' tab with a table of 100 rows and 5 columns: ID, DATE_MONTH, REVENUE, REVENUE_CHA, and a column for the first revenue. The table contains 100 rows of data, with the first row having ID '5ba05f22a774848e553497080020e46' and REVENUE '150'.

ID	DATE_MONTH	REVENUE	REVENUE_CHA	
5ba05f22a774848e553497080020e46	2019-01-01	150	150	
c8a70e63f72535a0d59e3a3e1f8a08	2019-07-01	4000	4000	
70894e5e05624543cc0e277f5b8796f5	2018-03-01	6000	3000	
c23b85ff2b8f084642b4d830c71a8251	2018-01-01	100	100	
6423414d9e9c9024e3b6f4d20c11a0538	2019-08-01	188.59	88.59	
03f56e009183c2f7a099239d8fcd570be	2019-07-01	2000	2000	
03d8c0bc5295e708d0e4c47d0475d7276	2019-04-01	350.32	90.32	
0b4ff603ec4f6a05174707f2a660bc8	2019-05-01	11000	5000	
9a0432e3a0808493457884f09f13d5u01	2017-06-01	6000	6000	
4362c7d0d832a0a424392f14f61f0f4	2018-07-01	5000	5000	
9d7081663e22f0a0e0e0d29a6161339	2019-07-01	100	100	
09af653f06e53201e93a0255a00b30	2018-07-01	1200	1200	
3165067175c09b38f90950a0b0e0d00	2018-04-01	4800	1800	
e6d349c08809e455a030120a0d7a910	2019-06-01	100	100	
2a0536ff1283d4c3d9953c254045d07e6	2018-09-01	3000	3000	
1ba9208262c34e50840c790c4111300a	2018-11-01	110.03	10.03	
00d0b0731e1a1331a0b370b0e4e36e08	2018-12-01	100.07	1.07	

Figure 3.8: DBT overview.

Finally, once the data is clean and accessible, all of the work of study and analysis, as well as the development of indicators, is completed using Metabase, an open-source tool used to facilitate data exploration. Metabase allows you to connect to a database automatically, extract the information it contains, and perform all necessary analyses using the SQL language.

3.1.3 Database

The database in use is a PostgreSQL database, a relational database that gathers information from many sources (Figure 3.9): the four products and the two software platforms HubSpot and Planhat, from which information is pulled via ETL.

In order to ensure an higher level of availability, data are replicated to the cloud using an open-source tool, Stitch.

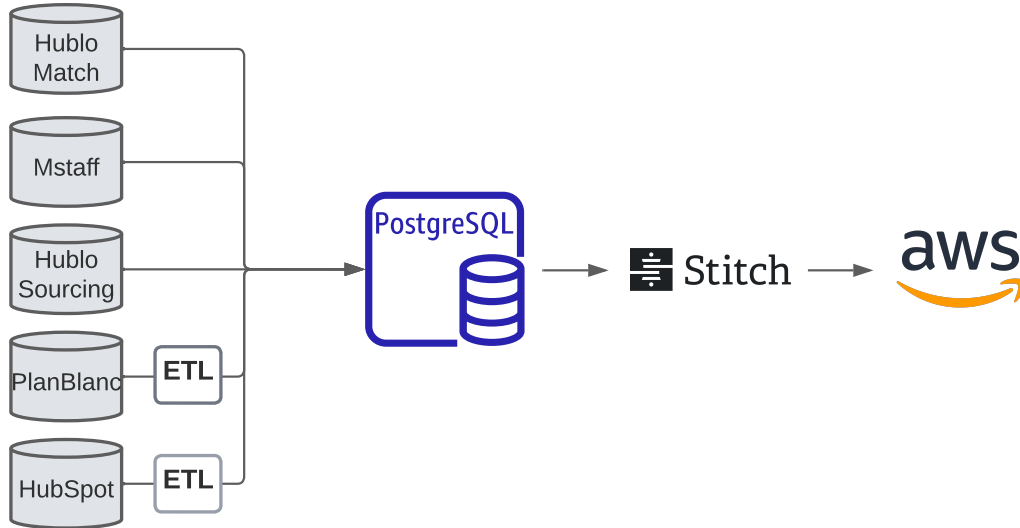


Figure 3.9: Database system and input sources.

Data Replication

Stitch, a cloud-based ETL data pipeline, is used to replicate all data in the database. Data replication ensures increased data availability and reliability, improved data backup, and faster data access. Indeed, if a system at one site fails due to hardware failure or other issues, users can access data stored at other nodes. Second, because data is replicated across multiple sites, it is simpler to recover deleted or corrupted data. Finally, because data is stored in multiple locations, users can retrieve data from the closest servers, resulting in reduced latency. Furthermore, because data can be retrieved from multiple servers, there is a much lower chance that any one server will become overburdened with user queries.

For small companies that can not guarantee a full-time cybersecurity staff,

like Hublo, it's safer to replicate data to the cloud.

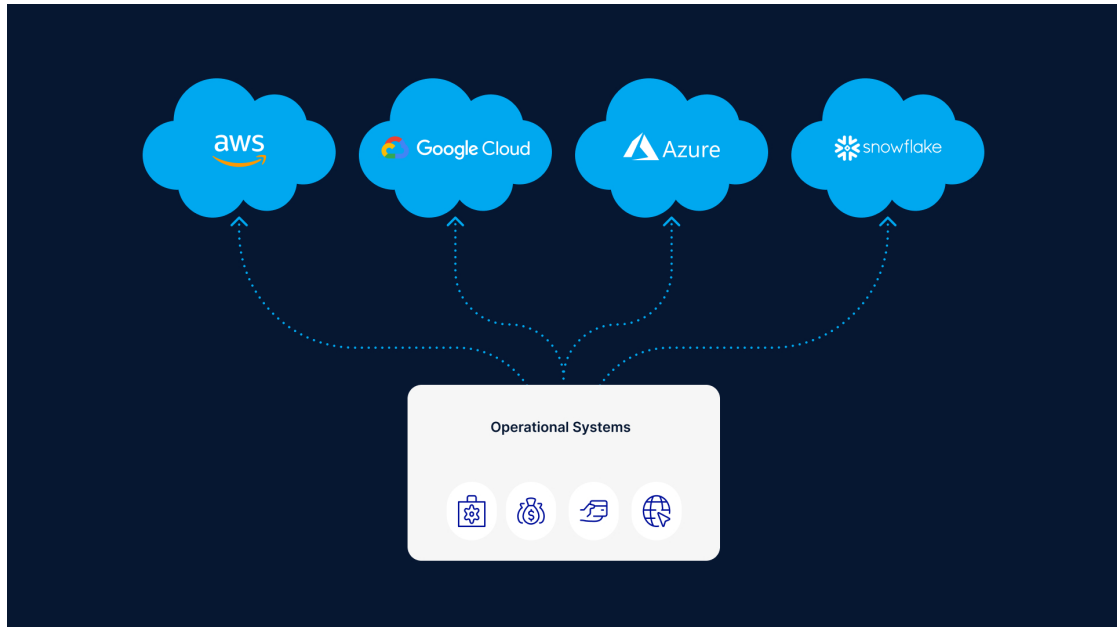


Figure 3.10: Data replication using Stitch.

The data replication strategy chosen is crucial as it impacts how and when data is loaded from source to replica and how long it takes.

Stitch is a cloud-based, ETL data pipeline. ETL stands for extract, transform, load, which are the steps in a process that moves data from a source to a destination.

Stitch's replication process consists of three distinct phases:

- Extract: Stitch pulls data from data sources and persists it to Stitch's data pipeline through the Import API.
- Prepare: Data is lightly transformed to ensure compatibility with the destination.
- Load: Stitch loads the data into your destination.

Stitch Internal Architecture

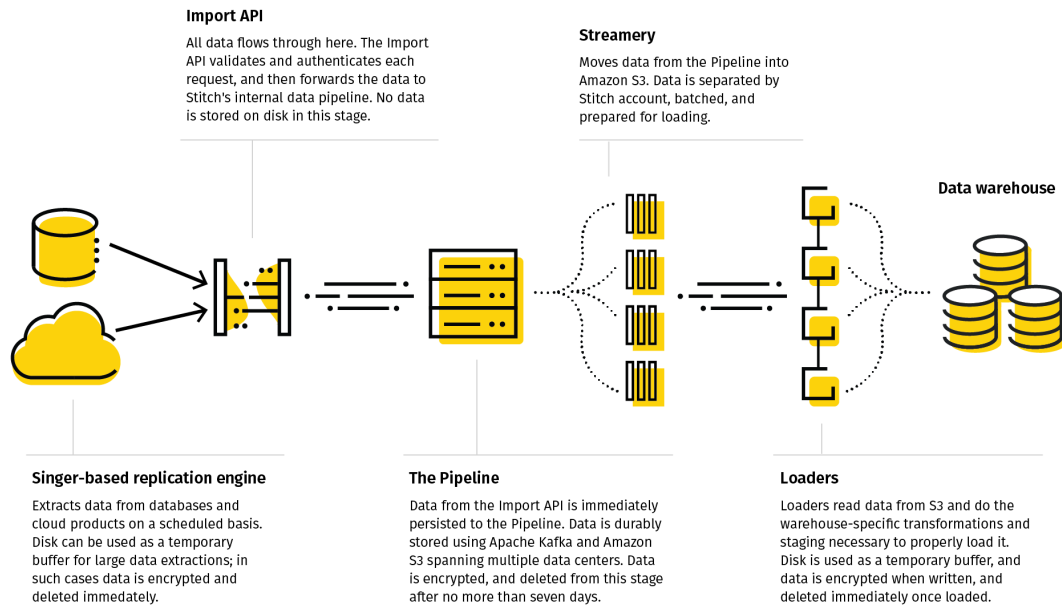


Figure 3.11: Extract, Prepare, Load phases.

The architecture is presented in Figure 3.11, while the functioning of the system is described below.

During the Extract phase, the Singer-based replication engine and the Import API are used. The first module allows to pull data on a schedule. At first all changes in tables and columns are detected and next, data are extracted based on the defined schema and sent to the Import API, which validates and authenticates each request. The Prepare phase consists in buffering data to the internal pipeline of Stitch and prepare it for the processing phase. The last phase in the replication process is called Load, in which the prepared data is transformed to be compatible with the destination, and then loaded.

The process to load data is performed in steps, meaning that the replication

is not done in real-time and there will be some delay between data extracted and data loaded.

Data of Hublo are replicated on AWS cloud, one of the major cloud services.

Entity Relationship Diagram

In order to dig into the project, it is interesting to understand the logical structure of the database and the relationship of entity sets stored. The Entity Relationship Diagram, also known as ERD, is a diagram that allows to logically describe a database, through three main concepts: entities, attributes, and relationships.

Figure 3.12 displays the ERD of the Hublo Database, which has a fairly complex structure. Because the database contains all of the product information, the number of tables is enormous, and the amount of information contained is extremely deep and difficult to comprehend. Almost all of the tables are related to one another, making understanding the database even more difficult.

In addition to the core database tables, there are multiple layers of transformation built with DBT that enable analysis and extraction tables to be created from various tools, which will be further explained in the next chapter. Each team member is permitted to construct layers based on their particular project, with the aim of generating layers that facilitate analysis by directly querying a table that contains all of the necessary data. This way, a significant amount of queries are exported to clients in order to provide them with real-time data on how the platforms are being used. This limits the number of tables that need to be accessed, allowing for an acceptable turnaround time for the query results.

However, if the number of tables to be accessed is extensive, the number of joins required may increase significantly, leading to an extended time required to receive the query results. In such cases, this may not be a suitable solution

for the client.

In the next chapters, we will look at the transformation layers that were constructed to help develop the project and speed up the analysis, beginning with the fundamental tables displayed in the ERD in [3.12](#).

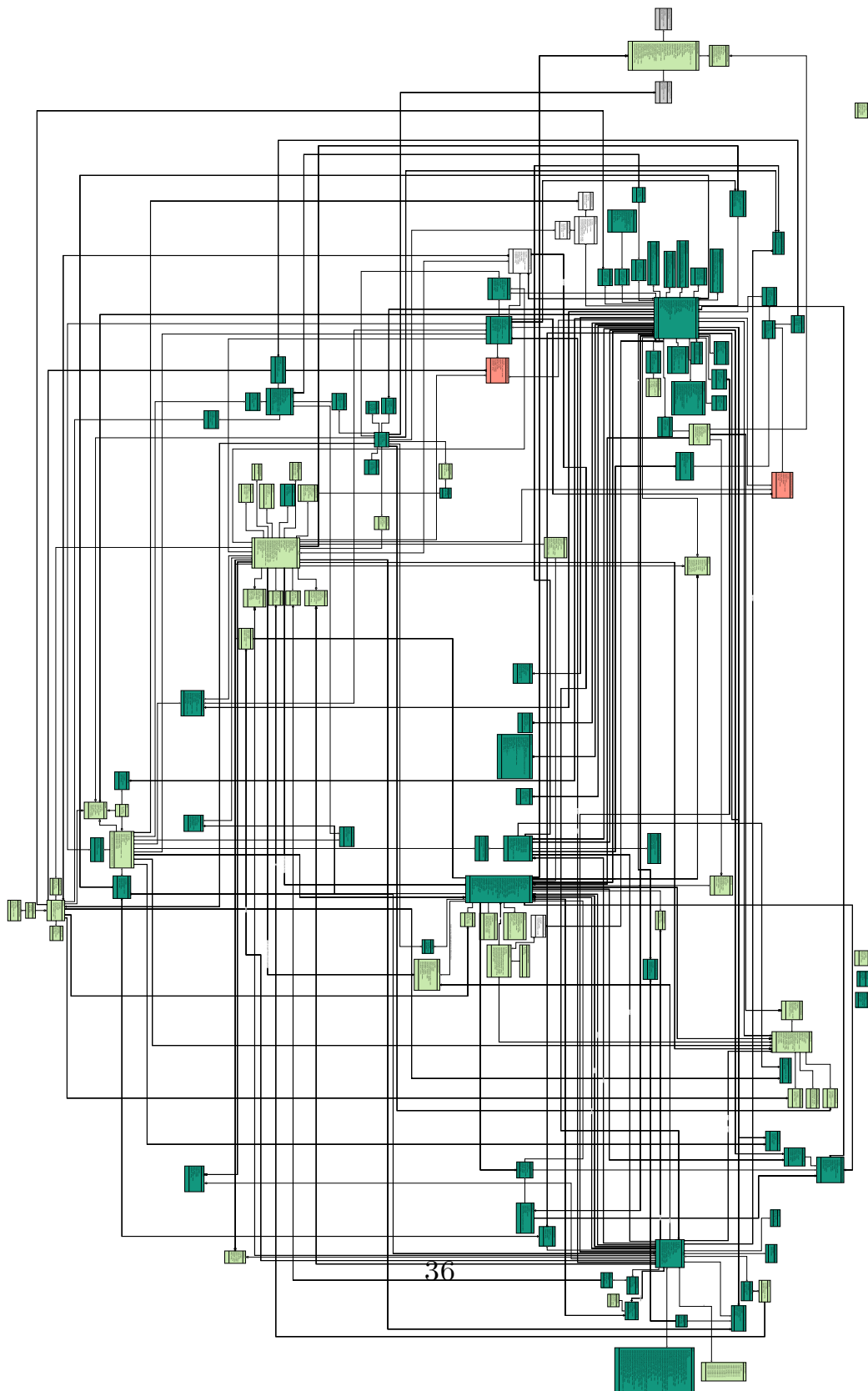


Figure 3.12: ERD.

3.2 Data collection

Data pipelines play a crucial role in all kinds of data platforms, be it for Predictive Analytics or Business Intelligence or maybe just for ETL between various heterogeneous data stores. They all rely on real-time or batch ingestion of data which is further processed to derive insights and make predictions or to just summarize and reshape the data for reports.

Data pipelines can be used to synchronize data between heterogeneous data stores, to move data from a staging area to production systems after data cleaning, reshaping and summarizing and also to distribute segmented data across various sub-systems from a central data source.

The key features of a data pipeline are:

- data frequency, which is the speed at which the destination systems require the data. The pipeline should be capable enough to maintain the frequency of data transfer required by the destination system.
- resiliency, which refers to how fault tolerant and resilient is the data pipeline.
- scalability, which refers to the fact that the tools and technology used in the pipeline must possess the capability of re-configuring it to scale out onto more hardware nodes if the data load increases.

The data pipeline built exploits an open source ETL tool, *Singer*, to fetch employee records from a REST API and insert them into a PostgreSQL database table.

An Application Programming Interface plays a significant role in facilitating the communication between products and services. APIs comprise a significant part of business revenues.

RESTful APIs are a type of Web APIs, which provide an interface for web

applications that need to connect each other via the Internet. APIs play a significant role in enhancing the agility, flexibility, and speed of a network [6]. A web service provides access to its services via URL or URI on the World Wide Web. The key feature is the format of how it presents the information to other applications, which made understanding easy. A web service uses HTTP/HTTPS to exchange information. The application sends an HTTP request where the required information is passed along with the URL as a path and then the web service sends an HTTPS response.

A REST API benefits from existing protocols and it takes advantage of HTTP. Since data is independent of its methods and resources, REST provides flexibility by handling different type of calls, different data formats, and dynamic structures. REST does not need any library support, it returns data without exposing methods and supports any content type but it's stateless, meaning that it does not have the capability to maintain the state such as sessions. Figure 3.13 shows the model of the RESTful API.

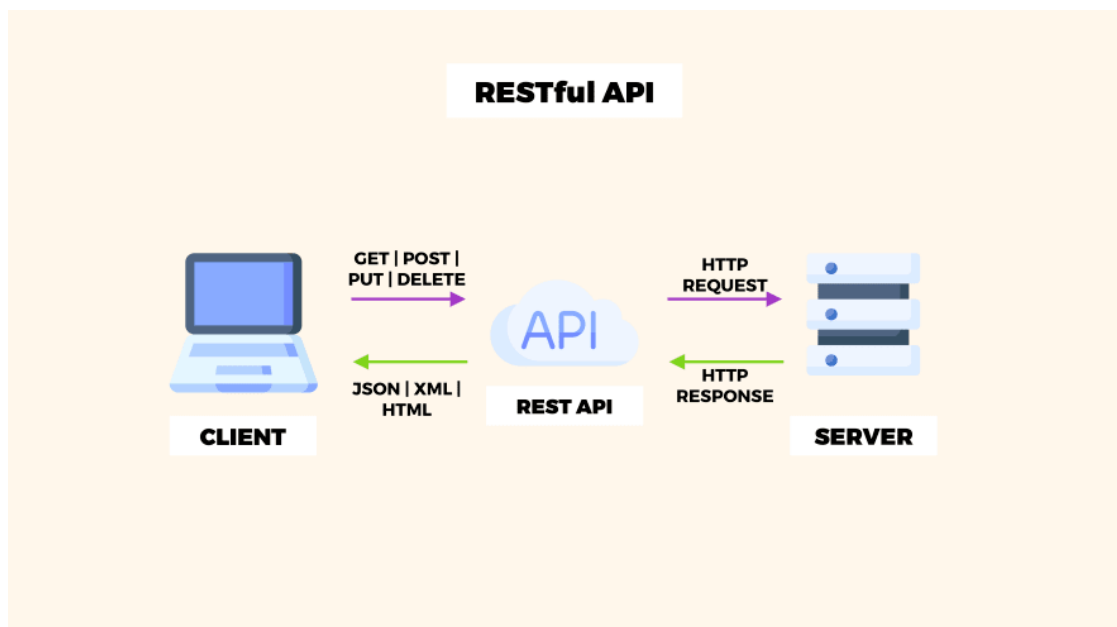


Figure 3.13: RESTful API model.

Singer allows creating a data pipeline, fetching data from a REST API and sending them to a specific destination, by facing two steps: *Tap* and *Target*.

A *Tap* is an application that takes a configuration file and an optional state file as input and produces an ordered stream of the record, state, and schema messages as output. A record is JSON-encoded data of any kind. A state message is used to persist information between invocations of a *Tap*. A schema message describes the datatypes of the records in the stream. A *Tap* may be implemented in any programming language.

The *Target* phase consists in reading lines from the standard input and processing schema, record and state messages. It performs a schema validation to ensure that records are written in a conform schema based on their format, it writes state messages once all data that appeared in the stream before the state message has been processed by the *Target*.

This process is designed to gather data from various sources and extract all the necessary information to build the pipeline. The collected information is then sent to the PostgreSQL database.

Figure 3.20 shows the major steps for the completion of the project.

The first stage consists in the extraction of raw data, using the described process of *Tap* & *Target*. The first input source is Hubspot, the CRM tool used by the Sales team, which stores and collects information about each client, including company specifications, location, size, and deal-related details.

The second input source is Planhat, a CRM platform used by the customer success team to track customer usage and follow-up information.

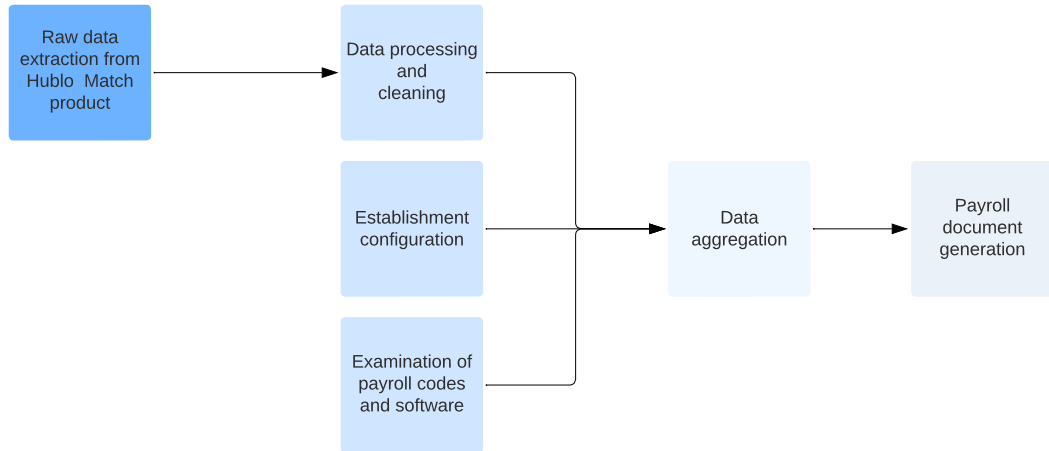


Figure 3.14: Plan of the project.

The third and final input source is a Google Sheet document created to capture the specific details of each institution. This information must be considered in the generation of documents, such as required pay codes and hourly rates for each job.

For each input source, a Python file is generated using the principles of the *Singer* tool. An output schema is defined for each input source to extract all the data and send it to the PostgreSQL database. This approach ensures that all data is available in the database and can be analyzed, cleaned, and integrated to build the pipeline.

In Figure 3.15 an example of import of data from an external source. The script is used to extract the information about the client companies from *Hubspot* and send them to the table *companies* in the database.

The main libraries used to perform this operation are:

- the *psycopg2* library, to install the connection with the database;

- the *requests* library, to extract the response from the url and fetch the extracted data;
- the *Singer* library, to create the table, with a specified schema and send the record fetched from the url to the database.

```
import Singer
import requests
import datetime
import psycopg2
from psycopg2 import Error
import utils

try:
    # Connect to the PostgreSQL database, using the json file credentials stocked in utils
    connection = psycopg2.connect(utils.credentials)
except (Exception, Error) as error:
    print("Error while connecting to PostgreSQL", error)

url = "https://api.hubapi.com/companies/v2/companies/18479339"
limit = 100

hapikey = 'demo'

schema = {'properties': {
    'company_id': {'type': 'integer'},
    'company_name': {'type': 'string'},
    'country': {'type': 'string'},
    'isDeleted': {'type': 'boolean'},
    'isClient': {'type': 'boolean'},
    'start_date': {'type': 'string'}
}}

table_name = 'companies'
singer.write_schema(table_name, schema, 'company_id')

def create_companies(offset, hapikey, limit):
    response = requests.get("https://api.hubapi.com/companies/v2/companies/18479339&limit={}&offset={}&hapikey={}", limit, offset, hapikey)
    offset = response['end']['offset']
    for row in response:
        record = dict.fromkeys(schema)
        record['company_id'] = row['companyId']
        record['company_name'] = row['name']['value']
        record['country'] = row['country']['value']
        record['isDeleted'] = row['isDeleted']
        record['isClient'] = row['isClient']
        utc_time = datetime.datetime.fromtimestamp(row['createdate']['value'])
        record['start_date'] = utc_time.strftime("%Y-%m-%d %H:%M:%S.%f+00:00 (UTC)")
        singer.write_record(table_name, record)
    return offset

while offset is not null:
    offset = create_companies(offset, hapikey, limit)

connection.close()
```

Figure 3.15: Example of code to import external data in the database.

In order to implement the script and extract data from a specific source, it is necessary to well understand the documentation, since the parameters configuration may change within each tool and the code needs to be adapted to it, i.e. the hapikey in Figure 3.15 is set to *demo*, since it's a personal information of the company.

In Figure 3.16, the documentation of *HubSpot* for the extraction of a company, with the URL and the specification of the given response.

```

Example GET URL:
https://api.hubapi.com/companies/v2/companies/10444744?hapikey=demo

Example response:
{
  "portalId": 62515,
  "companyId": 10444744,
  "isDeleted": false,
  "properties": {
    "description": {
      "value": "A far better description than before",
      "timestamp": 1403218621658,
      "source": "API",
      "sourceId": null,
      "versions": [
        {
          "name": "description",
          "value": "A far better description than before",
          "timestamp": 1403218621658,
          "source": "API",
          "sourceVid": [
            ]
        }
      ]
    },
    "name": {
      "value": "A company name",
      "timestamp": 1403217668394,
      "source": "API",
      "sourceId": null,
      "versions": [
        {
          "name": "name",
          "value": "A company name",
          "timestamp": 1403217668394,
          "source": "API",
          "sourceVid": [
            ]
        }
      ]
    },
    "createdate": {
      "value": "1403217668394",
      "timestamp": 1403217668394,
      "source": "API",
      "sourceId": null,
      "versions": [
        {
          "name": "createdate",
          "value": "1403217668394",
          "timestamp": 1403217668394,
          "source": "API",
          "sourceVid": [
            ]
        }
      ]
    }
  ]
}

```

Figure 3.16: Documentation of HubSpot for the extraction of data.

3.3 Data cleaning

Once all data are accessible in the database, a data cleaning pipeline is built in order to provide data consistency, reliability and usability. Since data is the core of the company, it highly influences every business decision and the accuracy of each provided result. Commonly, data collected from the various

resources are usually dirty, so the process of data cleaning is necessary in order to offer a better data quality which allows to make sure to have data ready for the analyzing phase. This process mainly consists of identifying the errors, detecting the errors and correcting them.

3.3.1 Data cleaning process

Data cleaning process consist of five phases, data analysis, definition of transformation workflow and mapping rule, verification, transformation and backflow of cleaned data. Figure 3.17 shows the data cleaning process, for further details check [7] and [8].

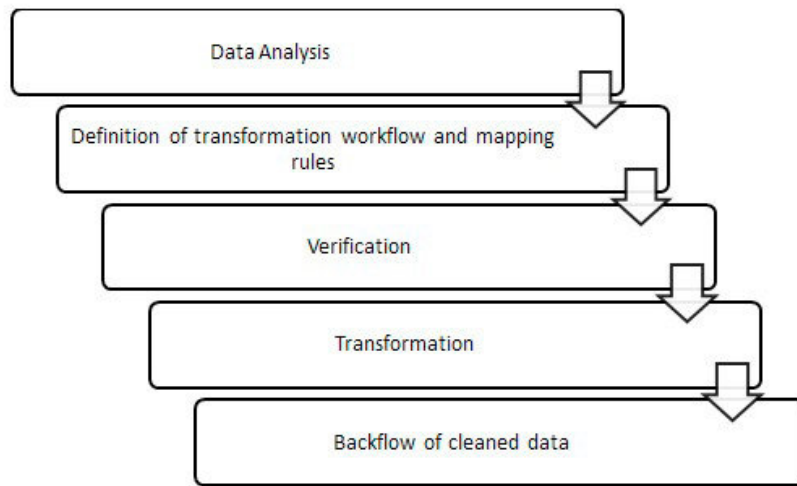


Figure 3.17: Data Cleaning process.

The first step consists in analyzing data to identify the errors and inconsistencies, by adopting two approaches: data profiling and data mining. Data profiling focuses on the instance analysis of individual attributes, while data mining on discovering the specific data pattern in the large dataset. Next, transformation workflow defines the detection and elimination of anomalies, by performing a sequence of operations on data.

The third step is the verification stage, in which the correctness and effectiveness of the transformations are evaluated.

Afterwards, transformations are applied to data and finally, after all the errors have been removed, the dirty data must be replaced with the cleaned data.

Following this process, after the extraction and collection of data from the different input sources, a data cleaning pipeline is built, in order to provide reliability and to ensure quality for further analysis.

The data used consists of information on all the missions carried out by workers in a healthcare facility, with details on the day, time, and associated information. To make the data clean and usable, the following data transformations were applied:

- Normalize date formats.

The recommended format is the ISO standard for working with multiple time zones. As Hublo's clients are spread between France and Germany, and possibly in other countries in the future, it is necessary to standardize all the data using the same format.

- Standardize time formats.

All time information was cleaned to be standardized in the ISO standard to avoid differences with daylight saving time (which can cause an incorrect number of hours worked and also incorrect information on a worker's total salary).

- Standardize institution information.

Each institution is associated with an identifier in the database, but each institution can be divided into several different sites where different missions take place. As the higher-level institution is the one responsible for remuneration, it is important that all missions are grouped and

associated with the same identifier to avoid problems during calculation and subsequent implementation steps.

- Errors and missing values check.

As the majority of information entered into the products is input manually, it is important to perform a thorough check for errors or missing data. To identify writing errors, the data were grouped by ID and a comprehensive review was conducted, followed by any necessary corrections. Additionally, to address missing information for certain missions, the necessary values for analysis were completed by replacing them with correct information from other rows in the table. This was done in order to establish a comprehensive and complete foundation for analysis.

3.4 Automation

In this section, we will elaborate on the benefits that administrative automation can bring to healthcare facilities and healthcare workers. By improving productivity and labor efficiency, product quality, and working conditions, administrative automation can significantly enhance the quality of business processes within the enterprise.

The main problem of modern enterprises is the lack of automation, especially in the medical sector. A large percentage of routine operations, which are manual input, search, data analysis become a source of errors, loss of time and a decrease in employee's motivation. Information technology implementation and the development of specialized programs and modules are the major approaches for addressing the issues associated with automating business processes. By using such applications in various functional units, the enterprise is able to make the information that has amassed clear, accessible, and organised while also decreasing the amount of human factor and needless paperwork [9].

Administrative automation or office automation consists in automatise all the administrative tasks in an organization, by improving efficiency and productivity. Indeed, administrative work is time-consuming, it is a tedious and repetitive work, in which human errors are guaranteed, the workload is continuously increasing and it causes a decreasing in employees' productivity.

According to Anatomy Work Index [10], employees spend on average 60% of their time on administrative tasks, as shown in Figure 3.18. The majority of global employees are still putting in long hours to manage workloads and execute on deliverables, often at the expense of true productivity gains. As a result, just 27% of time is dedicated to the skilled craft that employees have been trained and hired to do. The remaining 13% of time is dedicated to strategic planning and forward-looking analysis.

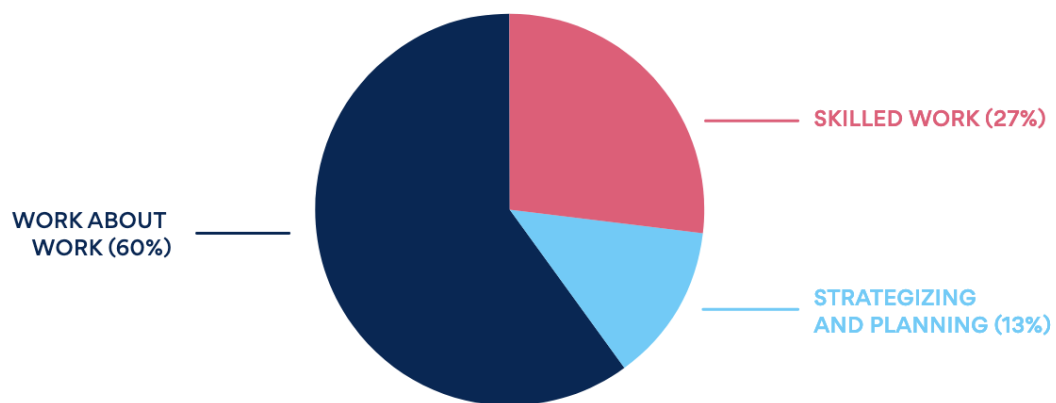


Figure 3.18: Employees' work activities.

Hence, administrative automation is the process of using technology to automate administrative tasks, in order to reduce the percentage spent in *Work about work* tasks, as explained in [10]. In simple terms, automation is the action of automatically fulfilling repetitive and monotonous tasks.

The main objective of this project is the administrative automation, by automatising the task of capturing data from invoices. Manual invoice capturing procedures are tedious and repetitive. Not only does it make employees a lot less productive, but it also invites expensive errors, which have enormous consequences on employees.

Therefore, the objective is to develop a pipeline that intends to automate administrative operations connected to employee payments for all clients who subscribe to the proposed module, utilizing the data gathered via the *Hublo Match* product.

Indeed, *Hublo Match* enables the client institution to assign a mission to a specific person within its network. As a result, at the end of each month, the personnel in charge of this procedure at each facility, must manually enter all data on missions performed by their employees into the payroll software in use. As previously stated, this takes many hours and significantly reduces employee productivity. As a result, the goal of this project is to create a pipeline that will generate a monthly payroll document for each institution, tailoring the format and specifications to the constraints of each.

3.5 Data Processing

After completing the steps of data collection and cleaning, the goal is to construct a database that enables rapid and efficient retrieval of the desired data. To achieve this, multiple layers were produced utilizing the DBT framework and implementing the SQL and Jira programming languages. To generate such documentation, it is important to recognize and understand the format for extracting missions carried out by healthcare workers. Each

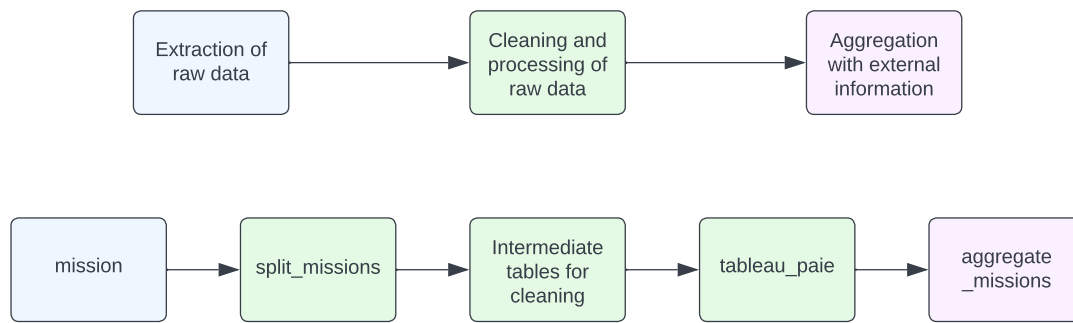
mission is recorded by saving various attributes, including start and end dates, working hours, breaks taken during that period, the operator who carried out the mission and their profession, the facility, department and service in which it was completed. Since each time period has a different remuneration system, which also varies based on the facility and profession, it is necessary to split each mission according to the pay code associated with each period.

Here is a list of all the transformations applied to data:

- If the mission lasts more than a day, it must be divided to have the details of each day to calculate the hourly totals correctly. For this reason, `split_mission` table contains a line for each day and multiple lines for the same mission. In this case, if the mission lasts more than one day, for the first day, the date and start time are those in the database, but the end time is set to midnight, and so on for each day.
- Each mission must be classified according to the day and time of day it was performed. It is, therefore, necessary to define the day of the week and time slot since hourly rates differ based on this information. This was done using `DAYOFWEEK()` SQL functions, able to give the day of the week associated to a specific date.
- It was also necessary to retrieve all data associated with the mission and aggregate it to provide complete details to the organization. Additional information is stored in the `tableau_paie` table, such as details about the replacement, service, mission pole, profession, and specialties. The entire dataset is saved in the final table `aggregate_missions`.

Figure 3.19 depicts all the tables that were generated to construct the final database, which will serve as the foundation for generating the relevant documents.

After cleaning and preparing the data for analysis, it was necessary to proceed with studying the system, including both the payroll codes and the

**Figure 3.19:** DBT Layers.

output format of the redacted document, in order to be readable from the payroll software adopted by each client. Additionally, it was important to understand and analyze the set of parameters that each client can specify in order to tailor the documentation to internal needs.

3.5.1 Payroll codes and formats

Hence, designing a payroll module that can fit all different payroll software and pay codes is a challenging task. The reason for this is that every country, region, and industry has its unique payroll rules and regulations that can significantly impact payroll processing. Payroll software vendors must be able to accommodate these local requirements and regulations while also accounting for the nuances of each client's business. As noted in a report by the National Bureau of Economic Research, *implementing a new payroll system is often expensive and time-consuming, and there can be a significant risk of error if the new system does not handle all the complex rules and regulations that apply to payroll*. Therefore, it is important to carefully consider the design and development of the payroll systems to ensure that they can accommodate a wide range of payroll requirements and pay codes. This involves working closely with clients to understand their specific payroll

needs and customizing their systems to meet those needs.

In France, there are several pay codes used to determine an employee's compensation. Some of the most common pay codes include:

1. *Salaire de base*: This is an employee's base salary, which is typically paid on a monthly basis.
2. *Primes*: Primes are bonuses or incentives given to employees in addition to their base salary. These can be performance-based, tenure-based, or related to specific tasks or projects.
3. *Heures supplémentaires*: This refers to overtime hours worked by an employee, which are compensated at a higher rate than regular hours.
4. *Indemnités de transport*: Employees may be reimbursed for transportation costs associated with their work, such as a monthly pass for public transportation or mileage for using their personal vehicle.

Moreover, there are various payroll tools available in France that companies can use to manage their payroll. Some of the most popular options include:

- *Sage*: Sage is a cloud-based software that allows companies to manage their payroll, track employee hours, and generate reports. It also includes features such as electronic pay stubs and tax form filing.
- *ADP*: ADP is a global payroll and HR management company that offers a range of solutions for businesses of all sizes. Their payroll tools include automated tax compliance, direct deposit, and mobile access for employees.
- *SAP SuccessFactors*: SAP SuccessFactors is an HR management suite that includes payroll tools, as well as features for performance management, talent acquisition, and learning and development. It also integrates with other SAP products for a comprehensive HR solution.

- Payfit: Payfit is a French-based payroll software that offers a user-friendly interface and automates many of the payroll processes. It also includes features such as HR management, benefits management, and time tracking.

A comprehensive study of all the specifications outlined above was conducted to gain a detailed understanding of the field and to develop a solution that is as flexible and adaptable as possible to meet the customers' needs. After analyzing the various existing remuneration codes, it was determined that there are nine basic codes that have different names depending on the client's request. Therefore, a corresponding table was generated for each remuneration code, which includes the calculation of the relevant code for each mission.

Table 3.1 provides a summary of all existing codes.

CP1	Number of hours worked during the day on weekdays
CP2	Number of hours worked at night on weekdays
CP3	Number of hours worked on Sundays/public holidays during the day
CP4	Number of hours worked on Sundays/public holidays at night
CP5	Number of hours worked (day + night) on weekdays
CP6	Number of hours worked (day + night) on Sundays/public holidays
CP7	Number of hours worked at night
CP8	Number of hours worked during both day and night
CP9	Number of hours worked during the day
CP10	Number of hours worked on the day before a public holiday

Table 3.1: List of basic payroll codes.

3.5.2 Clients parameterization

In order to generate such document, it is necessary to take into account during the execution and calculation of payroll codes, each client's own specifications.

To achieve this, a Google sheet has been created. It is accessible by the company's CS and it is connected to a questionnaire, to ease the filling procedure. This allows all subscribing establishments to register the list of necessary information for producing the files. The CS responsible for managing a given establishment must fill out the Google sheet with details such as the Hublo identifier of the establishment, start times for day and night for calculating hours worked, various parameters to indicate whether to consider only validated missions or the entire list, whether to exclude break minutes from the total, and finally the required pay codes, following the list provided in table 3.1 with the desired name to assign to each of them.

After this document is filled out, it must be interfaced with the database so that the calculations can be automatically adapted for each client based on the declared parameters. Using the *Singer* library and creating a configuration file to access the spreadsheet via the API, it has been possible to send the contents of the sheet to the database. This ensures that the adaptation to the specifications is immediate in the pipeline creation process.

The spreadsheet is then saved as a table in DBT (paysheet_parameters), so that it can be called upon during the definition and formulation of a project.

Moreover, since many different views can be asked by each client, two different final tables have been created: the details per day and per mission for each worker.

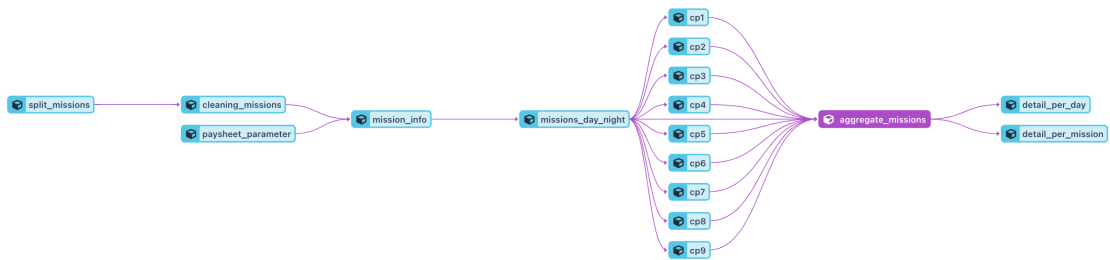


Figure 3.20: Data pipeline on DBT.

Figure 3.20 shows all the layers generated on DBT to finally have a global

view for the document generation. The final tables contain for each worker and establishment, the details per mission or per day, accompanied by all the attributes, such as the names, the total number of minutes worked and the payroll codes calculation.

3.5.3 Document generation

After the creation of all the fundamental tables containing essential information for payroll, it became necessary to develop a code to establish a connection with the database for accessing these tables and the Google Sheet, retrieving the specific requirements for each client, and generating one or more documents for each structure. To achieve this goal, a Python script was written utilizing API to connect to both the database and the parameterization document, which enables the creation of ad hoc queries for each customer based on their unique needs. Metabase, an open-source business intelligence and data analytics platform, was utilized for this purpose, which facilitates the connection to multiple data sources, construction of visualizations and dashboards, and report generation.

Therefore, to meet these requirements, a **Python** solution was implemented. By developing an API for connecting to the parameterization sheet and another for granting access to transformations created on *dbt*, and by using the Singer library, it was possible to establish a process for automatically generating public links to *Metabase*, with payroll files for each month, which can be easily shared with establishments. The algorithm accesses each row of the Google file to generate documents for all institutions that subscribe to the module and, for each one, based on the specified parameters, generates one or more public links with the desired files and details.

For each row of the file, using the *psycopg2* library, the code connects to the database and can access the *detail_per_day* and *detail_per_mission*

transformations. Based on the details required by the institution, the code creates `sql` queries that are sent to *Metabase*, using the *Metabase* API library. Metabase API library is a tool that allows developers to programmatically interact with Metabase’s functionality. The API provides endpoints for various functions such as querying data, creating and managing dashboards, and managing users and permissions. With this library, it is possible to automate tasks such as data ingestion, data processing, and report generation and also to create custom integrations with other tools and systems to streamline their data workflows. Additionally, the Metabase API provides secure access controls to ensure that sensitive data is only accessible by authorized users. Hence, by extracting the ID of this query, it was possible to generate a public link, also accessible to those who do not have permission to access the database, which is sent and saved directly in the Google spreadsheet. The administrative manager can therefore access this link and download the result as an Excel file to obtain the complete document with the payroll file for the previous month.

Here is the complete detail of the Python project built, along with the list of files and libraries used.

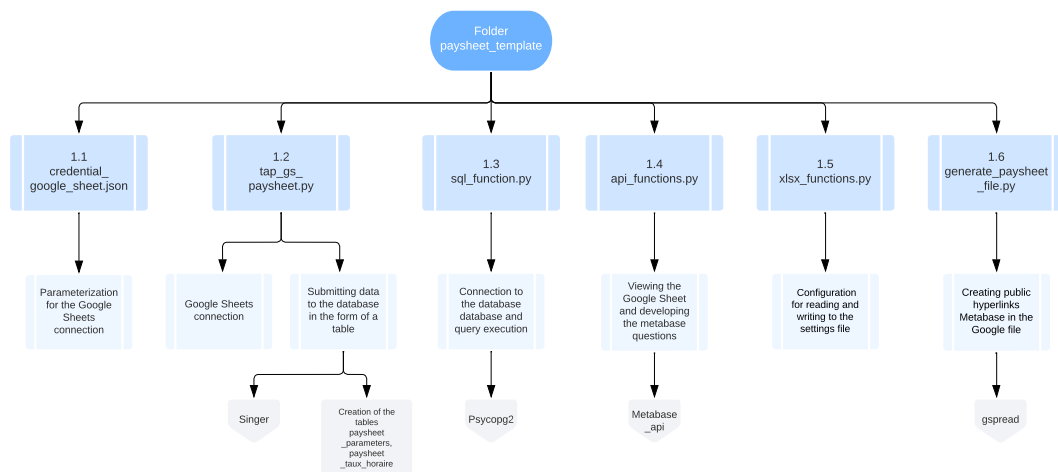


Figure 3.21: Project folder for document generation.

3.5.4 Process automation

The final stage of this project involves automating the execution of a script. Whenever a new client is added to the Google Sheet file, the paysheet documentation for the previous month should be generated. Additionally, the document should be generated at the end of every month. To ensure that the documents are always up-to-date, a tool needs to be adopted that can schedule the execution of the script, both when a new client subscribes and at the start of a new month.

One such tool is Heroku, which is a cloud-based platform that can be used with the Metabase API to create customized integrations for data analysis and visualization. Heroku allows developers to set up and manage their own instances of Metabase, an open-source business intelligence tool, and use its API to access and manipulate data. This enables companies to generate tailored dashboards and reports that align with their specific requirements, as well as automate the data analysis and reporting process.

By using the Metabase API, developers can easily create queries, visualize data, and establish automated reports using familiar tools and workflows. Furthermore, Heroku's built-in data services and tools can streamline the data pipeline, making it simpler to ingest and process data from different sources. This combination of Heroku and Metabase API provides a powerful platform for data-driven decision-making, which is flexible and scalable to cater to the needs of any organization.

This tool allows to connect to an environment and schedule the execution of jobs with a fixed frequency. Hence, with the following code [3.22](#), the execution of the script is scheduled every night, in a way that even if there is a delay of one day, the file is always up to date and the institutions receive the requested files.

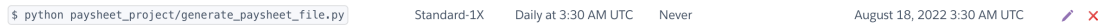


Figure 3.22: Heroku Scheduler.

As a consequence, Heroku Scheduler enables to schedule script execution and maintain up-to-date documentation for each client each month. Finally, an efficient and automated solution that does not require human participation in code execution whenever necessary is conceivable. In fact, when such a script is run, it checks the date and the number of lines in the Google file: if there are one or more lines without documentation, it generates the files while also checking the daily date, so that the file is generated at the beginning of each month for all customers on the list without the need for manual operations. If no adjustments are required, the script will end immediately.

Chapter 4

Analysis of results

The purpose of this study was to investigate the relevance of developing a pipeline to automatize the process of the payroll document generation, starting from the data collection until the creation of an automatized and efficient process for Hublo's customer plants. The primary goal of this thesis is to provide an automatic solution for a procedure that takes many hours to complete by healthcare staff who must sacrifice patient care time to perform administrative activities. Moreover, many studies have examined the significance of automation in the healthcare domain, citing the multiple benefits that administrative automation can bring to the system, particularly in a crucial context like today.

The introduction of automation brings about transformation. Automation will revolutionize the way labor is done, whether it is one task or an entire health care workflow. The ability to develop automation solutions and measure their impacts is dependent on how well the process and its intended goals are understood. The activities or processes, technology, organization and context, and persons engaged in the workflow that is being considered for automation must all be reviewed and accounted for in the proposed solution. Increasing the use of automation in health care and discovering automation

opportunities would necessitate an examination of the infrastructure, procedures, and systems employed by health care organizations in order to identify jobs that may be redundant across employees and operations. Comprehensive workflow studies using proven approaches, as well as technological assessments of current infrastructure and resources, are required to provide a full list of jobs, as well as their level of readiness and complexity for automation. To identify redundant workflows, informatics methodologies and technologies that use data from health IT systems and can be easily used across health care companies are required. Such techniques and technologies require research and demonstration projects to illustrate how they would perform at scale [11].

To accomplish the intended outcome, a data pipeline was utilized to collect and analyze data from the company's different platforms and products, and an automated procedure was constructed on top of it. The findings of this study are presented in this chapter, with the goal of providing insights into the benefits of administrative automation in the healthcare system by describing the proposed approach, GDPR compliance because personal data are involved, and potential future work that can be done starting with this project.

This chapter covers in detail the verification technique used to assure the authenticity and correctness of the created papers, as well as the significant challenges of the project. Hence, the verification step in generating payroll documents is crucial in ensuring the correctness and validity of the data used to generate the payroll. This step involves reviewing and validating the input data used to create the payroll, such as employee hours worked, deductions, and benefits. Verification ensures that the data used is accurate and complete, and that there are no discrepancies or errors in the calculations. The verification step is especially important as errors or inaccuracies in payroll calculations can result in incorrect employee payments, which can

negatively impact employee satisfaction and ultimately harm the reputation of the organization. Additionally, verification helps to ensure that the payroll is compliant with relevant regulations and standards, protecting the organization from legal liabilities. By ensuring the correctness and validity of data, the verification step plays a critical role in generating accurate and reliable payroll documents. As a result, this chapter describes how the process meets the major objectives of efficiency and performance, data correctness and coherence, and GDPR compliance, while also outlining the validation method carried out throughout the pipeline's execution and development.

The project's primary purpose is to deliver an efficient, high-performing solution that saves time for both the company and Hublo's consumers.

An initial solution for this project had previously been implemented, but it involved an entirely manual workflow using Excel. In actuality, the team manager was in charge of creating a monthly Excel file for each client, downloading data from the numerous platforms in use, and implementing the formulas required to determine the payroll codes required by each of them. This work was definitely demanding and took about two/three hours for each client, demonstrating that this solution was not applicable to an increasing number of clients but that the development of a scalable solution, one that would be efficient regardless of the number of clients who subscribed to this module, was required. In fact, at first, this module was only available to roughly 20 companies that requested it, but the amount of work was unmanageable on a monthly basis; in fact, the manager was losing approximately 60 hours of labor, which equated to more than a week, for generating such documents. Furthermore, the number of clients who began asking such a module grew rapidly, and this solution, which was not scalable, was no longer applicable. As a result, the creation of an automated pipeline was required in order to reduce workload and provide an effective solution

while also lowering the risk of drafting errors.

Therefore, the first goal had been achieved: as a result of the development of APIs and the scheduling of such scripts using Heroku Scheduler, the extraction of data and sending it toward the database became fully automatic, as did the generation of such documentation using the metabase API, which allows output files to be sent directly to the manager of each facility. Furthermore, the proposed solution includes the execution of such scripts every night to ensure that all data is always up to date, as well as a continuous check on the number of customers subscribing to the module, so that each new facility receives the generated documentation without delay the next day. Each script takes only a few seconds to execute, making such a pipeline efficient and speedy.

The second goal is to assure the security and accuracy of the data being transmitted.

As a result, two distinct control methods have been implemented: the first is to verify the accuracy of the information submitted on the applications utilized (Hubspot, Planhat, and Google Sheet), and the second is to monitor the execution of Python scripts.

Because certain client information is entered manually into Hubspot, Planhat, and Google Sheets, it is critical to build a check to ensure that this information adheres to the proper forms and does not cause mistakes in documentation production. In the case of the Google Sheet, input limitations have been set up to ensure that any information supplied satisfies the established format and character requirements.

In the case of Hubspot and Planhat, however, alarms were developed to ensure that the information provided is valid and matches the database's

standards. The Zapier Tool was used to accomplish this, which allows connecting to the database, implementing SQL queries for verification, and sending notifications if specific values do not fit the required parameters. Zapier provides workflows to automate the use of web applications together. It is often described as a translator between web APIs, helping to increase worker productivity by saving time through automation of recurring tasks, and business processes such as lead management. Through an interface in which users can set up workflow rules to determine how its automations function, it orchestrates flow of data between tools and online services that wouldn't otherwise communicate with one another. Furthermore, such a tool can connect to the Slack application and deliver predefined messages once an alert is highlighted. As a result, this approach allows to be informed at all times when certain information does not meet the required standards and to modify it before causing errors in the generated documentation.

Heroku's error management tools system was used to validate the execution of Python scripts, which can catch exceptions or errors caused by the application, whether they come from the code, dependencies, or a framework. You may inspect the stack trace of an error and trace it back to its root cause using error monitoring. Seeing the errors reported by the application during an incident can help you resolve problems more quickly. This tool allows you to check the status of each script's execution and send email notifications in case of errors: this system allows you to be always on the lookout for anything that goes wrong during execution and to double-check the documentation before delivering it to the client.

This double layer of security allows to ensure that the documentation is always right, that any errors are corrected before submission, and that the data being submitted is efficient and valid. In fact, because the project is

focused on the compilation of payroll records, it is critical that the information provided is accurate, as this data has a direct impact on the health care workforce. As a result, this method allows to handle any potential mistakes before they are sent to the facility in question, avoiding any inconsistencies.

The third purpose is to comply with the GDPR regarding the handling of personal data.

The General Data Protection Regulation (GDPR) is a set of regulations that govern the handling and processing of personal data within the European Union (EU). When using the Metabase API, it is important to ensure that you are respecting the GDPR by only collecting and processing personal data that is necessary for the intended purpose, obtaining consent when required, and implementing appropriate security measures to protect the data. Another key aspect of GDPR compliance is implementing appropriate security measures to protect the personal data that is collected and processed through the Metabase API. This may include using encryption to protect data in transit and at rest, limiting access to data based on user roles and permissions, and ensuring that data is deleted when it is no longer needed for its intended purpose. Overall, when using the Metabase API, it is important to keep GDPR regulations in mind and take appropriate measures to ensure the protection of personal data. This will not only help to maintain compliance with GDPR requirements but will also build trust with users and demonstrate a commitment to data privacy and security.

In fact, only the strictly necessary information of the health personnel, such as first name, last name, and occupation, is gathered and used in the compilation of this documentation to assure this. Furthermore, to ensure that the files are not made public but are only sent to the facility in question, a link of its own is generated, using the Metabase API, for each manager who can thus view only the information related to his or her own facility and is automatically sent to him or her, so that there are no problems of

non-compliance with the operators' privacy.

Thus, the established goals of efficiency, fairness, and GDPR compliance were met in the creation and development of this pipeline for creating the necessary documents.

The final step is to review the generated documentation to ensure that the pay code computation was correct and that the information entered was accurate. An automatic check could not be performed since the data differed for each facility and each pay code requested. To do this, a manual check was implemented: for all facilities for which the pre-existing solution existed, files from the same month were compared to see if the number of lines, pay code value, and mission length matched. Because there were enough existing documents, it was assumed that if all of them matched, the process was correct.

Therefore, for each existing file, all data were compared to verify correspondence, appropriate adjustments were made to fix problems, and the output format of each was confirmed. Naturally, verification was undertaken throughout the pipeline building process to ensure that each step, table, and extraction performed was consistent and that the data was legitimate. The analysis of the results was thus a part of the overall process, and the verification and validation step was an essential component of the progress. To assure the success and correctness of the created documentation, each creation process was juxtaposed with a verification and error correction step.

After the verification and correction procedure was completed, the project was given to the company, displaying the work's ultimate outcome. As a result of the proposed solution, the payroll document creation package could be offered to all of Hublo's client plants, rather than just the original few that profited from the module. The creation of such a pipeline allowed not

only for the solution to be scalable and configurable, but also for it to be included as a Hublo Match bonus package to which any client facility may subscribe.

Throughout this chapter, it has been emphasized that ensuring the quality and reliability of data is of utmost importance. Adopting a strategy that includes data cleaning, normalization, and verification is essential in identifying and addressing inconsistencies, errors, and missing data. By doing so, the accuracy and validity of results can be improved, enabling businesses to make more informed decisions and develop effective strategies.

In addition to ensuring data quality, adhering to GDPR regulations is crucial in establishing ethical data collection and management practices that are respectful of participants' privacy rights. By implementing measures such as obtaining informed consent, implementing appropriate security measures, and establishing clear data retention policies, businesses can mitigate the risk of data breaches, protect participants' personal information, and ensure compliance with GDPR regulations.

Furthermore, the benefits of administrative automation for Hublo and its clients cannot be overstated. By automating routine administrative tasks such as data entry, invoicing, and document management, the amount of time spent on these activities can be significantly reduced. This allows the company to free up valuable resources and personnel, allowing them to focus on more critical and value-adding tasks such as data analysis and strategy development. The benefits of administrative automation extend beyond time-saving, as it also reduces the risk of errors, enhances accuracy and precision, and improves overall efficiency.

In today's fast-paced and highly competitive business environment, automation is no longer a luxury but a necessity. By automating administrative

tasks, Hublo can become more agile and responsive, enabling it to meet the ever-changing needs and expectations of its clients. Furthermore, automation can improve communication and collaboration within the company, streamlining internal workflows and enhancing organizational effectiveness.

Chapter 5

Conclusion

The healthcare industry faces several major issues, including the rising cost of healthcare, shortages of healthcare workers, complex regulatory environments, and data privacy and security concerns. These challenges present significant hurdles for healthcare companies, policymakers, and patients alike. However, administrative automation offers a potential solution to many of these issues. By streamlining routine administrative tasks and increasing efficiency, healthcare companies can reduce costs, alleviate the burden on healthcare workers, and free up resources to focus on patient care. Additionally, automation tools can help to improve accuracy and reduce errors, which can lead to better patient outcomes. While healthcare companies must still navigate complex regulatory environments and address data privacy and security concerns, automation can help to streamline these processes and make healthcare operations more secure and sustainable. Overall, administrative automation offers a promising opportunity to address many of the challenges facing the healthcare industry and improve the overall quality of healthcare services for patients.

Payroll processing is a critical function for any organization, and this is especially true in the healthcare industry. With numerous employees,

varying pay rates, and complex regulatory requirements, payroll processing can be time-consuming, error-prone, and resource-intensive. Manual payroll processing methods are often inefficient, which can lead to delays, errors, and poor employee morale. Automating payroll paper generation is a viable solution that can provide numerous benefits for companies, including improved efficiency, accuracy, and cost savings.

The purpose of this thesis was to explore the technical solution for automating payroll paper generation in the healthcare industry, specifically for Hublo's customer plants. The proposed solution focused on the development of a comprehensive approach that leverages cloud-based platforms and APIs to integrate various systems and automate manual tasks. The solution was designed to meet the unique needs and requirements of Hublo's customer plants and to streamline their payroll processing.

The solution presented in this thesis involves the use of Google Sheets to manage employee data, a custom Python script to generate payroll papers, and the Heroku platform to automate the script execution. The proposed solution can generate payslips for each employee based on their worked hours and pay rates, which can then be exported as PDF documents. The solution is also designed to automate the process of generating payroll papers at the end of each month and whenever a new employee is added to the system.

The proposed solution has numerous benefits for Hublo's customer plants. Firstly, it can improve the accuracy and consistency of payroll processing, which is critical in the healthcare industry where regulatory compliance is essential. Secondly, it can streamline the payroll processing workflow, reducing the time and resources required to generate payroll papers. Finally, the solution can help reduce errors and delays, which can have a positive impact on employee morale.

There are several potential future works that could build upon the research presented in this thesis. One potential area for future exploration is the integration of additional automation tools and technologies to further streamline the payroll generation process. For example, the use of machine learning algorithms or natural language processing could be leveraged to automate the extraction of data from various sources, reducing the need for manual data entry.

Another potential area for future work is the implementation of more robust security measures to protect sensitive payroll data. As the healthcare industry becomes increasingly digital, the risk of data breaches and cyber attacks is a growing concern. The implementation of strong encryption and access controls could help to mitigate these risks and ensure the confidentiality and integrity of payroll data.

Finally, additional research could be conducted to explore the potential benefits of automation for other administrative tasks in the healthcare industry. Payroll is just one area where automation can provide significant benefits; other tasks such as scheduling, inventory management, and patient record-keeping could also benefit from automation. Further research in this area could help healthcare companies to identify additional opportunities for automation and improve overall efficiency and patient care.

Bibliography

- [1] World Health Organization. *Nursing and midwifery*. <https://www.who.int/news-room/fact-sheets/detail/nursing-and-midwifery>. 2022 (cit. on pp. 17, 18).
- [2] République Française: Direction de la recherche, des études, de l'évaluation et des statistiques. «Les travaux de la DREES liés à la crise sanitaire du Covid-19». In: (2021) (cit. on pp. 18, 20).
- [3] Fédération Hospitalière de France. «ENQUÊTE - Situation RH». In: (2022) (cit. on pp. 19, 20).
- [4] Déborah Berthier. *Europe has a shortage of doctors*. Ed. by European Data Journalism Network. <https://www.europeandatajournalism.eu/eng/News/Data-news/Europe-has-a-shortage-of-doctors>. 2018 (cit. on p. 20).
- [5] Julie Le Bolzer. *MedGo simplifie les remplacements en milieu hospitalier*. Ed. by Les Echos ENTREPRENEURS. <https://business.lesechos.fr/entrepreneurs/idees-de-business/0301430927014-medgo-simplifie-les-remplacements-en-milieu-hospitalier-319598.php>. 2018 (cit. on p. 24).
- [6] S.M.Hari Krishna and R. Sharma. «Survey on application programming interfaces in software defined networks and network function virtualization». In: (2021) (cit. on p. 38).

- [7] F. Ridzuan and W.M.N. Wan Zainon. «A Review on Data Cleansing Methods for Big Data». In: (2019) (cit. on p. [43](#)).
- [8] E. Rahm and H. Hai Do. «Data Cleaning: Problems and Current Approaches». In: (2000) (cit. on p. [43](#)).
- [9] A.V. Bataev and I.S. Davydov. «The role of automation in improving the quality of enterprise business processes». In: (2020) (cit. on p. [45](#)).
- [10] Asana. «Anatomy of Work Index: How people spend their time at work». In: (2020) (cit. on p. [46](#)).
- [11] Teresa Zayas-Cab Tracy H. Okubo and Steven Posnack. «Priorities to accelerate workflow automation in health care». In: (2022) (cit. on p. [58](#)).