PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
الجمهـــوريـــــة الجـزائـريـــــة الديـمقـراطيـــــة الشـعـبيـــــة
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليـم العالـــي والبحـــث العلمـي
HIGHER SCHOOL OF COMPUTER SCIENCE AND DIGITAL TECHNOLOGIES-BEJAIA
المـــدرسة العـــليا في علـــوم وتكنـولوجيـات الاعـــلام الالـــي بجـــاية

**Dissertation Submitted to the Department Of Computer Science in Partial Fulfillment of the Requirements for Engineer's Degree in Computer Science Specialty: Artificial Intelligence and Data Sciences**

Submitted By:

**SAIDA Haithem & BENAFIA Mohamed Abdessamed**

# Supply Chain And AI, Case Of Cevital

Supervised by:

**Pr.** KHERBACHI Hamid ESTIN

**Mr**. IDIR Anis CEVITAL

**Members of jury:**

| | | |
|---|---|---|
| ▪ **Dr.** ZENADJI Sylia | President | ESTIN |
| ▪ **Dr.** KHALFI Lynda | Examiner | ESTIN |
| ▪ **Dr.** ISSADI Badredine | Examiner | ESTIN |
| ▪ **Mr**. IDIR Anis | Internship Tutor | CEVITAL |

Academic year: 2024/2025

# Acknowledgement

First and foremost, we thank God, the Most Gracious and Most Merciful, for granting us the strength, patience, and perseverance to complete this thesis. Without His guidance and blessings, none of this would have been possible.

We would like to express our sincere gratitude to all those who contributed to the completion of this master's thesis, titled *"Supply Chain and AI: Case of Cevital."*

We extend our deepest thanks to our academic supervisor, Pr. KHERBACHI Hamid, for his valuable guidance, insightful feedback, and continuous support throughout this research. His expertise and encouragement have been instrumental in shaping the direction and quality of our work.

We also wish to warmly thank each other as teammates for the collaboration, dedication, and mutual support that made this journey both productive and enriching. The strong teamwork between us was a key factor in the successful realization of this project.

This thesis was carried out in collaboration with Cevital, and we are especially grateful to Mr. IDIR Anis for his availability, guidance, and trust. His contributions and the opportunity to work closely with a major industrial actor have greatly enriched our research.

Above all, we would like to express our deepest and most heartfelt gratitude to our families. Their unwavering support has been the foundation of this journey. To our parents, thank you for your sacrifices, your unconditional love, and your constant belief in us. Your encouragement and presence, even from afar, have given us the strength and resilience to persevere through the most challenging moments. This achievement is as much yours as it is ours.

# LIST OF ACRONYMS

| | |
|---|---|
| **SCM** | Supply Chain Management |
| **AI** | Artificial Intelligence |
| **ERP** | Enterprise Resource Planning |
| **TMS** | Transportation Management System |
| **RL** | Reinforcement Learning |
| **MAS** | Multi-Agent System |
| **ML** | Machine Learning |
| **NLP** | Natural Language Processing |
| **EDA** | Exploratory Data Analysis |
| **ARIMAX** | Autoregressive Integrated Moving Average with Exogenous Variables |
| **GAN** | Generative Adversarial Network |
| **CGAN** | Conditional Generative Adversarial Network |
| **LLM** | Large Language Model |
| **DL** | Deep Learning |
| **ETL** | Extract, Transform, Load |
| **BI** | Business Intelligence |
| **RAG** | Retrieval-Augmented Generation |
| **IoT** | Internet of Things |
| **XAI** | Explainable Artificial Intelligence |
| **RPA** | Robotic Process Automation |
| **NER** | Named Entity Recognition |
| **SVM** | Support Vector Machine |
| **RF** | Random Forest |
| **LTSM** | Long Short-Term Memory |
| **RNN** | Recurrent Neural Network |
| **COT** | Chain of Thought |
| **CSV** | Comma-Separated Values |
| **WAN** | Wide Area Network |
| **LAN** | Local Area Network |
| **API** | Application Programming Interface |
| **SQL** | Structured Query Language |
| **UI** | User Interface |

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# INTRODUCTION

In today's global and digital economy, supply chains are increasingly complex and vulnerable to disruptions. Ensuring efficiency, agility, and resilience is critical, especially in the agri-food sector, where strict regulations, fluctuating demand, and environmental constraints add pressure.

Recent crises such as the COVID-19 pandemic, the war in Ukraine, and climate change have exposed weaknesses in traditional supply chain models. These challenges have accelerated interest in innovative solutions, with Artificial Intelligence (AI) emerging as a transformative technology. From demand forecasting to intelligent routing and anomaly detection, AI brings unprecedented capabilities to optimize supply chain operations.

While academic research has explored machine learning, deep learning, and reinforcement learning in SCM, real-world adoption faces limits such as data quality issues, high implementation costs, and lack of sector-specific solutions. Yet, leading companies are already leveraging AI to automate decisions, gain visibility, and improve performance.

## Background

Our analysis of over 30 studies and market reports highlights a major shift: AI in supply chains is projected to grow at a CAGR of 45.55%, reaching USD 10.1 billion by 2025. Robotics adoption is also booming, with over 4 million robots expected in 50,000 warehouses by the same year [1].

AI systems already deliver tangible value cutting logistics costs by up to 15%, reducing inventory by 20%, and boosting service levels by 40%. Yet, many firms still struggle to scale AI due to fragmented data, limited training, and unclear ROI. Adoption is rising nonetheless: over 60% of companies use AI in analytics, and 91% of manufacturers aim to combine AI with supply chain systems to boost agility and visibility [1].

Experts emphasize the need for clean data, agile systems, and workforce upskilling. Even small manufacturers now leverage lean AI tools to stay competitive. These insights confirm a global shift: AI is no longer optional it is essential for modern supply chains.

# Research Objectives

This thesis explores how AI can improve supply chain performance, with a focus on CEVITAL, a major agri-food company in Algeria. It addresses the question: *How can AI be realistically applied to enhance decision-making and resilience in industrial supply chains?*

The study combines a literature review with a practical case study. An audit of CEVITAL's infrastructure revealed key limitations: fragmented data systems, limited real-time visibility, and poor data quality, especially in the Transport Management System (TMS).

Rather than building complex predictive models, the thesis focuses on designing practical, low-cost AI tools. We developed a locally hosted, AI-powered reporting system using open-source LLMs, pandas agents, and LangChain. This tool automates report generation from supply chain data using natural language, reducing reliance on the BI team and speeding up decision-making.

By aligning with CEVITAL's technical and financial constraints, this project demonstrates how AI can bring immediate value and serve as a foundation for future digital transformation.

# Thesis Structure

This thesis is divided into five chapters covering the theoretical foundation, case study, data analysis, system development, and evaluation of an AI solution for supply chain management.

- **Chapter 1: Literature Review** Covers definitions, theoretical background on supply chain management, Artificial Intelligence, and recent developments in generative AI and AI agents, as well as related work using AI techniques in SCM.
- **Chapter 2: CEVITAL Company Study** Provides a detailed overview of CEVITAL, including its organizational and supply chain structure, technological infrastructure, challenges, and how this thesis proposes to address them.
- **Chapter 3: Exploratory Data Analysis** Describes the data collection process, evaluates data quality, and presents key insights derived from analyzing supply chain-related data.
- **Chapter 4: System Design and Implementation** Details the design and architecture of the AI-based system, including its components such as the data pipeline, agent design, reasoning engine, and interaction flow.
- **Chapter 5: Benchmarking and Evaluation** Presents the benchmarking methodology, evaluation metrics, test results, and a discussion on the system's performance and limitations.
- **Conclusion:** Summarizes the main contributions of the thesis and highlights potential directions for future work.

# CHAPTER 1

# LITTERATURE REVIEW

Supply Chain Management (SCM) involves the coordination of procurement, production, and distribution to ensure the efficient flow of goods and services. With increasing complexity and global challenges, traditional supply chains require more agility and responsiveness.

Artificial Intelligence (AI) is playing a critical role in this transformation. Technologies such as machine learning, predictive analytics, and optimization algorithms are enabling real-time decision-making, risk detection, and process automation. These tools help organizations build smarter, more efficient, and resilient supply chains.

This chapter introduces the key technical concepts of modern SCM, focusing on how AI-driven solutions are reshaping operations and enhancing overall performance.

## 1.1 Definition of Supply Chain

A supply chain is a complex network of interconnected organizations including suppliers, manufacturers, distributors, retailers, and customers that collaborate to produce, distribute, and deliver products. This network facilitates the flow of goods, information, and finances throughout the entire process, from the procurement of raw materials to the delivery of finished goods to end consumers. Operationally, the supply chain comprises a system of facilities responsible for acquiring raw materials, transforming them into components and finished products, and ensuring their distribution. The term "chain" highlights the interdependence of the various actors and flows within the system, encompassing upstream-to-downstream material flows, bidirectional information exchanges, and downstream-to-upstream financial flows. This integrated system applies to both manufacturing and service sectors [2].

According to Misra et al. [3], a supply chain is defined as a network of suppliers, factories, warehouses, distribution centers, and retailers, through which raw materials are acquired, transformed, produced, and delivered to the customer. It encompasses all activities associated with the flow and transformation of goods from the raw material stage through to the end user, along with the associated information flows.

## 1.2 Artificial Intelligence, Machine Learning, and Deep Learning

Artificial Intelligence refers to the simulation of human intelligence in machines, enabling them to perform tasks that typically require human cognition such as reasoning, problem-solving, perception, and decision-making. AI systems can process large amounts of data, recognize patterns, and make informed decisions. The field of AI encompasses various techniques, including some older methods that laid the foundation for modern developments.

Historically, AI relied on symbolic reasoning and rule-based systems. Traditional AI techniques, such as **Fuzzy Logic**, were designed to handle uncertainty and vagueness by allowing systems to reason with imprecise or approximate information. Similarly, **Genetic Algorithms** were inspired by the process of natural evolution and used in optimization problems, where a solution is evolved over generations based on principles of selection, crossover, and mutation. These early AI approaches provided valuable insights and still find applications today in areas like control systems, optimization, and decision-making under uncertainty.

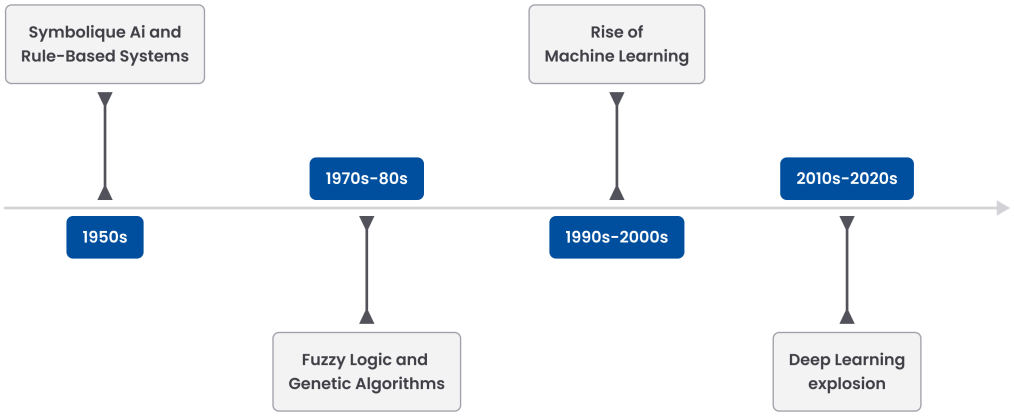


Figure 1.1: Timeline of Major Milestones in the History of Artificial Intelligence

Despite these advancements, traditional AI techniques often struggled with handling complex, high-dimensional data and real-world variability, which led to the rise of Machine Learning (ML) as a more dynamic and data-driven approach.

Machine Learning (ML) is a field of artificial intelligence that enables systems to automatically learn and improve from experience without being explicitly programmed, by analyzing data and identifying patterns to make decisions or predictions. [4]

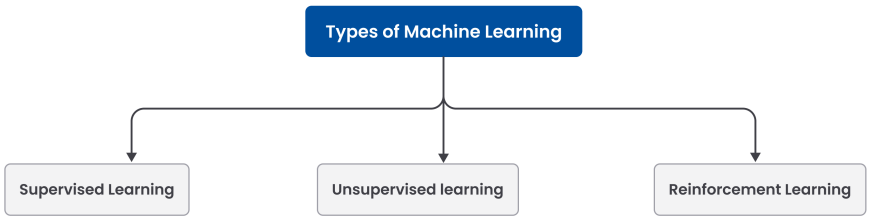


Figure 1.2: Types of Machine Learning Techniques

Machine learning is typically categorized into three types:

- **Supervised Learning.** In this approach, the model is trained using labeled data, where the input data is paired with the correct output. The goal is for the model to learn the relationship between the input and output so that it can predict the output for new, unseen data. Common algorithms in supervised learning include *Linear Regression*, *Decision Trees*, and *Support Vector Machines (SVM)*.
- **Unsupervised Learning.** This approach involves training models on data that is not labeled, meaning the output is not provided. The goal is for the model to identify patterns, structures, or groupings within the data. Examples of unsupervised learning algorithms include *K-Means Clustering* and *Principal Component Analysis (PCA)*.
- **Reinforcement Learning.** In reinforcement learning, an agent learns how to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on the actions it takes, and it aims to maximize the cumulative reward over time. Common algorithms include *Q-Learning* and *Deep Q Networks (DQN)*.

Machine learning has a wide range of applications, including natural language processing, computer vision, fraud detection, predictive analytics, and autonomous systems. It relies heavily on the availability of large datasets and computational power to train complex models that can generalize and make predictions based on new, unseen data.

Deep Learning (DL) is a subset of Machine Learning (ML) that focuses on the use of artificial neural networks, particularly deep neural networks, to model and solve complex problems. Deep learning models are designed to automatically learn hierarchical representations of data, allowing them to handle tasks such as image recognition, speech processing, and natural language understanding.

Deep learning models consist of multiple layers of interconnected neurons, each layer transforming the input data into increasingly abstract ways. The depth of these networks hence the term ”deep learning” enables the models to learn from large and complex datasets with minimal feature engineering.

Key components of deep learning include:

- **Neural Networks.** Neural networks are the foundational structures in deep learning. A basic neural network consists of an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to neurons in the adjacent layers, and the strength of these connections is adjusted during training.

- **Convolutional Neural Networks (CNNs).** CNNs are a specialized type of neural network primarily used for image and video recognition. They use convolutional layers to scan input data and capture spatial hierarchies in images, making them particularly powerful for tasks like object detection and facial recognition.
- **Recurrent Neural Networks (RNNs).** RNNs are used for sequential data such as time series or text. They have connections that form cycles, allowing information to persist across time steps. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are popular variations of RNNs, designed to mitigate the vanishing gradient problem and capture long-term dependencies.
- **Generative Models.** Deep learning includes generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which are used for generating new data based on the patterns learned from the training data. GANs, for example, have been used for generating realistic images and deepfake videos.

Deep learning is particularly effective at solving problems where traditional machine learning methods struggle, such as in unstructured data like images, audio, and text. The main advantages of deep learning models are their ability to learn directly from raw data and their ability to scale with the amount of data and computational resources available. However, they often require large labeled datasets, significant computational power, and can be computationally expensive to train.

Applications of deep learning include autonomous driving, medical imaging, natural language processing (e.g., speech recognition, sentiment analysis), and content recommendation systems.

## 1.3 Generative AI and AI Agents

Generative Artificial Intelligence (Generative AI) refers to a type of AI system capable of producing original content, such as text, images, audio, video, and synthetic data, by learning from existing real-world examples. It functions by analyzing patterns, structures, correlations, and trends within data to simulate aspects of human creativity and cognition. Generative AI systems operate by receiving an input such as a prompt, image, or audio segment and generating new outputs that resemble the learned data distributions. The development of Generative Adversarial Networks (GANs) in 2014 significantly advanced this field, allowing for the creation of highly realistic content. In the context of supply chain management, generative AI offers transformative capabilities, including improved forecasting, synthetic data generation for simulation, operational optimization, and enhanced adaptability in increasingly volatile environments [5].

An AI agent is an autonomous entity that perceives its environment, processes information, and takes actions to achieve specific goals. AI agents are designed to mimic intelligent human behavior and operate without direct human intervention. They are commonly used in a wide range of applications, including robotics, gaming, decision-making, and supply chain management.

AI agents are built to interact with their environment, adapt to changes, and make decisions based on their goals and the information they gather. They can be as simple as

a reactive system responding to immediate stimuli or as complex as a learning agent that continuously improves its performance through experience.

It consists of several key components that enable it to perform its functions autonomously. These components include:

- **Perception.** The agent's ability to perceive its environment is critical for decision-making. Perception involves collecting data through sensors, cameras, or any other input mechanism. This data may include environmental conditions, actions taken by other agents, or state changes in the system.
- **Reasoning/Decision-Making.** After perceiving the environment, the agent must process the information to decide on the appropriate course of action. This is where reasoning algorithms come into play. The agent can use rules, machine learning models, or probabilistic reasoning to decide on the best possible action.
- **Action/Actuation.** Once a decision is made, the agent must take action. The action component is responsible for executing the selected decision. It may involve sending commands to robotic actuators, altering the system state, or communicating with other agents.
- **Memory (optional).** Some AI agents may possess memory, allowing them to store and recall past experiences. Memory enables the agent to make better decisions by considering previous interactions and outcomes, which is particularly important in learning agents.
- **Learning (optional).** In more advanced AI agents, the learning component allows the agent to improve its decision-making over time by learning from its environment. Machine learning techniques, such as reinforcement learning or supervised learning, are used to refine the agent's behavior based on feedback or experience.
- **Autonomy.** The degree of autonomy determines how much control the agent has over its actions without needing human intervention. An autonomous agent can perform tasks independently and adapt to its environment, while less autonomous agents may require more guidance or supervision.

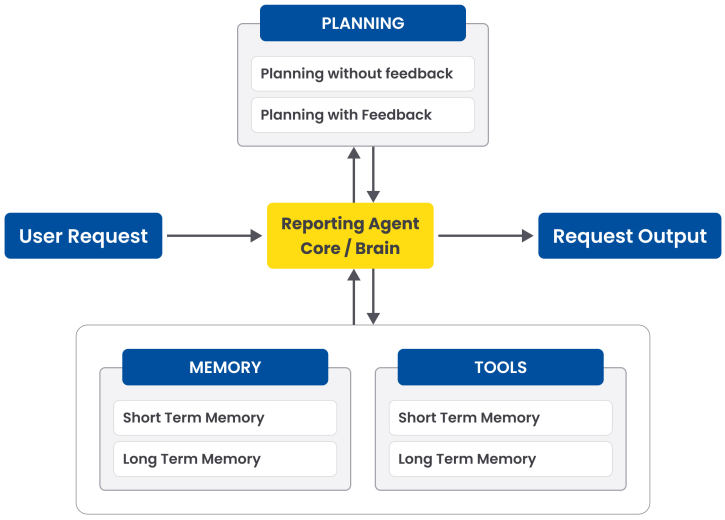


Figure 1.3: Main components of AI agents.

In conclusion, AI agents consist of multiple components that work together to enable autonomous functioning. These components perception, reasoning, action, memory, learning, and autonomy allow AI agents to interact effectively with their environment and achieve their goals. Depending on their application, AI agents can vary in complexity, from simple reactive systems to sophisticated learning agents capable of adapting and improving their behavior over time.

## 1.4 Related Work

In this section, we present two complementary approaches for applying Artificial Intelligence in Supply Chain Management. The first approach leverages classical Machine Learning techniques to identify patterns, optimize processes, and enhance decision-making through predictive analytics. This includes models trained on historical supply chain data to forecast demand, detect anomalies, and support inventory management.

The second approach is centered on AI Agents autonomous systems capable of reasoning, learning, and interacting with data and users. These agents integrate advanced techniques such as Large Language Models, and transformer-based architectures. Specifically, we explore the use of multi-agent systems and AI reasoning frameworks to facilitate intelligent decision support, dynamic planning, and adaptive responses across the supply chain.

Together, these approaches demonstrate the potential of AI to transform traditional supply chain operations into intelligent, data-driven systems capable of responding to complex, real-world challenges.

### 1.4.1 Machine Learning Techniques in Supply Chain Management

Machine Learning (ML) techniques enhance supply chain operations by enablinxg accurate forecasting, anomaly detection, real-time decision-making, and optimization across logistics, inventory, procurement, and distribution. These techniques can be categorized into two main families: statistical and classical methods, and deep learning or advanced learning paradigms.

Statistical and classical machine learning methods are proven, efficient models that work well with structured supply chain data. They offer good accuracy and easy interpretation, making them useful for many tasks like forecasting, classification, and decision-making. From these methods, we mention several key examples used in supply chain management:

- **Random Forests (RF).** is a set of decision trees used extensively in demand forecasting and supplier classification, due to robustness against overfitting and ability to capture feature importance [6, 7].
- **Support Vector Machines (SVM).** is effective in binary classification problems such as quality control and supplier risk evaluation [6].
- **Linear and Logistic Regression.** is widely used for price forecasting, trend analysis, and binary outcomes (e.g., delivery success/failure) [8].
- **Decision Trees (DTs).** is employed in supply segmentation, warehouse site selection, and rule-based decision-making due to their interpretability [6].

- **Gradient Boosting Trees / XGBoost.** is used in high-stakes scenarios like delivery time prediction and inventory level classification, with superior performance on tabular data [9].
- **Clustering Techniques (e.g., K-means).** are applied in customer segmentation, market clustering, and classification of supply routes based on historical data [9].
- **Hybrid Models (e.g., ARIMA + ML).** Combine time-series statistical models (like ARIMA) with machine learning algorithms (e.g., XGBoost, SVM, or neural networks) to model both trend and seasonality as well as non-linear behaviors. These are particularly useful for improving demand forecasting under variable or uncertain environments [8]. For example, ARIMA models capture the trend and seasonality, while XGBoost captures residual error components resulting from external factors such as promotions or weather.

Deep Learning and Advanced Learning Paradigms techniques excel in capturing complex, high-dimensional, and temporal patterns, often outperforming classical methods in scenarios with rich data.

- **Artificial Neural Networks (ANNs).** Deployed in supply chain forecasting and inventory control, ANNs are able to model complex input–output relationships, especially when domain features are known but their relationships are nonlinear [7].
- **Long Short-Term Memory (LSTM).** Networks: A type of Recurrent Neural Network (RNN) specifically designed to handle long-range dependencies in time series. LSTMs are applied in demand planning and transportation delay forecasting because they can retain temporal context over many steps. For example, in inventory management, LSTMs forecast future stockouts based on seasonality, historical sales, and lead time disruptions [9].
- **Deep Neural Networks (DNNs).** DNNs are used for multi-modal data integration tasks such as predicting product returns using transaction history, customer reviews (text), and item images. These models handle large, diverse datasets and provide high accuracy but require large volumes of labeled data [7].
- **Reinforcement Learning (RL).** RL is suited for dynamic decision-making problems such as order batching, warehouse picking optimization, and adaptive inventory replenishment. It learns through trial-and-error interactions with the environment, aiming to maximize cumulative reward (e.g., minimized cost or maximized service level) [6].
- **Conditional Generative Adversarial Networks (CGANs).** CGANs are advanced generative models used to simulate realistic supply chain disruption scenarios, such as delayed shipments or demand surges, based on historical data. By conditioning on external variables (e.g., region, supplier type), CGANs allow the generation of synthetic scenarios for resilience testing, training robust forecasting models, or augmenting small datasets [7].

Machine learning techniques offer several strengths that make them valuable for supply chain management. They often provide superior predictive accuracy, especially ensemble methods and deep learning models, which outperform many traditional forecasting tools. Their flexibility allows advanced methods like reinforcement learning (RL) and conditional

generative adversarial networks (CGANs) to adapt to complex and dynamic environments. Additionally, deep learning approaches can integrate multiple data types, handling structured data alongside unstructured sources such as text and images, as well as temporal information.

However, these techniques also have limitations. Many models, including LSTMs and CGANs, require large amounts of high-quality, labeled data for effective training, which can be a barrier in some contexts. The complexity of these models often leads to high computational costs and difficulty in interpretation, potentially restricting their use in resource-constrained or highly regulated environments. Moreover, there is a risk of overfitting, meaning these models may struggle to generalize well during rare or unforeseen disruptions, such as pandemic-like shocks.

Evaluating the performance of machine learning models is a crucial step in supply chain applications to ensure their reliability and effectiveness . Whether predicting future demand, classifying suppliers, or optimizing logistics, different models require appropriate metrics to assess their accuracy, robustness, and generalization. We will outline the most commonly used evaluation techniques along with their mathematical formulations [10].

For regression problems, commonly encountered in demand or lead-time forecasting, the Mean Absolute Error (MAE) provides an intuitive measure of the average magnitude of errors, regardless of their direction. It is easy to interpret and widely used in operational settings.

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t| \tag{1.1}$$

The Root Mean Squared Error (RMSE) is another standard regression metric that penalizes larger errors more heavily than MAE, making it suitable for applications where large deviations are particularly costly.

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{1.2}$$

For classification tasks such as supplier risk categorization or defect detection, accuracy measures the proportion of correct predictions. While easy to understand, it may be misleading in imbalanced datasets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.3}$$

Precision focuses on the relevance of positive predictions, making it critical in contexts where false positives have high cost, such as flagging a reliable supplier as high-risk.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.4}$$

Recall (or Sensitivity) is vital in scenarios where missing a positive case is costlyfor example, failing to detect a defective batch.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.5}$$

The F1-score provides a harmonic mean between precision and recall, balancing their trade-off and being especially useful in skewed datasets.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.6}$$

In time series modeling, particularly for ARIMA models, information criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are widely used to compare different model configurations . These criteria penalize model complexity to prevent overfitting while rewarding better fit [11].

$$\text{AIC} = 2k - 2\ln(L) \tag{1.7}$$

$$\text{BIC} = k\ln(n) - 2\ln(L) \tag{1.8}$$

Here, $k$ is the number of parameters in the model, $n$ is the number of observations, and $L$ is the likelihood of the model. Lower AIC or BIC values indicate a better model under their respective penalty schemes.

### 1.4.2 AI Agents in Supply Chain Management

AI agents are increasingly deployed in supply chain management (SCM) to automate decision-making, enhance real-time responsiveness, and improve operational efficiency. These autonomous systems analyze vast amounts of data to support functions such as demand forecasting, procurement, logistics planning, and risk mitigation. By simulating human reasoning and adapting to dynamic contexts, AI agents play a vital role in modernizing SCM practices.

AI agents bring several advantages that contribute to the efficiency and resilience of supply chains, the strengths are:

- **Scalability.** Capable of handling large-scale, complex operations across global supply chains.
- **Automation.** Reduce manual workloads by managing repetitive tasks such as supplier evaluation, stock replenishment, and document generation [12].
- **Real-Time Optimization.** Enable adaptive responses to disruptions (e.g., weather, supplier delays) through continuous monitoring and dynamic re-planning.
- **Cost Efficiency.** Improve resource allocation, reduce excess inventory, and minimize operational costs [5].
- **Forecasting Accuracy.** Enhance demand and inventory predictions using machine learning techniques that recognize complex patterns in historical and real-time data.

Despite their potential, AI agents in SCM face several challenges:

- **Data Dependency.** Require access to large, high-quality, and well-integrated datasets. Fragmented or inaccurate data can compromise performance [12].

- **Interpretability.** Many models function as black boxes, making it difficult to explain or trust their decisions without proper explainability frameworks.
- **Deployment Complexity.** Integrating AI agents into existing enterprise systems can be technically challenging and resource-intensive.
- **Ethical and Legal Concerns.** Use of AI must align with regulatory standards and address issues related to bias, privacy, and accountability [5].
- **Computational and Energy Overhead.** Transformer-based agents, such as SustAI-SCM, require significant computing resources and training time, raising sustainability and accessibility concerns for small and medium enterprises [13].

To overcome these limitations and maximize their utility in SCM, future research and development efforts may focus on:

- **Industry-Specific Training.** Expanding datasets to cover more domains (e.g., pharmaceuticals, food logistics) for improved generalization [13].
- **Energy-Efficient Architectures.** Employing optimization techniques such as model distillation and quantization to reduce the computational footprint.
- **User-Centric Interfaces.** Developing tools that allow end-users to adjust optimization priorities (e.g., cost vs. sustainability) based on project-specific needs.
- **Explainable AI.** Promoting trust and accountability through transparent decision-making mechanisms.
- **Scalable Integration.** Designing flexible APIs and modular systems to ease integration into diverse SCM platforms.

To evaluate AI agents effectively in SCM applications, advanced techniques such as LLM-as-a-Judge have emerged. This evaluation method relies on leveraging powerful language models to assess and compare responses generated by AI agents for specific supply chain scenarios. The core idea is to use a pre-trained large language model (LLM) to act as an impartial judge, evaluating the quality, relevance, and correctness of agent responses in context [14].

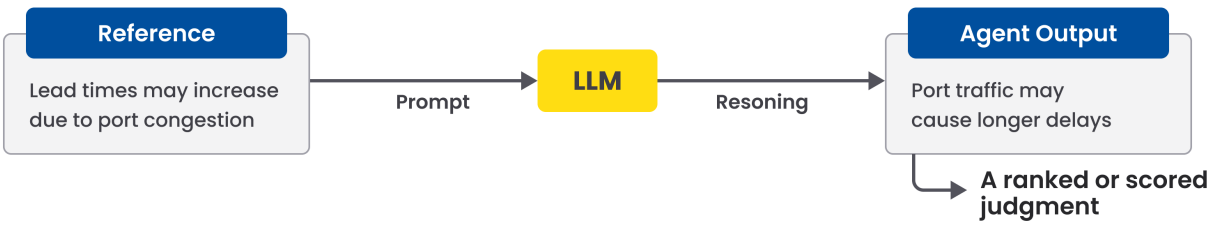


Figure 1.4: LLM-as-a-Judge technique overview.

To ensure robustness in this approach, several scoring metrics and embedding-based techniques are often integrated. Among the most common are BERTScore and Sentence-BERT (SBERT), which measure semantic similarity.

BERTScore evaluates the similarity between predicted and reference text using contextual embeddings from a BERT model. It computes precision, recall, and F1 scores based on cosine similarity of token embeddings [15]:

$$\text{Precision} = \frac{1}{|x|}\sum_{i=1}^{|x|}\max_j \cos(e(x_i), e(y_j)), \quad \text{Recall} = \frac{1}{|y|}\sum_{j=1}^{|y|}\max_i \cos(e(x_i), e(y_j))$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Sentence-BERT (SBERT), on the other hand, maps entire sentences to fixed-size dense vectors using a Siamese or triplet network structure. It then compares vectors using cosine similarity [16]:

$$\text{Score}_{\text{SBERT}}(x, y) = \cos(\text{SBERT}(x), \text{SBERT}(y))$$

These techniques provide a quantifiable measure of an agent's performance across tasks such as reasoning, consistency, and context comprehensionmaking them highly suitable for evaluating decision-support agents in supply chain contexts.

# CHAPTER 2

# COMPANY PRESENTATION: CEVITAL SPA

This chapter profiles CEVITAL SPA, Algeria's largest private industrial group and a key player in the country's economic landscape. Established in 1998 by entrepreneur Issad Rebrab, Cevital has grown into a diversified conglomerate spanning agro-industry, steel manufacturing, consumer electronics, glass production, and logistics. With 26 subsidiaries and operations across several continents, the group plays a dominant role in the national market while pursuing international expansion. Its headquarters are strategically located in the port city of Béjaïa, providing access to maritime trade routes and supporting large-scale industrial operations.

The chapter explores Cevital's organizational structure, with particular attention to operational divisions central to digital transformation namely, the Supply Chain and Information Technology departments. We examine the structure of its logistics network and review core digital tools, including ERP and TMS. While these systems offer a solid technological foundation, the analysis highlights limitations in integration, real-time data exchange, and the use of predictive analytics key elements for true AI readiness.

These findings provide a foundation for upcoming chapters that will explore AI strategies aimed at optimizing supply chain performance. By outlining both Cevital's industrial strength and untapped technological potential, this chapter sets the stage for intelligent, data-driven transformation.

## 2.1 Executive Summary

Cevital is the largest private industrial group in Algeria, founded by entrepreneur Issad Rebrab on May 2, 1998, with private capital of approximately 68.76 billion DZD. Structured as a joint-stock company (SPA), its major shareholders are Mr. Issad Rebrab and his children. Cevital is a diversified conglomerate encompassing 26 subsidiaries and employing over 18,000 individuals across three continents.

Headquartered at the eastern end of the port of Bejaia, Cevital benefits from a strategic location near the national road RN12, the port, and Bejaia's airport, enabling it to efficiently distribute its products domestically and internationally. This geographical advantage also strengthens its role in national economic development.

The group operates across multiple sectors, including:

- Agro-food industry (notably sugar, vegetable oils, and margarine)
- Steel and metallurgy
- Electronics and household appliances
- Maritime and logistics
- Real estate and services
- Automotive distribution

Cevital Agro-industrie, one of its main subsidiaries, is the largest private agro-industrial complex in Algeria and a leader in the Mediterranean basin and Africa. It has helped Algeria transition from an importer to an exporter of agro-food products and serves major global clients such as Coca-Cola, Kraft Food, and Danone.

Cevital's history is marked by continuous diversification and international expansion. The company grew from a metallurgy business to become a multi-sector conglomerate, leveraging modern infrastructure, strategic acquisitions, and strong export performance.

| | |
|---|---|
| 1998 | Founding of Cevital SPA (agro-industrial activities begin) |
| 2006 | Acquisition of COJEK; Creation of NUMIDIS and IMMOBIS |
| 2007 | Creation of SAMHA (SAMSUNG distribution) and MFG (flat glass production) |
| 2008 | Creation of Nolis (maritime transport), NUMILOG (logistics); expansion into flat glass market in Europe |
| 2009 | Sugar production scaled up to 1 million tons/year |
| 2010 | Start of sugar exports to Europe |
| 2013 | Acquisition of OXXO (France) and investment in ALAS (Spain) |
| 2014 | Acquisition of Brandt (France) and AFFERPI (Italy), former Lucchini Piombino |
| 2015–2020 | Strong export growth across Africa, Europe, Latin America |
| 2020 | Reached $3.5 billion in revenue |
| 2025 (goal) | Targeted revenue of $25 billion |

Table 2.1: Historical Advancements of Cevital

Cevital's evolution reflects a strong vision of becoming an integrated and influential player in regional and global markets. The group maintains its leading status by aligning strategic investments with international standards and national development goals.

## 2.2 General information about CEVITAL SPA

### 2.2.1 Company Activities

Cevital SPA is the parent company that encompasses a wide range of industrial and commercial activities across various sectors. Among these, the agro-industrial division holds a central role, accounting for approximately 80% of the total revenue (*chiffre d'affaires*) of Cevital Spa. With operations that span food processing, logistics, packaging, and beverage production, the agro-industry not only drives the group's economic performance but also positions Cevital as one of the most advanced and diversified private industrial groups in Algeria and the broader Mediterranean region.

Cevital Agro-Industrie forms the core of the group's operations and includes:

| **Activity** | **Brands / Products** | **Production Capacity** | **Export Info** |
|---|---|---|---|
| Sugar Production | Refined white sugar, Liquid sugar (industrial use) | 2,210,000 tons/year | Maghreb, Middle East (up to 900,000 tons/year) |
| Vegetable Oil Production | Fleurial Plus, Elio, Fridor | 570,000 tons/year | Maghreb, Middle East, planned expansion to Europe |
| Margarine and Vegetable Fats | Matina, Rania, Beurre Gourmant, Fleurial, La Parisienne, MEDINA Smen | 180,000 tons/year | Europe, Maghreb, Middle East |
| Beverage Production | Lalla Khedidja (water), TCHINA (juice) | 3,600,000 bottles/day | Domestic-focused; exports under development |
| Packaging (PET Bottles) | PET bottle manufacturing | 9,600 units/hour | No export activity |
| Logistics Support | Grain storage, unloading terminals | Terminal: 2,000 tons/hour | Serves domestic and regional logistics |

Table 2.2: Summary of Agro-Industrial Activities at Cevital

In addition to its core agro-industrial operations, Cevital has progressively diversified its activities across several key sectors, creating a broader industrial ecosystem and reinforcing its presence in both national and international markets.

One of the major areas of *diversification is steel and metallurgy.* Through subsidiaries such as **METAL SIDER**, Cevital is active in the production of construction steel and flat glass. These materials are essential for the construction and automotive industries, positioning the group as a significant player in Algeria's industrial base.

Cevital has also invested in the electronics and home appliances sector. This is exemplified by companies like **SAMHA**, which handles the assembly and distribution of Samsung products in Algeria, and BRANDT (France), a well-known manufacturer of household appli-

ances. The group also operates MFG, which specializes in flat glass production for various applications across both Algerian and European markets.

In terms of logistics and maritime transport, Cevital has developed a solid infrastructure through subsidiaries such as **NUMILOG** and **NOLIS**. These entities manage the internal and external logistics chain, including road and sea freight, and support the distribution needs of Cevital and its partners across different regions.

The company is also present in retail and distribution, mainly through **NUMIDIS**, which operates supermarket chains, and **IMMOBIS**, a real estate and commercial development company. These ventures help create a direct link between industrial production and consumer markets, while also expanding the group's investment footprint.

Lastly, Cevital is involved in several real estate and infrastructure development projects, aimed at supporting its logistics and industrial expansion. These strategic developments reflect the group's long-term vision to build an integrated and self-sustaining industrial network that supports economic growth across Algeria.

### 2.2.2 Company Main Sectors

Cevital's agro-industrial operations are primarily anchored in the Wilaya of Béjaïa, specifically around the port zone, which serves as the company's principal production and logistics center. This site concentrates the bulk of industrial activity and infrastructure, making it the strategic core of Cevital's agro-business. In comparison, the installations located in El-Kseur (Béjaïa) and Tizi-Ouzou function as specialized branches that support the main ecosystem by focusing on beverage and mineral water production respectively. Together, these sectors form an integrated and scalable industrial network serving national and regional markets. THe main installations are:

- **Béjaïa (Port Zone and Surroundings):** This central sector constitutes the heart of Cevital's agro-industrial platform. Situated at the port and its immediate surroundings, it integrates refining, manufacturing, packaging, storage, and maritime logistics at a large scale. The proximity to port infrastructure enables efficient importation of raw materials and exportation of finished goods, giving Cevital a strong logistical advantage.

- **El-Kseur (Béjaïa):** The town of El-Kseur, in the Béjaïa region, hosts one of Cevital's key diversification sites: the **COJECK** unit, a modern beverage processing facility. Dedicated to producing *fruit juices* and *soft drinks*, this plant supports Cevital's strategy of expanding its consumer goods portfolio and reinforcing its position in the regional market for fast-moving consumer products. With a production capacity of around **14,400 tons per year**, the COJECK plant reflects Cevital's focus on industrial scale, efficiency, and competitive output. It meets growing local demand while contributing to the economic development of the El-Kseur area and adding value to the group's agro-industrial ecosystem.

- **Tizi-Ouzou (Agouni Gueghrane – Djurdjura Mountains):** Located in the heart of the Djurdjura mountain range, the Agouni Gueghrane unit is a major facility dedicated to the bottling of natural mineral water under the well-known Lalla Khedidja brand. This site plays a significant role in supplying the national market with high-quality bottled water and reflects Cevital's commitment to leveraging local natural

resources sustainably. With an impressive daily production capacity of 3 million bottles, the plant stands as one of the largest in the country, reinforcing Cevital's position in the beverage sector and contributing to both regional development and national distribution efficiency.

- **National Distribution and Support Functions:** Beyond production activities, Cevital ensures national market coverage through subsidiaries such as *NUMIDIS* and *NUMILOG*. These entities manage distribution, logistics, and various support services across the country, guaranteeing efficient nationwide coverage for both domestic sales and export operations.

## 2.3 Organizational structure

The organizational structure of Cevital is illustrated in the figure below, providing a comprehensive overview of its main subsidiaries, operational divisions, and their interrelations. This visual representation helps clarify how the group is structured and coordinated across its diverse business sectors.

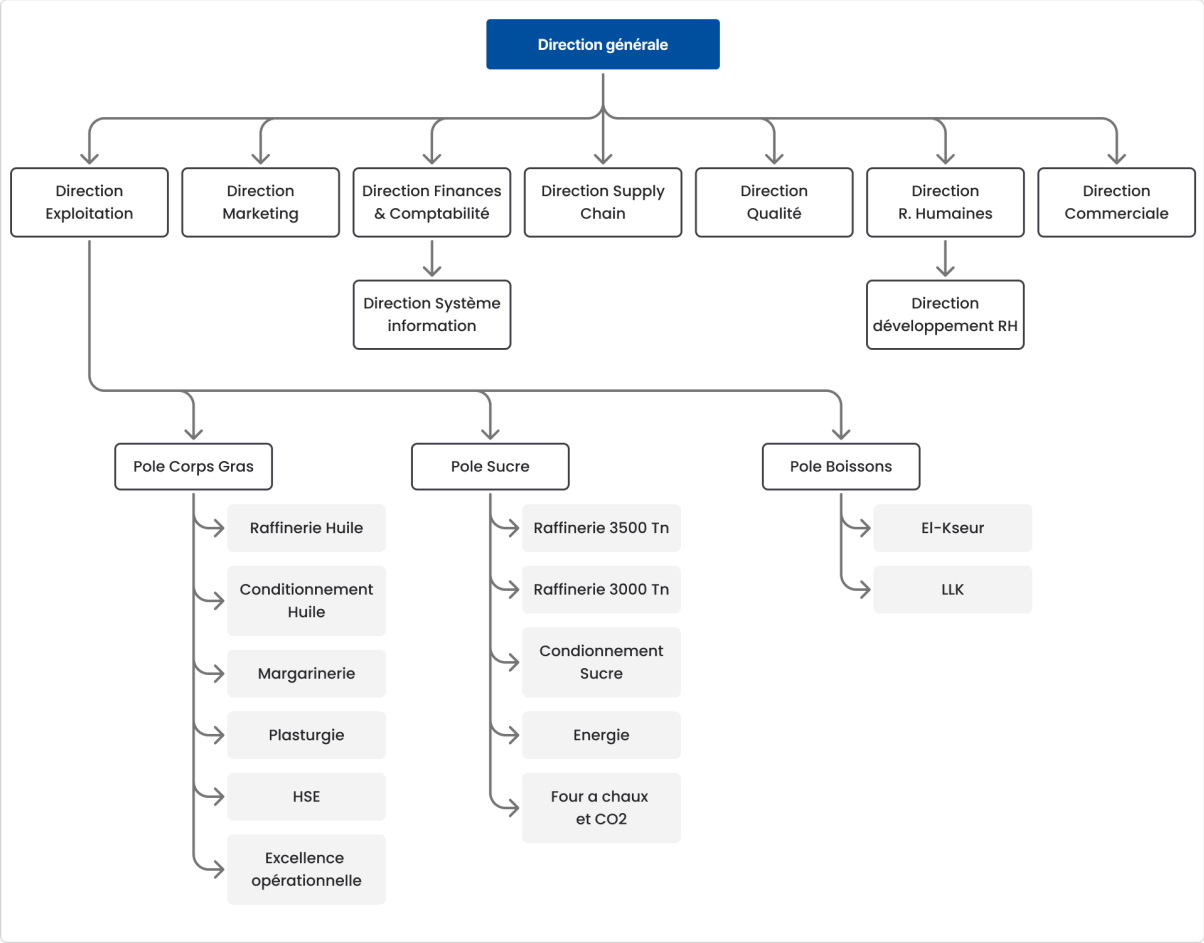


Figure 2.1: Cevital organisational structure

## Information System Department

The Direction du Système d'Information (DSI) is the central pillar of Cevital's digital transformation strategy, playing a pivotal role in the integration of artificial intelligence and innovative technologies across the enterprise. It is responsible for deploying and maintaining the information systems that underpin the company's operational performance and long-term competitiveness. The DSI is composed of four specialized sub-departments: the Information System unit, which oversees system performance, anticipates future technological needs, and aligns IT development with corporate strategy; the Information Technology unit, tasked with applying emerging technologies to solve complex business challenges and ensure alignment between IT tools and enterprise objectives; the Applications Métiers unit, which focuses on the automation of core business operations including HR, finance, and production while managing internal applications and databases; and finally, the Digital Transformation unit, which drives innovation through the adoption of AI and data analytics, ensuring Cevital remains agile and competitive in a fast-evolving digital landscape.

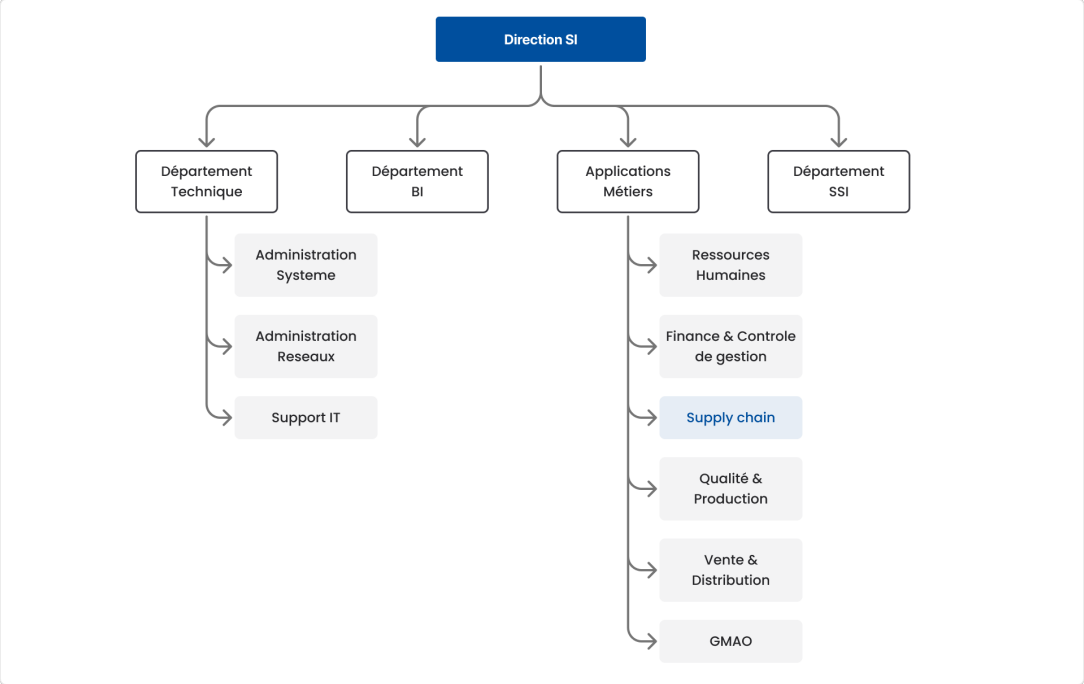


Figure 2.2: Information System departement structure

In the Information Systems Department, there is a dedicated sub-department called the Supply Chain Management Department. This team is responsible for managing the overall logistics flow across Cevital, from procurement to product delivery. It ensures coordination with production units, distribution platforms, and retail outlets. Its core responsibilities include planning and scheduling product deliveries, optimizing storage and transportation through software systems, and ensuring product availability across 18 regional delivery centers (CLR). Given its central role in operational efficiency, this sub-department is an ideal candidate for the integration of AI-powered solutions such as demand forecasting, inventory optimization, route planning, and intelligent reporting.

## 2.4 Supply chain structure

Cevital's supply chain management (SCM) department, created in 2013, plays a pivotal strategic and operational role within the agro-industrial division. As a structure directly linked to the General Directorate, SCM ensures the coordination of core functions from raw material procurement to final product delivery. Its mission is to guarantee product availability, cost efficiency, and customer satisfaction while optimizing the company's competitiveness.

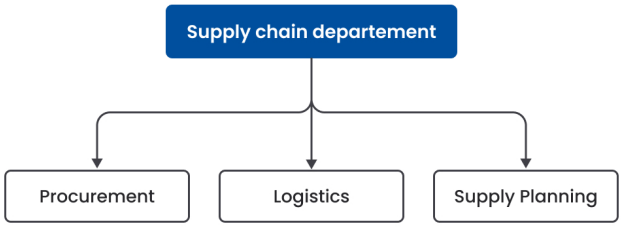


Figure 2.3: Supply chain departement structure

1. **Procurement**: Responsible for sourcing and supplying raw materials on time, at optimal cost and quality.
2. **Logistics**: Manages warehousing, internal transport, and distribution across national and regional centers.
3. **Supply Planning**: Aligns sales forecasts with production and inventory planning, including detailed scheduling and reporting.

Key challenges include a lack of coordination with production, limited integration of transport operations currently handled by Numilog, and several procurement risks such as nonconformities, delivery delays, and delays in information flow.

Cevital's planning system follows a layered structure that begins with the **RFC** (Référentiel des Forecasts Commerciaux), followed by the **PDP/PDA** and **DRP** levels, all originating from the commercial department. The final layer is execution, which involves close coordination between Numilog and internal logistics services. This structure ensures a sequential and hierarchical approach to planning, moving from strategic forecasting to operational execution. To support these processes, Cevital relies on *SAGE 1000* for planning and production management, *COSWING* for procurement activities, and uses *Excel and Outlook* as complementary tools for coordination and communication.

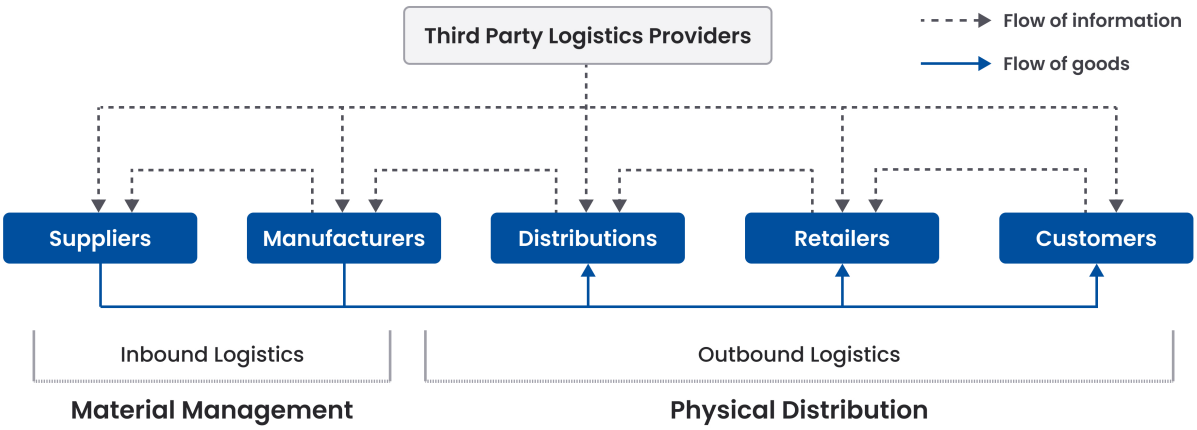


Figure 2.4: Data and goods flow in the Cevital's supply chain lifecycle

### 2.4.1 Logistics infrastructure

Cevital agro-industry possesses a highly developed logistics backbone, essential for handling high-volume flows of sugar, oil, and margarine. Its *logistics system is segmented into infrastructure, tools, and flow management:*

#### Storage Capacities:

Cevital has established extensive storage infrastructure to sustain its large-scale production and distribution operations. This infrastructure includes both internal and external storage capacities. Internally, the group manages white sugar silos with a capacity of **120,000 tons**, raw sugar storage facilities holding up to **200,000 tons**, as well as cold and ambient storage capable of accommodating **1,400 and 1,600 pallets** respectively. In addition to these internalAssets, Cevital also utilizes external warehouses such as **ICOTAL** with a storage capacity of 2,442 tons, JUTE with 5,130 tons, and ENAEB with 15,955 tons, enhancing the group's ability to manage and buffer its supply chain flows efficiently.

#### Production Loading Capacities

Production loading capacities across Cevital's facilities enable high daily output of sugar, oil, and margarine products.

| **Product** | **Loading Capacity** |
|---|---|
| 1kg Sugar | 1,300 tons/day |
| 5kg Sugar | 120 tons/day |
| 50kg Sugar | 1,200 tons/day |
| Oil | 1,200 tons/day |
| Margarine | 600 tons/day |

Table 2.3: Production Loading Capacities

### Transport and Distribution Network

Cevital's distribution system is structured around a broad national network that leverages both internal logistics capabilities and strategic external partnerships. The company operates 18 Regional Logistics Centers (CLR) spread across the country, in addition to key depots and distribution platforms located in major cities such as Bejaïa, Algiers, Oran, and Sétif. Its transport fleet is composed of a mix between company-owned trucks and subcontracted carriers, all coordinated through Numilog. To ensure efficient planning and operations, Cevital integrates its distribution activities with a Transport Management System (TMS), allowing for better tracking, routing, and delivery performance across its supply chain.

### Warehouse and Flow Management Tools

Cevital deploys advanced warehouse management systems designed to align inventory operations with both production forecasts and sales demand. These systems enable real-time tracking of inventory levels and movements, ensuring continuous visibility and responsiveness across the supply chain. They are interconnected with production units and commercial planning tools, creating a synchronized flow of goods based on forecasted needs. Additionally, performance is monitored using key performance indicators such as stock rupture rates, particularly those caused by document-related errors, which supports continuous improvement and operational reliability.

| **Logistics Element** | **Details** |
|---|---|
| Sugar Storage (white + raw) | 320,000 tons combined |
| Cold & Ambient Storage | 3,000 pallets combined |
| Regional Storage (external) | ICOTAL, JUTE, ENAEB ( 23,000 tons total) |
| Loading Capacity – Oil | 1,200 tons/day |
| Loading Capacity – Sugar | 1,300 / 120 / 1,200 tons/day (1kg/5kg/50kg) |
| Fleet | Internal trucks + Numilog-managed external transport |
| Regional Logistics Centers | 18 CLR across Algeria |
| Systems Used | SAGE 1000 (planning), COSWING (procurement), TMS (transport) |

Table 2.4: Summary of Logistics Capacities and Infrastructure

## 2.5 Technological infrastructure

### 2.5.1 Existing Data Systems and Architecture

At CEVITAL, the data infrastructure forms the backbone of business operations, especially within the agro-industrial sector. It ensures reliable access to critical information, enables traceability across production and logistics processes, and supports both operational and strategic decision-making.

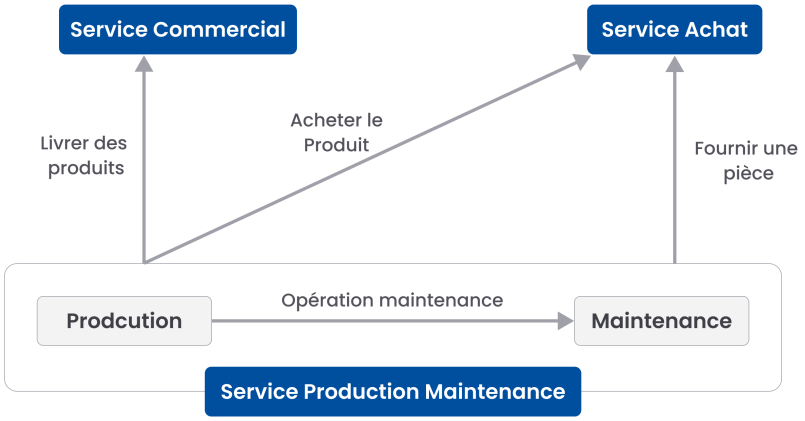


Figure 2.5: Overview of the communication between systems in cevital

The ecosystem is composed of an array of interconnected systems designed to meet specific functional needs:

- **Enterprise Resource Planning (ERP – Sage X3)**:
  Centralizes management of accounting, procurement, inventory, and sales. Integrates workflows across departments to ensure coherence and traceability.
- **Customer Relationship Management (CRM)**:
  Supports structured client management, tracks interactions, enhances customer satisfaction, and informs marketing strategies.
- **Computerized Maintenance Management System (GMAO – COSWIN 8i)**:
  Organizes preventive and corrective maintenance, tracks equipment lifecycles, and manages spare parts inventory to ensure industrial reliability.
- **Transport Management System (TMS – Transwide)**:
  Oversees transportation planning, optimizes delivery routes, monitors real-time shipments, and provides logistic performance KPIs.
- **Private Cloud Storage and Office Tools**:
  Internal document servers enable centralized data sharing. Productivity tools such as Microsoft Office, VMware, and virtual collaboration platforms support day-to-day operations.

- **Big Data Systems and Data Lakes**:
  CEVITAL is deploying Big Data technologies with a data lake architecture to accommodate structured, semi-structured, and unstructured data, paving the way for future advanced analytics and AI developments.

### 2.5.2 Network and Hardware Architecture

The technological foundation of CEVITAL ensures secure, high-performance connectivity across all its operational and administrative sites. At the heart of the network infrastructure is a ***Cisco Catalyst 4507R*** switch, which functions as the central backbone, managing inter-VLAN routing and handling enterprise-wide data traffic. Across the various operational zones, ***Cisco Catalyst 2960/2950*** switches are deployed at the access layer to connect endpoints and IoT devices, while ***Cisco 2900*** Series routers ensure reliable WAN connectivity between different company sites.

To protect the entire infrastructure, *dual Palo Alto firewalls* provide robust security, including traffic filtering, internet access control, and network segmentation. A comprehensive wireless network offers full WiFi coverage across facilities, enabling mobile-enabled operations and real-time monitoring. Critical infrastructure is hosted within a secured, climate-controlled data center, which houses the company's servers, telecommunications systems, and core network appliances ensuring both performance and physical security.

For external site communication, CEVITAL uses a *fiber optic link* to connect major hubs like Béjaïa and Algiers, while remote locations such as El Kseur, Tizi Ouzou, and Constantine rely on VSAT satellite links. This hybrid setup ensures uninterrupted connectivity across all regions, supporting the company's digital operations regardless of geographic constraints.

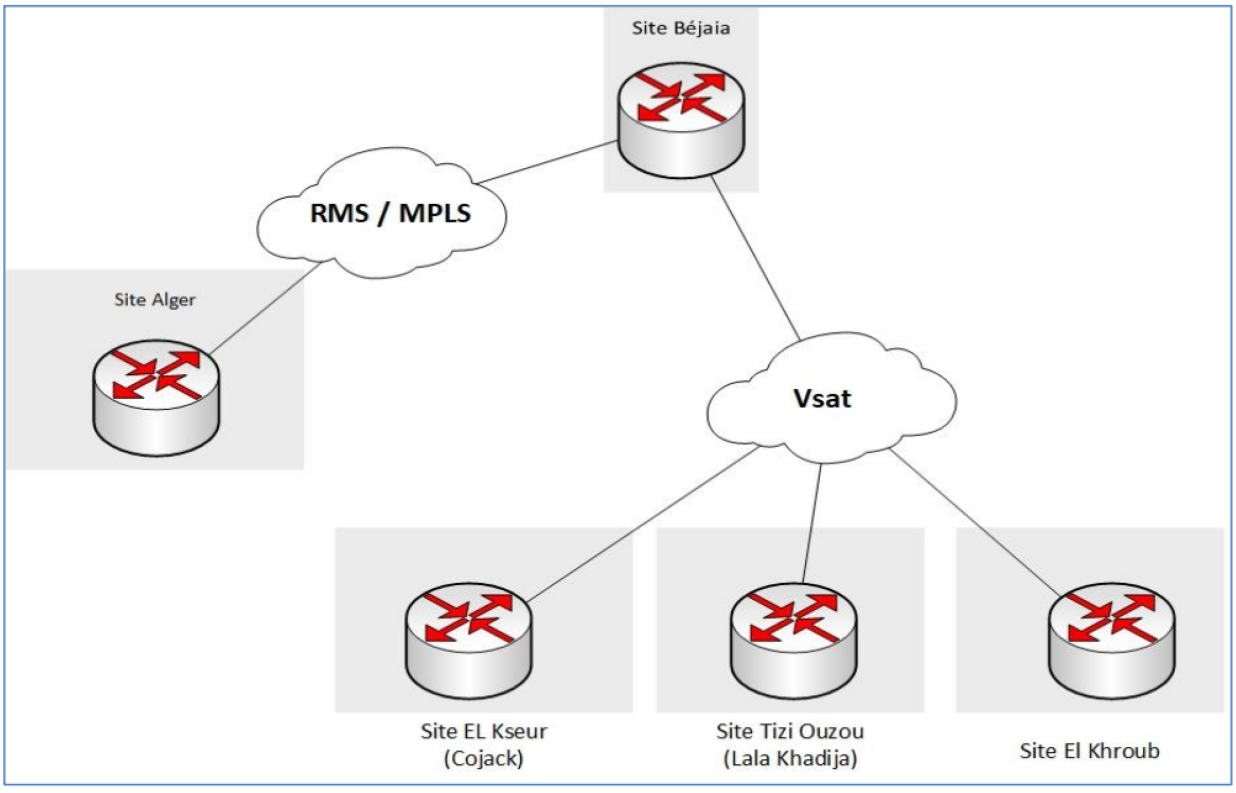


Figure 2.6: WAN Architecture in cevital intra System

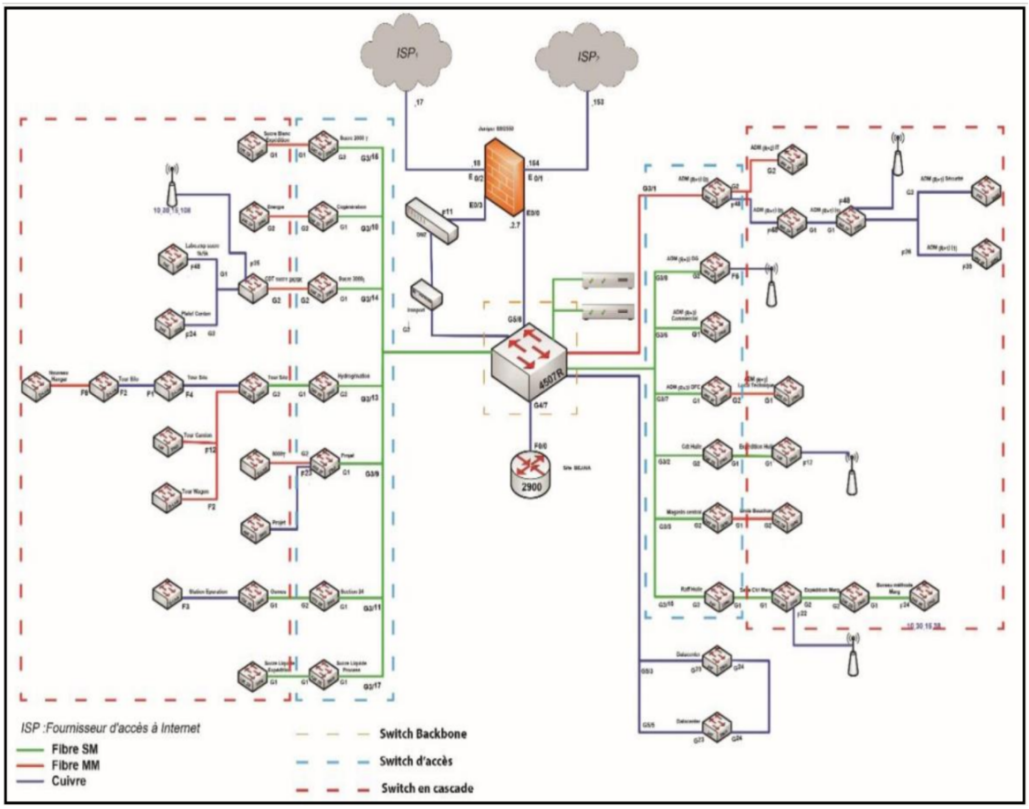


Figure 2.7: Network and Hardware Architecture of CEVITAL

## 2.6 Toward AI Readiness

While CEVITAL has made considerable investments in modernizing its data and technological infrastructure, current system integrations are still primarily optimized for operational continuity. Real-time data harmonization across services and advanced predictive analytics are still in development.

Given the ongoing deployment of Big Data environments and the adoption of a *data lake* architecture, CEVITAL is now at a critical stage in its digital transformation. To fully unlock the potential of Artificial Intelligence across its operations, the company must make a strategic shift toward consolidating and harmonizing data pipelines across departments, ensuring that data flows are unified, standardized, and accessible. This should be accompanied by the implementation of intelligent data governance and data quality frameworks that guarantee the reliability, consistency, and usability of the data. Building on this foundation, scalable AI solutions can then be designed and deployed across key domains such as supply chain, production, and logistics, with the aim of enhancing efficiency, agility, and decision-making. Finally, to ensure long-term success and adaptability, it is essential to establish continuous monitoring and learning processes for AI models, allowing them to evolve alongside the business and remain aligned with operational realities.

Our thesis research is precisely positioned to address these themes: identifying and proposing optimal solutions for data management enhancement, integrating AI into CEVITAL's operations, and establishing monitoring frameworks to ensure robustness, learning, and sustainability of AI systems.

## 2.7 Position of company problems

As one of Algeria's most prominent industrial groups, CEVITAL has consistently demonstrated a strong ambition to modernize its operations especially within the agro-industrial sector. Major strides have already been made like the Implementation of advanced ERP systems, Deployment of data lakes and big data infrastructure and investing in digital transformation.

But Today's Reality Is Different, Despite these efforts, the global industrial environment continues to evolve rapidly. And the message is clear "**Digitization alone is not enough anymore.**", To remain competitive, CEVITAL must now start making data-driven decisions, invest in new technologies to ipmrove existing operations and do it as fast as possible.

Despite having extensive digital tools (ERP, TMS) and large data repositories, CEVITAL still faces major obstacles to effective AI integration, especially in logistics. Through in-depth analysis, we aim to identify the **primary pain points** and define actionable strategies for enabling intelligent transformation.

1. ***Lack of functional AI integration:*** While data is collected and stored in a centralized architecture, it is rarely exploited by AI systems to automate or enhance logistics processes such as routing, forecasting, inventory optimization, or anomaly detection.

2. ***Insufficient interoperability between systems:*** Many logistics processes remain semi-manual or isolated, making it difficult to develop AI models that depend on synchronized real-time data streams.

3. ***Absence of AI-readiness indicators:*** The company lacks a formal AI-readiness framework or maturity model to assess how existing data and infrastructure can be mobilized for machine learning or multi-agent systems.

4. ***Underexploited predictive potential:*** Although TMS system exist, it is not currently connected to AI modules that could predict delays, recommend alternatives, or adapt planning in real-time.

## 2.8 Strategy of our thesis to adress problems & find the right solutions

This thesis is not about designing a perfect, ready-made system for CEVITAL. Instead, it serves as a strategic and technical response to the urgent need for intelligent transformation within the company's logistics operations. Rather than focusing on superficial digitalization, our work goes one step further: it explores how AI can be introduced gradually and meaningfully, starting with the most accessible and high-impact use case report automation based on historical transport data. This allows us to bring AI into real operations without waiting for full AI-readiness or perfect data conditions.

The project addresses several of CEVITAL's core pain points. First, it introduces functional AI integration by embedding a locally hosted large language model (LLM) into the reporting process using LangChain and pandas agents. This tackles the issue of underused data and transforms static datasets into dynamic insights, accessible by simple prompts. Second, by merging distributed sources into a single structured dataset and linking them to the AI agent, the project partially solves interoperability challenges paving the way for future multi-tool integration, including ERP and TMS systems.

Finally, the system serves as a practical foundation for AI-readiness. It helps expose current limitations in data quality and structure, which can inform future improvements in governance and infrastructure. It also delivers a working prototype that CEVITAL can build on starting with automated report generation and eventually extending to more complex tasks like anomaly detection, delay prediction, or autonomous decision-support. In this way, the thesis not only provides a proof of concept but also a realistic roadmap for integrating AI into logistics one that aligns with the company's strategic ambition to lead, not follow, in digital transformation.

# CHAPTER 3

# EXPLORATORY DATA ANALYSIS

This chapter examines the critical process of evaluating and preparing CEVITAL's transport data for analysis. Focusing on order records from their Transport Management System (TMS), we investigate the challenges and opportunities present in the company's supply chain data infrastructure.

Our exploration begins with assessing data quality and structure, followed by methods to transform raw operational records into analyzable information. The chapter documents the practical realities of working with industrial datasets, where theoretical ideals often meet real-world constraints.

Through this process, we establish the foundation necessary to bridge CEVITAL's current data capabilities with its ambitions for AI-powered supply chain optimization. The insights gained here inform both immediate reporting improvements and longer-term digital transformation strategies.

## 3.1 Data Collection Process

No two people will disagree that CEVITAL's data ecosystem is such a massive and complex one. For the scope of this thesis, we decided to focus on a specific and strategic segment of data within the supply chain: transport order data. This type of data is the most tracked and accessible across the company's logistics operations, and it forms the backbone of the transport activity within CEVITAL.

The primary source of this data is the Transport Management System (TMS – Transwide). This system records and manages all transport operations, including details about the transported products, associated companies, loading docks, scheduling, and shipment statuses. In other words, this dataset captures everything that happened in the physical flow of goods making it the best entry point to experiment with AI-powered reporting and automation.

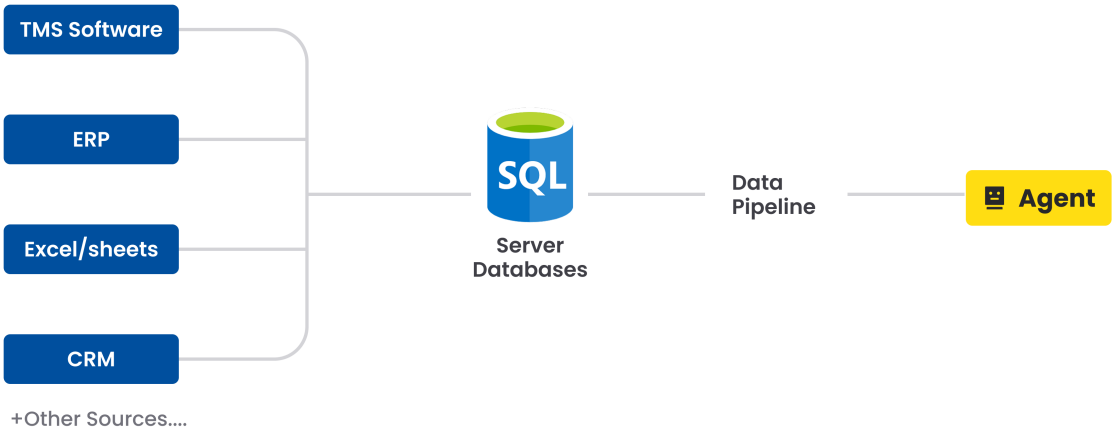


Figure 3.1: The main data sources used in the thesis

## 3.2 Data Quality and Limitations

When evaluating whether a dataset is ready for analysis or AI integration, several fundamental factors must be taken into account. In our study, we applied a structured evaluation framework to assess the data collected from Cevital's transport and logistics systems, focusing on key dimensions that influence usability, reliability, and future scalability.

First, we looked at completeness, which measures whether all the required data points are present. Missing values, incomplete records, or entirely empty fields can significantly limit the reliability of insights derived from the data. Alongside that, we considered accuracy, which refers to how well the data reflects actual operations. Outdated or incorrect entries can lead to flawed analyses and undermine any predictive or decision-support models that might be built on top of the data.

We also evaluated consistency across tables and systems, checking for discrepancies such as vendor names spelled differently in different sources. Timeliness was another important factordata entry lags or unsynchronized updates between systems make real-time forecasting or reporting difficult. Uniqueness refers to the presence of duplicate records, which can distort metrics if not identified and consolidated properly. Relevance was equally important; we found that not all fields served a meaningful analytical purpose, and some existed without practical value for our reporting goals. Finally, accessibility and format played a significant role. Many datasets were fragmented, stored in inconsistent formats, or required complex joins to extract value conditions that are not optimal for AI-driven workflows.

In the context of our study, we focused on the transportation dataset used within Cevital's logistics operations. The following observations summarize the specific data challenges encountered.

Throughout our work, we encountered several challenges related to the structure, accessibility, and readiness of Cevital's data for AI-driven analysis. While the company possesses a large volume of operational data, several limitations both systemic and technicalsignificantly impacted our ability to leverage this data for advanced analytical tasks.

The data available across Cevital is undeniably rich. However, the absence of a central data integration platform such as a data lake or real-time pipeline made it extremely difficult to identify what data exists, where it resides, and whether it could be used together. This

lack of unified access led to repetitive checks, manual exploration, and sometimes overlapping sources, hindering the segmentation and consolidation of relevant datasets.

Most of the accessible data, including our main dataset (transportation data), was historical and procedural. It primarily served as a log of operations rather than a resource designed for prediction. For instance, transport records typically followed a fixed pattern: pick product, assign vendor, select date, and ship. There was no contextual data such as road conditions, traffic, or contract deviationsfactors that are crucial for building accurate AI forecasting models.

Many supply chain operations at Cevital are not fully digitized. There is limited real-time tracking, minimal automated data collection, and insufficient documentation. These limitations restrict both operational understanding and the potential for advanced analysis.

One of the biggest barriers was the lack of metadata. Descriptions of fields were missing, column meanings were not explained, and there was no internal guide to help us interpret the data structure. This forced us to spend a significant amount of time simply trying to understand what each field represented, delaying preprocessing and modeling phases.

After evaluating all available data sources, we determined that the most usable and accessible dataset for our thesis was the **transportation data**, primarily managed through the *Transport Management System (TMS)* and stored in an *SQL Server* database. A large portion of operational data, particularly daily entries, is still maintained in **Excel files**, limiting its integration with centralized systems. Additionally, access to certain datasets was restricted due to **confidentiality policies**.

From a quality perspective, our assessment included the following dimensions:

- **Consistency:** We identified two separate sources for transport order data: one from the SQL Server and another from the Transportation Management System (TMS). These sources showed inconsistencies in structure and content, which introduced ambiguity. After a detailed comparison, we chose to retain the SQL Server data due to its completeness and closer alignment with actual transport operations. The TMS data was excluded as it did not offer additional value for our analysis.

- **Timeliness:** Although we did not experience significant delays in data availability, frequent modifications after the initial entry made it difficult to determine the most accurate version. For instance, discrepancies between scheduled and actual loading dates complicated any time-sensitive analysis.

- **Uniqueness:** Some transport orders appeared duplicated but had different IDs. These cases generally represented single large orders that were intentionally split into smaller segments for logistical reasons. We verified and retained them as valid entries.

- **Relevance:** Overall, the data aligned well with our reporting and analytical goals. However, some fields lacked clarity or were sparsely populated. Despite the absence of contextual metadata, we were able to extract meaningful insights through manual interpretation and targeted filtering.

- **Accessibility and Format:** The structured storage of data in SQL Server facilitated access and preliminary cleaning. Data types were mostly consistent. Nevertheless, many fields used foreign key IDs (e.g., for product types or order priorities) without corresponding descriptive labels, requiring additional joins with reference tables to fully interpret the dataset.

Throughout the datasets, we observed frequent null values, missing context, and inconsistent entries. These issues limited the usefulness of the data to basic reporting tasks rather than advanced AI use cases such as automation or forecasting.

In the end, we were able to extract a reliable and well-structured dataset from the SQL Server system, which enabled meaningful analysis. However, the broader systemic challengessuch as the absence of centralized integration platforms, lack of metadata, and low digitizationstill present major obstacles to scalable AI deployment. Addressing these will be critical for future success in data-driven decision-making at Cevital.

## 3.3 Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in any data-driven project. It refers to the process of analyzing and summarizing datasets to discover patterns, detect anomalies, test assumptions, and better understand the structure and meaning of the data. The main goal of EDA is not to build models, but rather to gain insights that will guide data preparation, model design, feature selection, and informed decision-making throughout the pipeline.

In our case, EDA was especially critical due to the noisy, incomplete, and distributed nature of the data we collected from Cevital's transport systems. Since AI agents are not yet capable of handling raw and low-quality data on their own, this step became essential to manually clean, validate, restructure, and prepare the data for future automation and modeling tasks. It also served as a collaborative checkpoint to align technical and domain-specific knowledge. Without this foundational work, downstream processes like inference, analysis, or reporting would have been severely compromised.

Key objectives of the EDA in this project included:

1. Understanding the data structure and content, so we could later document it clearly for both human and machine consumption.

2. Preparing the data for the AI agent. Our EDA helped us identify and fix formatting issues, detect missing or inconsistent values, and eliminate unnecessary columns. This preparation step was crucial to ensure that the agent could later interpret the data accurately when generating reports.

3. Designing a future pipeline capable of performing automated cleaning on newly ingested data, allowing us to move toward automated daily reporting powered by the AI agents.

4. Many fields in the dataset were either unclear or stored as foreign keys. Through EDA, we explored and interpreted the content and role of each column, which helped us later write precise instructions for the AI agent. This semantic clarification is really important for reliable report generation.

5. Creating an evaluation dataset. A big part of our thesis involved testing the AI agent's reporting abilities. To do that, we needed a benchmark dataset with predefined user questions and correct answers. EDA allowed us to generate these questions in a consistent and representative way, based on real patterns and structures in the data.

### 3.3.1 Analysis Execution & Observations

For our exploratory data analysis (EDA), we relied primarily on Python as the main programming language due to its flexibility and robust ecosystem for manipulating, visualizing, and transforming transport-related data. The core tools used included **pandas** for data manipulation, **NumPy** for numerical operations, **matplotlib** and **seaborn** for data visualization, and **Jupyter Notebooks** as our interactive analysis environment. We also used **GitHub** for version control and collaborative tracking of changes.

Unlike working with clean artificial datasets, real-world industrial contexts often involve large volumes of noisy, incomplete, or irrelevant data before reaching anything usable. That was exactly our case. We initially had access to 21 raw datasets, each varying in structure, quality, and relevance. After thorough inspection, cleaning trials, and multiple iterations of analysis, we discovered that datasets such as timestamp logs, TMS SYNERGY exports, and Sage CoSoft extracts although initially promising were either redundant, outdated, or not aligned with our reporting needs. These trial-and-error steps were time-consuming but essential to isolate what we now define as the "core reporting dataset."

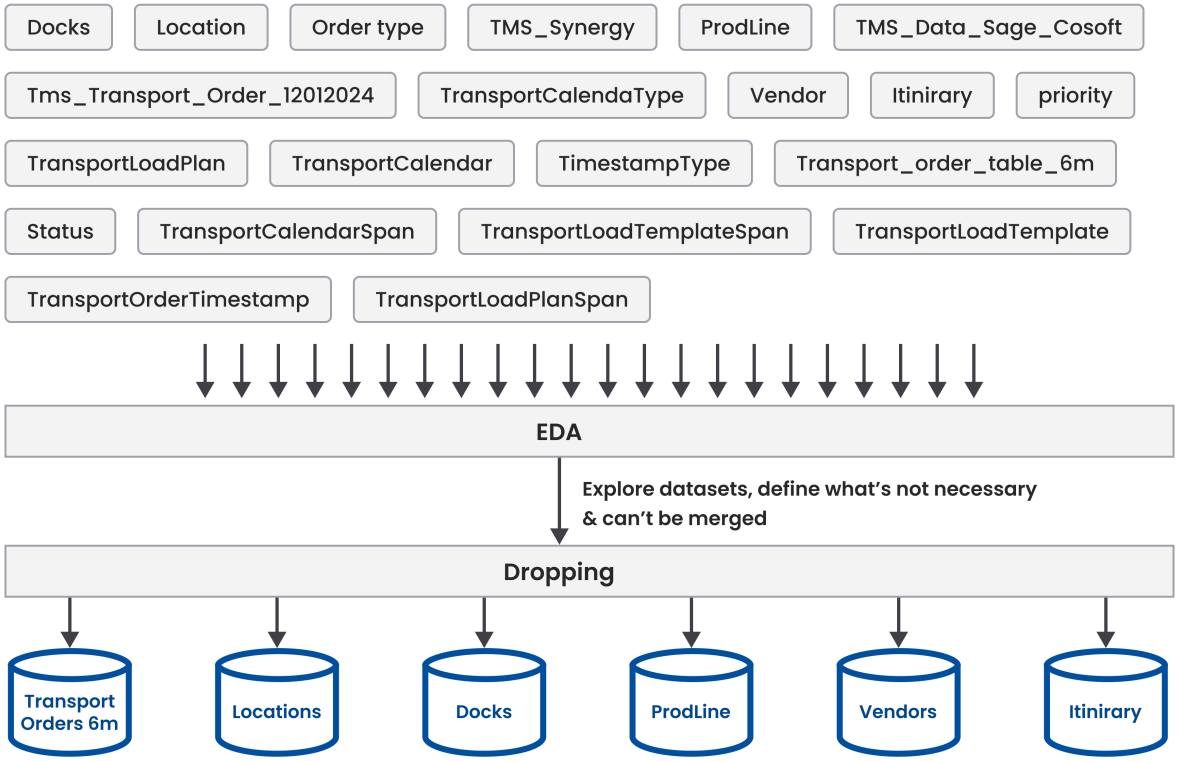


Figure 3.2: Datasets Selection Process

In the end, we carefully selected six datasets that were most relevant to the goals of AI-powered reporting and analysis. Rather than blindly integrating everything, we focused on those that matched the AI agent's target schema and supported meaningful business insights. This filtering step was critical to ensure high data quality, reduce redundancy, and streamline the data pipeline. These selected datasets were then thoroughly cleaned, aligned, and merged into a single, unified structure optimized for downstream AI processing. This selection process was a foundational component of our EDA and directly contributed to the reliability of the AI reporting system.

Here are the main datasets we used during EDA:

- **transport_orders_6_months.csv**: This was our primary dataset. It contains most of the transport operation data: loading times, vendor names, destinations, priorities, statuses, etc. We invested the most effort in cleaning, enriching, and integrating this dataset, and it shaped the logic and prompt design of the AI agent.
- **docks.csv**: Contains the list of all docks used by Cevital for storing products. This dataset includes dock characteristics and is critical for identifying where each transport operation started or ended.
- **location.csv**: A foreign key reference table with all predefined locations (warehouses, factories, etc.). It was useful for later building location-based reports and will be valuable in future applications like route optimization.
- **ProdLine.csv**: Stores product-related data for client orders, including product names and quantities. It bridges the gap between production and transport, helping clarify the relationship between what was made and what was delivered.
- **vendors.csv**: Includes the list of logistics partners, such as subcontractors and key players like NUMILOG. This table was necessary to analyze vendor performance and build vendor-specific reports.
- **itinirary.csv**: A static mapping of predefined itineraries, with origin-destination pairs that standardize communication between internal teams and transport managers.

We started exploring each dataset separately. Since no documentation was available, we manually reviewed all columns to understand their purpose, which took a considerable amount of time. Once the structure was understood, we handled missing values either by dropping incomplete rows or columns, or by imputing them using calculations or foreign key relationships. We also inspected data types and performed necessary conversions such as transforming string-formatted dates into datetime objects, and correcting columns marked as generic objects into appropriate types like string, integer, float, or boolean.

Unnecessary columns were removed to retain only those relevant for our AI use case. We then cleaned column values by correcting typos, handling outliers, and standardizing formats. This step was crucial due to the inconsistent and noisy state of the raw data.

Several challenges required manual intervention. We encountered redundant and duplicated data across multiple datasets, introducing noise and confusion during merging. Some key columns had high missing value rates, leading us to eliminate certain fields to preserve reliability. The absence of internal documentation or metadata significantly hindered visibility and interpretation. Additionally, many datasets had unclear purposes or structural inconsistencies, requiring manual analysis to determine their relevance.d.

Finally, we merged all datasets using the `TransportOrderID` key, with supporting foreign key tables like ***status*** and ***Docks*** also integrated using left joins. During this merge phase, we encountered challenges with missing values and foreign key mismatches, which we resolved using smart imputation and fallback logic. The result was a single, AI-ready dataset that laid the foundation for report generation and agent-based analysis.

You can view the EDA process in the diagram below, which visually summarizes the key steps from dataset selection to final merging and cleaning.

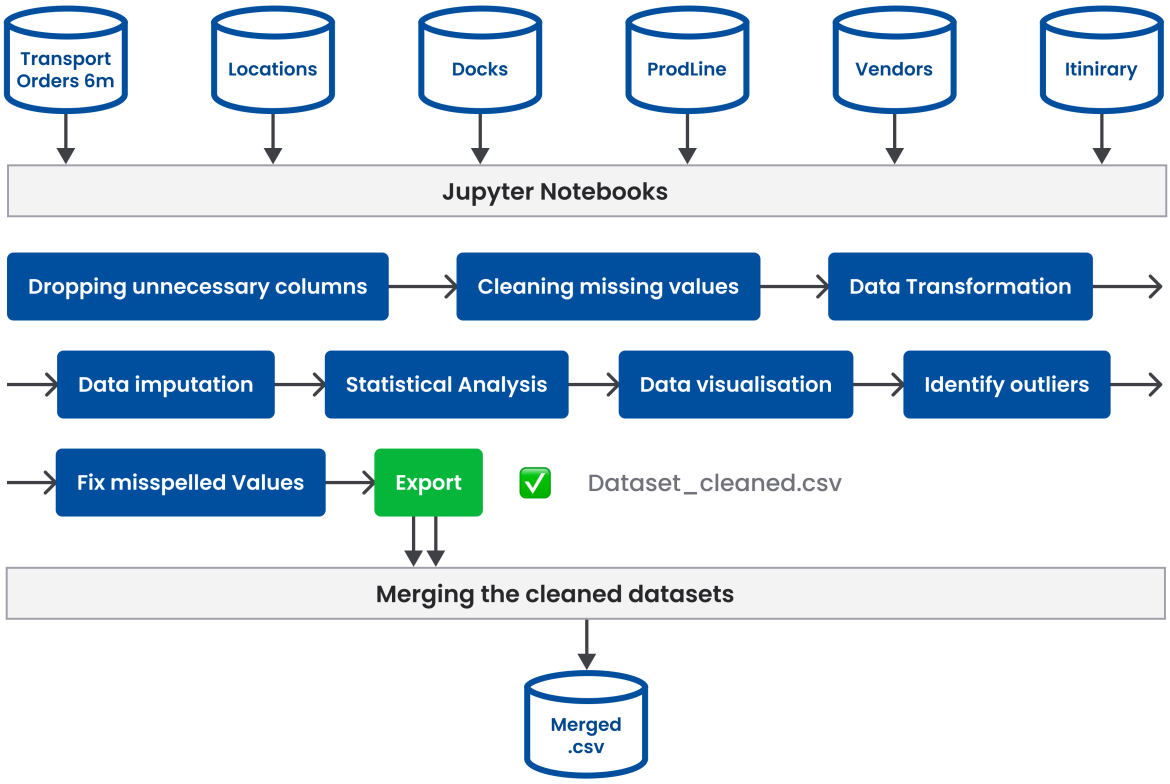


Figure 3.3: EDA Process Overview

The schema of the data exported from the TMS is highly normalized, as expected from enterprise-grade systems. It follows standard relational database design principles to ensure performance and data integrity. Tables are separated by topic, with heavy use of foreign keys to reference auxiliary information. While this setup is ideal for transactional use, it's not optimal for running dynamic analyses or powering an AI agent, which would need direct access to multiple aspects of the data at once.

To solve this, we merged the necessary tables into a single flat table for experimentation and agent testing. This flattened schema enables faster querying and simplifies context retrieval during AI prompt execution. In the future, especially when integrating the system into production, we aim to use big data and data lake technologies to unify all sources of truth into a central, query-optimized architecture.

At the final stage of data preparation, we created a centralized dataset by merging all relevant sources. This dataset became the foundation for the AI agent, allowing it to generate reports based on simple prompts and structured logic.

The most important columns used during analysis and integration were:

- **Order-related columns**: These included essential fields such as `order_id`, `product`, `quantity`, `loading_date`, `order_type`, `priority`, and `status`. They helped describe the what, when, and how of each transport operation.
- **Vendor and client information**: Columns like `vendor_name`, `client_name`, and `contact_name` were useful for building relationship-based reports, analyzing vendor activity, and generating summaries per partner.

- **Storage and location data** Fields such as `departure_address`, `departure_caption`, `destination_address`, `destination_caption`, `dock_status`, and `dock_location` provided geographic and logistical context to each order.
- **Date fields and additional attributes**: We also relied on fields like `creation_date` for vendors, docks, and orders, and special tags or flags that described certain operational characteristics such as whether the product was a "big bag," or whether the operation was related to the El Kseur site.

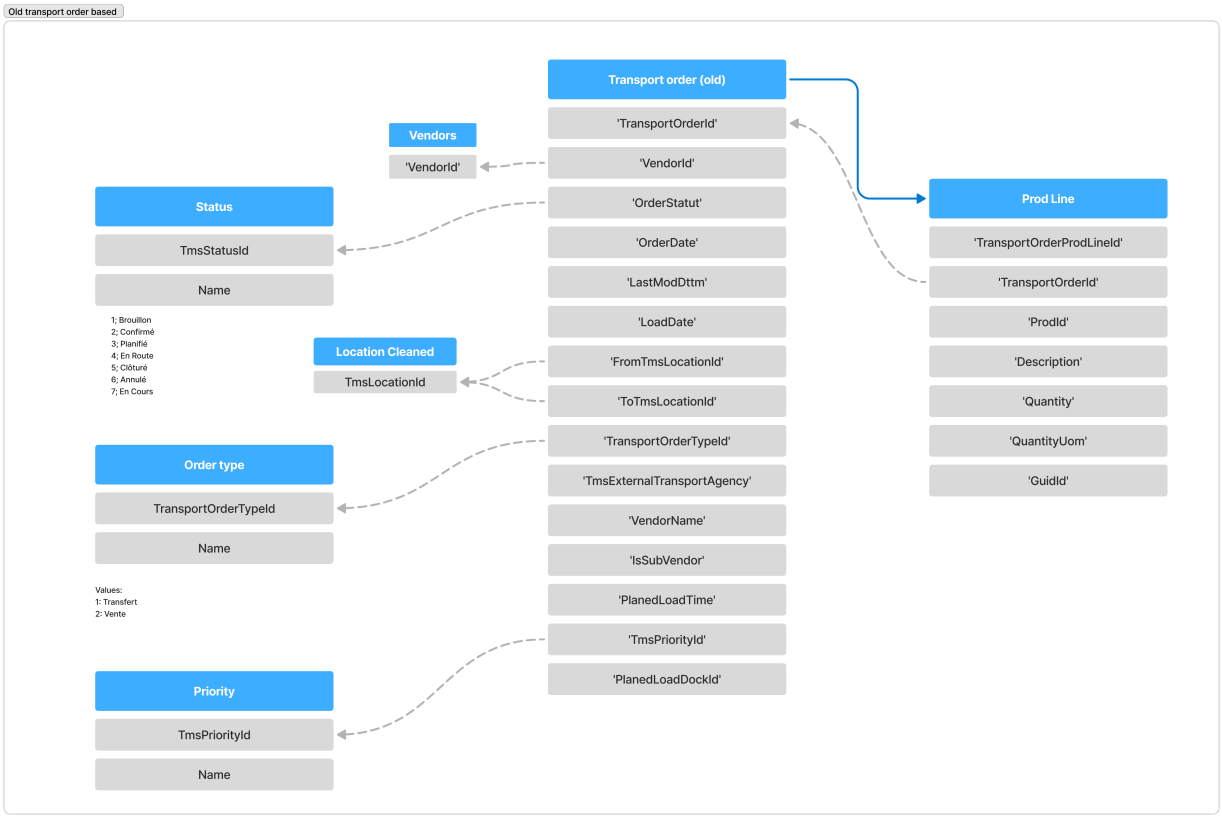


Figure 3.4: Datasets used during EDA and their relationships

### 3.3.2 Results & Impact on the project

As a direct outcome of the EDA phase, we successfully merged all relevant datasets into a single file `merged.csv` containing approximately **66,000 rows**. This unified dataset became the *backbone* of our reporting system and served as the central source for **AI-powered analysis**. In parallel, we built a complete data documentation file from scratch, describing each column's *purpose*, *type*, and *typical values* ensuring that both human users and AI agents can interpret and work with the data effectively.

This phase proved to be one of the **most critical stages** of the entire project. It not only enabled us to *clean and structure* the data, but also fundamentally shaped the base dataset used by the AI reporting agent. It laid the technical foundation for future developments, including the implementation of an **automated data pipeline**, which we plan to build once a stable and reliable live data source is secured.

The EDA process helped us understand the data deeply. By thoroughly exploring each dataset, we were able to produce detailed documentation for two main purposes: first, to serve as the "*brain*" of the AI agent helping it interpret each column's *meaning*, *data type*, and *analytical role* for accurate responses; second, to provide Cevital with reusable **metadata** that can be integrated into their internal information systems. You'll find a figure below illustrating the structure we followed to document the dataset.

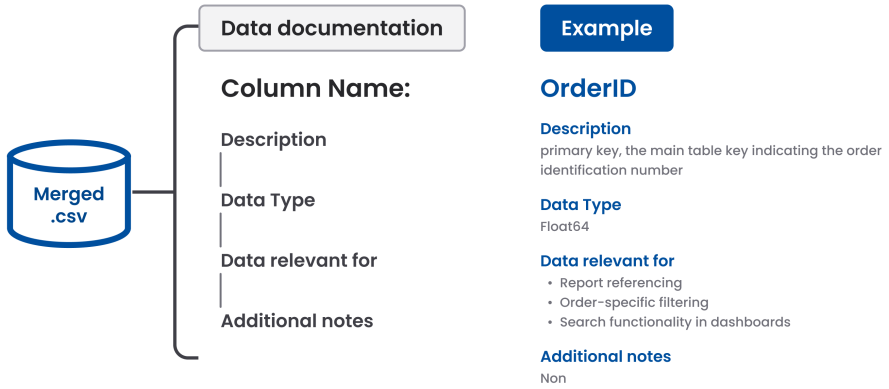


Figure 3.5: Dataset Documentation Structure

Additionally, our deep exploration of the datasets allowed us to construct a **reliable evaluation dataset**. By analyzing real-world patterns and business logic, we generated a collection of realistic user questions paired with verified answers. This benchmark dataset was instrumental in *evaluating and fine-tuning* the AI agent's performance.

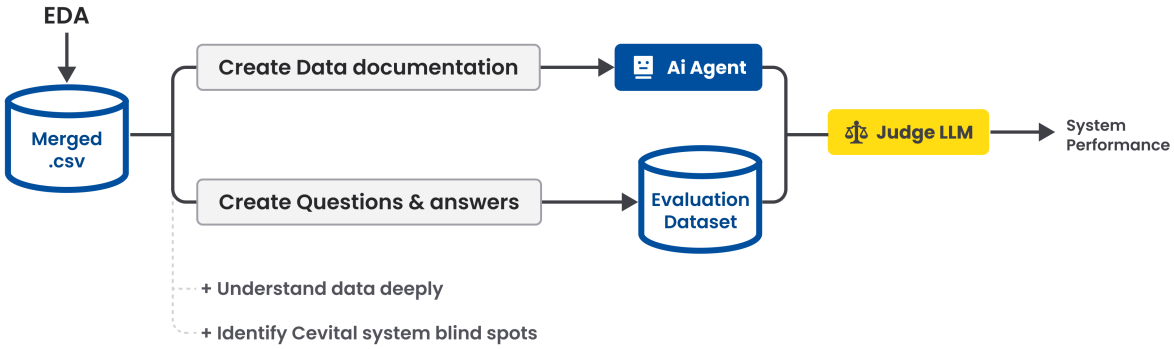


Figure 3.6: Impact of EDA on the project

The exploration also exposed critical **blind spots** in Cevital's current data ecosystem. We identified issues such as the *absence of real-time data*, *poor traceability*, and *inconsistent records* across systems. These findings highlighted systemic limitations that need to be addressed before scaling AI-powered solutions.

The insights gained from EDA underlined the broader need for structural improvements. Cevital would benefit from modern data infrastructure such as **data lakes** and **real-time tracking systems**, as well as specialized roles focused on managing and *streamlining data flows* across departments.

Finally, without this EDA process, we wouldn't have been able to reach a solid understanding of the system, build our AI solution, or even measure its performance accurately. It served as both a *technical* and *strategic milestone* for the project.

# CHAPTER 4

# SYSTEM DESIGN AND IMPLEMENTATION

Traditional Business Intelligence (BI) solutions are increasingly inadequate for the challenges of modern supply chain management. Despite major investments in data infrastructure, the promise of data-driven decision-making remains out of reach for many operational teams. The root issue lies in conventional BI tools' inability to deliver the speed, flexibility, and accessibility required in today's volatile environment. Most platforms demand technical skills like SQL, excluding frontline staff from direct access. This creates inefficient workflows, where business users depend on overloaded data teams causing delays and misinterpretations, especially harmful during time-sensitive disruptions.

Traditional reporting is also rigid. Predefined dashboards cannot adjust to sudden events like port strikes or supplier delays. In these moments, decision-makers need tailored insights quickly something static BI tools can't provide. Legacy BI systems also struggle to keep pace with evolving data needs, requiring time-consuming reconfigurations that widen the gap between available data and actionable insights. In our case, the challenge was deeper: the company's data lacked structure, consistency, and label quality making it unsuitable for training robust machine learning models. Digital maturity remains focused on descriptive analytics and static reporting, with AI adoption still in an exploratory stage centered on understanding, not automation.

To address this, we developed a multi-agent AI system focused on analytical support rather than prediction. This allowed us to deliver value without complex data restructuring or machine learning pipelines. Each agent specializes in a task reporting, supplier monitoring, anomaly detection and collaborates to produce adaptive, evolving insights. Our system reimagines BI by enabling natural language interaction with supply chain data. This removes technical barriers, allows flexible queries, and adapts easily to business context changes.

This chapter outlines our approach to overcoming BI limitations. We describe the architectural principles that enable accessible, adaptive analytics and show how they translate into real-world implementation. We then evaluate performance across operational scenarios and conclude with an honest review of current limitations and future directions. Together, these elements point to a new generation of intelligent decision support systems that deliver on the promise of accessible, actionable insights for all supply chain stakeholders.

## 4.1 Global System Architecture

In this section, we introduce the Multi-AI Agent System for Supply Chain Data Analysis, an architecture specifically designed to enhance decision-making through intelligent and user-friendly data interaction. This AI-powered system enables users particularly supply chain professionals to derive insights from complex datasets using natural language, eliminating the need for traditional BI tools or technical expertise.

The architecture integrates heterogeneous data sources through a robust data pipeline that handles ingestion, cleansing, and transformation. All processed data is stored in a centralized Data Lake, which ensures consistency, scalability, and traceability. At the heart of the system lies the Multi-AI Agent, a collaborative ensemble of agents capable of interpreting user queries, reasoning over data, and generating insightful responses. This core is supported by a natural language interface that powers intuitive interactions between users and the backend analytics engine.

A key strength of the system is its ability to abstract technical complexity away from the user. By allowing natural language queries, the system democratizes access to analytics, empowering non-technical stakeholders to engage with operational data meaningfully and efficiently.

Designed with modularity, scalability, and interpretability in mind, the system allows each component from data ingestion to reasoning to evolve independently. The architecture also ensures explainable outputs and transparent processing steps, which are crucial for fostering trust in AI-driven decision support.

As illustrated in the high-level system architecture figure 4.1, the core components include: Data Sources, the Data Pipeline, the Data Lake, the Multi-AI Agent, and the User Interface. This diagram outlines the full flow from data acquisition to insight delivery while highlighting the orchestration of AI agents and backend services responsible for natural language understanding, query generation, and intelligent data retrieval.

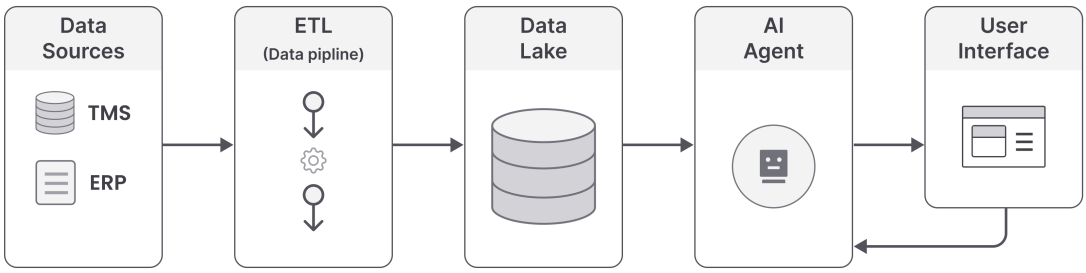


Figure 4.1: High-level system architecture

This architecture ensures that the data is always up to date and structured in a way that facilitates fast and accurate responses from the AI agent.

### 4.1.1 Data Pipeline and Lake Architecture

Our Data Lake serves as the central repository for all structured and unstructured supply chain data ingested by our pipeline, primarily sourced from the Transportation Management System (TMS) and the ERP system. It is intentionally designed as a **separate layer** from the company's production databases, and architectural choice that improves scalability, safety, and long-term flexibility.

This separation ensures **separation of concerns**, allowing us to run resource-intensive AI and analytical tasks without interfering with production systems. It also improves **fault isolation**, offering a safe space to experiment and evaluate without risking live operations. Furthermore, this approach introduces **cleaner system boundaries**, enabling development teams to work independently of core infrastructure.

More than a storage layer, the Data Lake supports system-wide scalability by centralizing both raw and processed data. It enables multiple downstream applications, such as:

- **AI Agent Logs and Artifacts:** Including generated plots, execution traces, and evaluation resultsused for continuous testing and monitoring.
- **Analytics and Dashboards:** Business intelligence tools use the lake to compute metrics, trends, and operational KPIs.
- **Machine Learning Models as Tools:** Historical, clean data from the lake can be used to train or evaluate other modelse.g., for forecasting, optimization, or anomaly detection.
- **Testing and QA Automation:** Evaluation logs and benchmark datasets support automated pipelines for regression testing and CI workflows.

Organized using time-segmented files, metadata, and document embeddings, the Data Lake is designed to scale. New AI components or analytics applications can integrate seamlessly, making the lake a durable foundation for innovation across the organization.

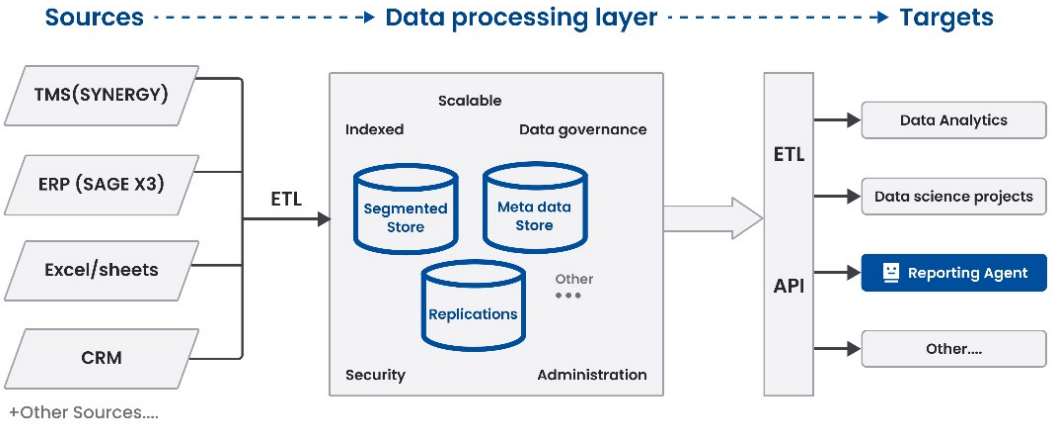


Figure 4.2: Overview of the Data Lake

The system ingests data from a variety of operational sources that reflect the complexity and breadth of the company's supply chain processes. These include structured and semi-structured datasets originating from different business units and information systems. The main data sources used throughout this thesis are summarized in Figure 3.1.

To ensure consistency, accessibility, and readiness for analysis, all incoming data passes through a robust data pipeline. This pipeline is responsible for collecting, cleaning, transforming, and organizing data from the various sources described earlier.

The pipeline operates in multiple stages. First, data is extracted from the ERP, TMS, and other local sources using scheduled connectors and ingestion scripts. Next, it undergoes preprocessing, which includes standardization of formats, handling of missing values, and enrichment with relevant metadata. This step also ensures that the data aligns with a unified schema suitable for downstream consumption.

The processed data is then stored in a centralized **data lake**, which serves as the main repository for both structured tabular data and semi-structured information. Alongside traditional tables, the data lake also hosts a **vector store** that contains semantic representations (embeddings) of relevant documents and supply chain knowledge. These embeddings allow the AI agent to perform more intelligent searches and reasoning based on natural language queries.

The Data Lake is built on top of **MinIO**, an open-source, high-performance, S3-compatible object storage system. MinIO provides scalable, cloud-native storage for our time-segmented files and ensures reliable data access across components.

The current architecture is organized around three core components, each serving a specific function in the system's data infrastructure:

- **Segmented Data Store**, which contains time-segmented raw data files (e.g., `.csv`) from the TMS and other operational systems. These files are organized by date and stored as objects in MinIO for easy retrieval and historical analysis.
- **Data Documentation Store**, which hosts internal documentation, metadata, data schemas, and lineage information. This store supports schema comprehension and enables better interaction with the underlying datasets through descriptive and structured context.
- **Logs**, which include AI agent execution traces, generated outputs, evaluation results, and performance metadata. These logs are critical for continuous monitoring, debugging, and improvement of the system.

This modular setup ensures that both operational data and AI-generated artifacts are accessible and reusable. While the system currently contains only these three components, its design allows for additional storage layers to be integrated seamlessly as the system scales or evolves to meet new requirements.

### 4.1.2 Data Segmentation Strategy

Segmentation defines how incoming data is split and stored over time. This structural decision directly impacts query performance, failure isolation, and the ease of auditing. In

our system, segmentation serves both organizational and operational roles by aligning with the natural rhythm of TMS data generation (typically in daily batches).

We examined four segmentation strategies random, daily, hourly, and weekly to determine the most appropriate fit for our use case.

Random segmentation stores data in arbitrary partitions, often based on storage size or ingestion events. While this method simplifies file management and reduces initial system complexity, it performs poorly for temporal queries, making analytics inefficient. Auditing is particularly difficult, as records may span across unrelated timeframes, and failure isolation is low, increasing the risk of data corruption spreading across files. Furthermore, it does not integrate well with systems that expect time-based organization.

Daily segmentation, on the other hand, organizes data into day-specific files. This approach aligns well with natural reporting cycles and operational patterns in TMS systems. It provides excellent query efficiency for time-based analysis such as daily KPIs or anomaly detection. It also ensures high failure isolation and clear audit trails by date. Although this strategy increases the number of files and requires moderate management effort, it fits strongly within the TMS ecosystem and simplifies automation through predictable naming conventions.

Hourly segmentation increases granularity by storing data at an hourly level, enabling fine-grained traceability and real-time analysis. It maintains high failure isolation and strong query performance for high-frequency systems. However, it introduces substantial overhead in file handling due to the large number of generated files. While suitable for systems requiring high temporal resolution, its added complexity makes it less appropriate for simpler workflows.

Finally, weekly segmentation reduces the number of files by aggregating data on a weekly basis. This makes file storage and management easier and reduces system fragmentation. However, its coarser granularity results in reduced auditability and less efficient queries for daily or sub-daily analysis. It offers a reasonable trade-off for systems with low-frequency reporting needs, but it lacks the precision required for finer-grained operations.

Table 4.1: Brief Comparison of Data Segmentation Strategies

| **Criteria** | **Random** | **Daily** | **Hourly** | **Weekly** |
|---|---|---|---|---|
| Query Efficiency | Poor | Excellent | Very good | Moderate |
| Failure Isolation | Low | High | High | Medium |
| File Management | Easy | Moderate | Hard | Easy |
| Auditing | Difficult | Easy | Fine-grained | Coarse |
| Fit with the ecosystem | Weak | Strong | Complex | Acceptable |

After careful consideration, we adopted the **daily segmentation** strategy for the following reasons:

1. **Query Efficiency:** Temporal queries such as daily reports, KPIs, or anomaly detection are simpler and faster to execute.

2. **Failure Isolation:** Data corruption or missing entries are limited to a single day's file, making debugging and recovery more manageable.

3. **Auditability and Traceability:** Analysis results can be traced back to specific inputs from a known date, improving trust and interpretability.

Files are stored using a date-based naming convention (e.g., `2025-06-05.csv`), supporting both human readability and programmatic access. Despite the increased number of files, daily segmentation offers the best balance between granularity, operational simplicity, and compatibility with our upstream data ingestion.

### 4.1.3 AI Agent Design

The AI agent is designed to bridge the gap between complex supply chain data and the decision-makers who rely on it. By enabling natural language interaction, it empowers non-technical stakeholders to access, understand, and analyze key operational insights without needing to write code or rely on traditional BI tools. This significantly improves data accessibility and enhances the decision-making process across departments.

Built using the `LangChain` framework in `Python`, the system integrates multiple specialized agents capable of parsing queries, invoking appropriate tools, and performing multi-step reasoning. It operates over both structured data such as CSV files and unstructured data including vectorized documents stored in the Data Lake. This architecture allows the agent to respond intelligently to a wide range of supply chain-related questions, delivering fast and context-aware insights.

#### 4.1.3.1 Single vs. Multi-Agent Systems

In AI systems, agents are entities capable of perceiving their environment and taking actions to achieve specific goals. A **single-agent system** typically involves one autonomous agent responsible for managing all tasks and interactions. While this approach is relatively simple to implement and maintain for basic use cases, it quickly reaches its limits when faced with complex workflows. Single agents often struggle with multi-step reasoning, tool orchestration, and scalability especially when interacting with heterogeneous data sources.

To overcome these limitations, **multi-agent systems (MAS)** offer a modular and scalable alternative. In a MAS architecture, distinct agents are assigned specialized roles such as query parsing, data retrieval, analysis, or natural language generation. These agents collaborate often through message passing or structured handoffs so that the output of one agent becomes the input for the next. This collaborative structure enables parallel task execution, improves fault tolerance, and allows for better maintenance and debugging.

The MAS design philosophy aligns naturally with the layered demands of supply chain data interaction, where tasks often proceed in stages: from validation, to analysis, to reporting. Furthermore, the multi-agent approach makes it easier to integrate diverse tools and support both structured (e.g., CSV) and unstructured (e.g., document) data, a crucial capability for our AI agent built using `LangChain` in `Python`.

Given the nature of supply chain tasks which often involve distinct stages such as validation, analysis, and reporting we opted for a **multi-agent system**. As shown in Table 4.2,

Table 4.2: Comparison Between Single-Agent and Multi-Agent Architectures

| Aspect | Single-Agent | Multi-Agent (MAS) |
|---|---|---|
| Architecture | One agent does everything | Tasks split across multiple agents |
| Scalability | Limited with growing complexity | Easily scales with system size |
| Tool Integration | Hard to manage many tools | Easy via modular components |
| Fault Tolerance | Single point of failure | Failures are isolated |
| Execution Style | Sequential | Parallel and collaborative |
| Supply Chain Fit | Poor for multi-step tasks | Ideal for staged workflows |

this architecture offers greater modularity, scalability, and fault tolerance compared to a single-agent setup. It also simplifies integration with various tools and data formats, making it a better fit for the dynamic and heterogeneous environment of our use case.

#### 4.1.3.2 Supervisor vs. Swarm Architectures

There are two major architectural patterns for organizing multi-agent systems: **Supervisor-Based (Hierarchical)** and **Swarm-Based (Decentralized)**. Each model provides different trade-offs in terms of coordination, control, scalability, and fault tolerance.

In a **supervisor-based architecture**, a central agent acts as the orchestrator, coordinating the work of several subordinate agents. This supervisor has a global view of the system and is responsible for decomposing tasks, delegating subtasks, and monitoring the execution flow. Such an architecture enforces strict execution order, making it suitable for structured workflows like supply chain analytics. It also facilitates debugging and auditing, as the supervisor tracks the status and output of each component. For example, our system enforces data validation before proceeding to analysis or summarization, which is naturally handled by a central agent.


Figure 4.3: Supervisor-Based Architecture: Centralized control and task delegation.

In contrast, **swarm-based architectures** rely on decentralized coordination, where all agents are autonomous and interact locally. This design is inspired by natural systems such as ant colonies or bird flocks. There is no central control each agent follows simple behavioral rules, and coordination emerges from these interactions. This approach is highly scalable and robust to individual failures, making it more suitable for dynamic, exploratory, or optimization tasks. However, it is less effective when tasks require strict ordering, traceability, or integration of multiple complex tools.

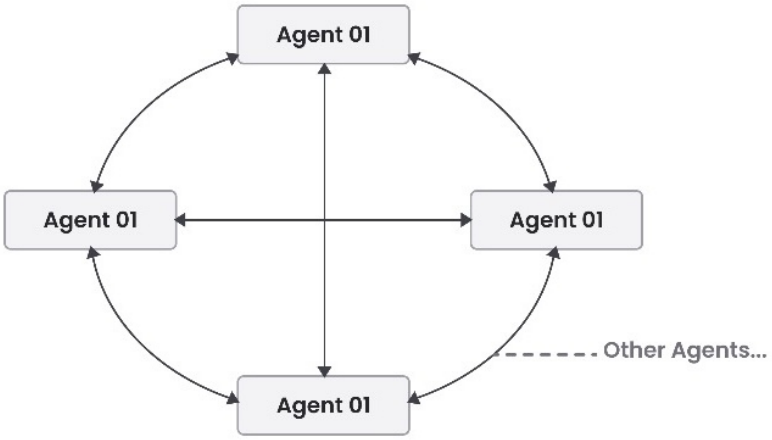


Figure 4.4: Swarm-Based Architecture: Decentralized coordination via local rules.

A summary of the differences between the two architectures is presented in Table 4.3.

Table 4.3: Comparison Between Supervisor-Based and Swarm-Based Architectures

| **Aspect** | **Supervisor-Based** | **Swarm-Based** |
|---|---|---|
| Control | Centralized (supervisor agent) | Decentralized (agents act autonomously) |
| Coordination | Top-down and explicit | Emergent via local interactions |
| Agent Complexity | Specialized with defined roles | Simple, uniform behavior |
| Scalability | Limited by supervisor's capacity | High; easy to add agents |
| Fault Tolerance | Lower; central point of failure | Higher; resilient to agent loss |
| Workflow Structure | Structured and traceable | Flexible and adaptive |
| Best Use Cases | Pipelines, task orchestration, analytics | Exploration, optimization, real-time adaptation |

Given the structured nature of supply chain processes and the need for traceability, we opted for a **supervisor-based architecture**. This design enables strict control over multi-step reasoning tasks (e.g., validation, document retrieval, and summarization) while maintaining transparency and auditability. It also simplifies tool orchestration and ensures that each query is processed in a logical and consistent manner.

### 4.1.4 Retrieval-Augmented Generation Pipeline

To enable dynamic, context-aware responses from agents, our system integrates a Retrieval-Augmented Generation (RAG) pipeline centered around the data documentation previously detailed in Section 3.3.2. This pipeline allows agents to query relevant information in real time from structured, embedded sources.

The pipeline works by first chunking the generated data documentationprovided in YAML formatinto semantically meaningful sections. Each chunk is then embedded using the `sentence-transformers/all-MiniLM-L6-v2` model from Hugging Face, a lightweight transformer known for its balance between speed and semantic accuracy.

These embeddings are stored in a vector database implemented with FAISS (Facebook AI Similarity Search), which provides fast and efficient similarity search over high-dimensional vectors. When a user or agent issues a query, the pipeline dynamically retrieves the most relevant chunks of documentation and integrates them into the reasoning or generation process.

FAISS is an open-source library developed by Meta AI that enables efficient similarity search and clustering of dense vectors. It is specifically designed to handle large-scale vector search problems by optimizing both memory usage and retrieval speed. FAISS supports various indexing structuressuch as flat indexes, inverted file systems (IVF), and Hierarchical Navigable Small World (HNSW) graphsthat can be selected based on application constraints like latency, precision, and scalability. In our system, FAISS enables the rapid retrieval of the most semantically relevant documentation chunks, ensuring that agents can reason with up-to-date, context-specific knowledge in real time.

This RAG architecture ensures that agents are constantly grounded in task-specific documentation, enhancing both the accuracy and relevance of their responses. It also supports modular expansion, allowing future integration of additional knowledge sources or documentation formats.

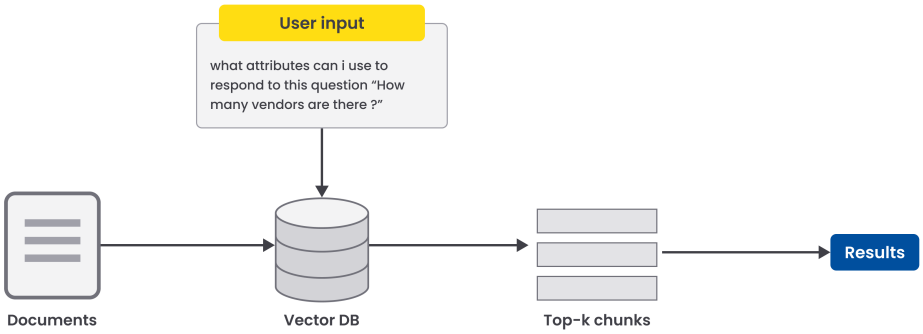


Figure 4.5: Overview of the RAG pipeline.

## 4.2 System Implementation

The architecture is composed of multiple specialized agents and tools that collaborate in a modular and loosely-coupled environment. Each component handles a specific task in the data analytics pipeline, orchestrated by a central Supervisor Agent.


Figure 4.6: Multi-Agent System Architecture

### 4.2.1 LLM-Powered Reasoning and Capabilities

Large Language Models (LLMs) serve as the central reasoning engine behind our AI agents. These models provide the linguistic and cognitive capabilities that enable agents to understand, decompose, and act upon user queries expressed in natural language. In our system, LLMs play a crucial role in supporting several advanced mechanisms that are essential for effective and interpretable data analysis.

First, LLMs support **Chain of Thought (CoT)** reasoning, which allows agents to break down complex analytical tasks into a sequence of logical steps. This not only improves the quality of the agent's responses but also enhances transparency by making the reasoning process traceable and auditable.

Second, LLMs are equipped with the ability to perform **tool calling**. This enables the agent to dynamically interact with external tools and systems such as querying databases, invoking APIs, or running analytical scripts when the task requires capabilities beyond the language model itself.

Third, the system leverages the LLM's ability to generate **structured outputs**, which ensures consistency and clarity in the interaction between agents, tools, and the supervisor module. Structured output includes fields like reasoning trace, selected tool, input/output data, and observations, all of which are necessary for traceability and coordination in a multi-agent setting.

To mitigate typical LLM limitations such as hallucinations, the system integrates validation agents that cross-check outputs with available metadata and enforce consistency constraints.

### 4.2.2 Prompting Strategy

The system relies on advanced prompting techniques to guide each agent's reasoning and execution. Prompting plays a critical role in shaping the behavior of large language models (LLMs), allowing the system to achieve high-quality, context-aware responses.

Among the techniques employed, we highlight the use of **zero-shot prompting**, where agents are given task instructions without any examples, relying solely on their pre-trained knowledge to interpret and respond accurately. This is particularly useful for general queries or when domain knowledge is embedded within the model.

In addition, the system adopts a modular approach by using **custom prompt templates** tailored for each agent. These templates encapsulate task-specific instructions, expected response formats, and context elements, ensuring consistency, accuracy, and structured output across different analytical scenarios.

### 4.2.3 Reasoning Engine: ReAct Paradigm

All agents in our system follow the **ReAct (Reason + Act)** paradigm, a powerful framework that combines large language model (LLM) reasoning with tool use. Rather than separating thinking and execution, ReAct agents interleave reasoning steps with actionable operations, enabling them to tackle complex, dynamic tasks in a structured and interpretable manner.

```
Thought: The analysis continues to show zero blockages when grouped by dock type, despite the value_counts() showing 239 True values.
This inconsistency suggests there might be missing or misaligned data in the 'MainGroup_Dock' column for the blocked dock records.

Given this data quality issue, I'll proceed to the next part of the analysis while noting that the dock blockage analysis cannot be
completed reliably with the current data.

2. Vendor Performance Check:
Thought: For vendor performance, I need to analyze order completion times and compare vendors. From earlier RAG results, I have
'VendorName' and order date columns.

Action: python_repl_ast
Action Input:
```python
# Check available columns related to vendors and timing
print([col for col in df.columns if 'vendor' in col.lower() or 'date' in col.lower() or 'time' in col.lower()])
```
Observation : ['OrderDate', 'LoadDate', 'PlanedLoad_Date', 'VendorGroupID', 'VendorName', 'Vendor_CreationDate', 'Dock_CreationDate']
```

Figure 4.7: A ReAct agent running: interleaving reasoning steps, actions, and observations

ReAct agents are designed with several core capabilities that make them particularly effective for complex, tool-based reasoning tasks:

- **Unified Reasoning and Acting:** ReAct agents use natural language reasoning to decide what action to perform next. Instead of simply retrieving answers, the agent interprets user input, formulates intermediate thoughts, and selects the best next step using external tools or APIs.
- **Dynamic Tool Utilization:** Agents are capable of choosing among multiple tools (e.g., calculators, databases, query engines) based on the task at hand.
- **Iterative Problem Solving:** ReAct agents operate in repeated cycles of `Thought → Action → Observation`, which enables the agent to refine its approach dynamically.

To ensure consistency, interpretability, and safe execution, all tools and agents in the system produce **structured outputs**. These outputs include clearly defined fields such as the reasoning trace, selected action, input/output of the tool, and any observations generated.

### 4.2.4 System Flow and Interaction

The multi-agent system is designed to decompose and solve complex supply chain analytics tasks by coordinating a set of specialized tools and agents. The architecture follows a structured and secure pipeline, where each component performs a clearly defined subtask. The overall interaction follows a four-stage flow:

1. **Query Reception and Planning:** The user submits a natural language query. The Supervisor Agent interprets the query and generates a high-level plan consisting of discrete analytical steps.
2. **Plan Validation:** The generated plan is passed to the Validation Agent, which checks for syntactic and semantic correctness, ensuring that only executable and relevant tasks are forwarded to downstream agents.
3. **Tool and Agent Invocation:** Once validated, the Supervisor dynamically delegates sub-tasks to appropriate agents such as the Ingestion Tool, Analytics Agent, or any specialized components.

4. **Result Aggregation and Response:** Each agent returns a structured JSON output containing results and metadata. The Supervisor collects and synthesizes these outputs into a coherent final response for the user.

This structured flow ensures that communication between components remains consistent, traceable, and machine-readable. All tools and agents communicate via structured JSON payloads, which define the input/output schema and enable modularity and extensibility. This design simplifies debugging, logging, and the future integration of new capabilities.

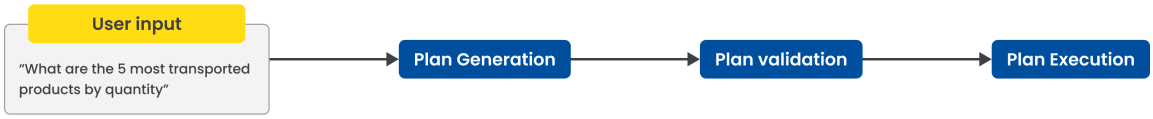


Figure 4.8: System Flow Diagram

### 4.2.5 System Tools

The architecture integrates several core tools that provide essential functionality for data access, secure computation, visual interpretation, and contextual disambiguation. These tools serve as the operational backbone of the system, used by agentsespecially the Supervisor and Analytics Agentto perform advanced and interpretable data tasks.

- **PythonAstREPLTool:** Executes Python code securely using an AST-based interpreter in a sandboxed environment. It supports essential libraries like `pandas` and `numpy` for safe, real-time data manipulation and analysis. By avoiding unsafe functions like `eval`, the tool ensures system security while enabling agents to compute statistics, transform tabular data, and perform numerical operations during reasoning.
- **Ingestion Tool:** Loads time-segmented supply chain data from the Data Lake (MinIO-backed), primarily organized as daily CSV files. It receives a `start_date` and `end_date`not from the user directly, but from the Supervisor Agent. It then aggregates the relevant files into a unified DataFrame stored in memory for fast access and processing.

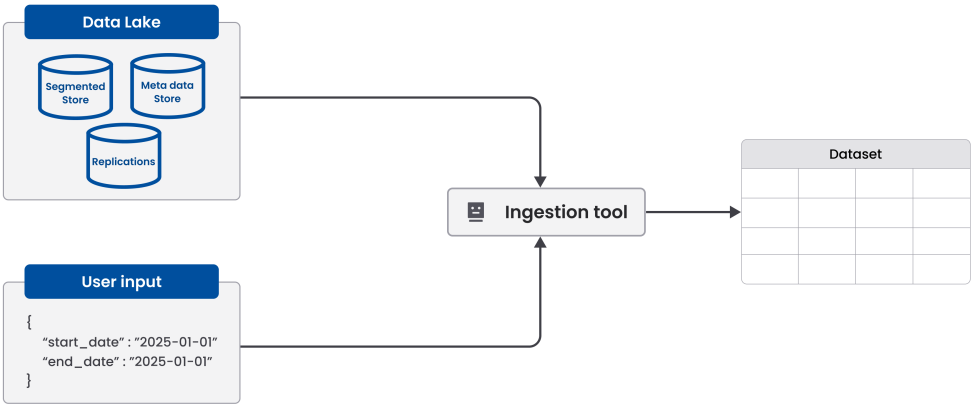


Figure 4.9: Ingestion Tool: Daily CSV aggregation based on start and end dates

- **Documentation Retrieval Tool:** This tool enables dynamic access to technical knowledge by leveraging a vector-based retrieval system. It operates on the YAML-formatted data documentation (see Section 3.3.2), which is chunked and embedded using the `sentence-transformers/all-MiniLM-L6-v2` model from Hugging Face. The resulting embeddings are stored in a FAISS vector store. When agents need contextual information, the tool queries this store to retrieve only the most relevant chunks, rather than loading all documentation upfront. This approach optimizes latency, reduces token consumption, and ensures the agent's responses remain grounded in task-specific context. The full pipeline is described in Section 4.1.4.

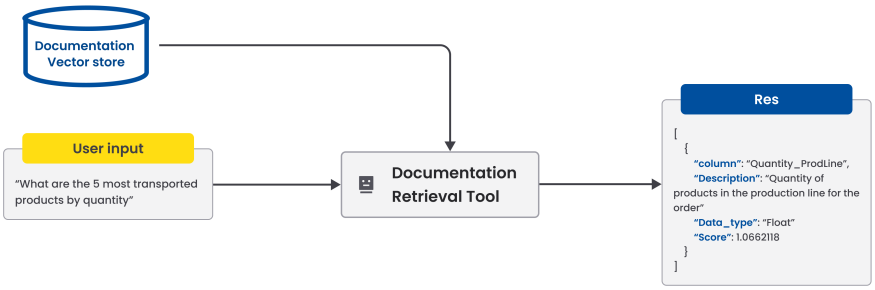


Figure 4.10: Retrieving contextual information from embedded documentation

- **Insight Renderer Tool:** Generates plots and charts using `matplotlib` from structured JSON inputs. It supports various chart types, including line, bar, scatter, and pie charts. The tool applies clean and accessible styling (using Seaborn's colorblind palette), adds meaningful annotations, and returns a base64-encoded PNG image. It is used by the Supervisor Agent to turn tabular outputs into visual insights for the user.
- **Documentation Loader Tool:** This tool provides structured access to dataset documentation stored in YAML format. Each entry in the documentation describes an attribute of the supply chain dataset, including fields such as `description`, `data_type`, and optional `additional_notes`. For example, the attribute `Caption_DockLocation` may be described as:

  `description:` "Label or name of the dock's physical location."
  `data_type:` "string"
  `additional_notes:` "value example: Unité raffinerie d'huile pour conditionnement"

  At runtime, this tool parses and loads the YAML documentation into memory, enabling agents to understand the semantic meaning of dataset fields. This improves the quality and accuracy of the agent's reasoning, especially when generating explanations, forming queries, or validating user inputs. The Documentation Loader Tool plays a key role in grounding the system's behavior in real-world data semantics.

### 4.2.6 System Agents

Agents are responsible for coordinating logic, validating plans, and executing specific operations using the tools described above. Each agent encapsulates a distinct function within the architecture, contributing to a modular, maintainable, and intelligent system.

- **Supervisor Agent:** The central orchestrator of the system. It receives the user query, generates a high-level plan, assigns tasks to other agents, and synthesizes the results. It also uses the **Insight Renderer Tool** to generate final visualizations that help users interpret the results of analytical queries. The Supervisor handles error reporting and can retry or reroute subtasks if failures occur.
- **Validation Agent:** Responsible for ensuring that the analytical plan is syntactically and semantically valid. To do this, it uses the **Documentation Retrieval Tool** to validate tool usage against official documentation. This helps prevent hallucinated or logically inconsistent steps and ensures that only well-structured plans are executed downstream.

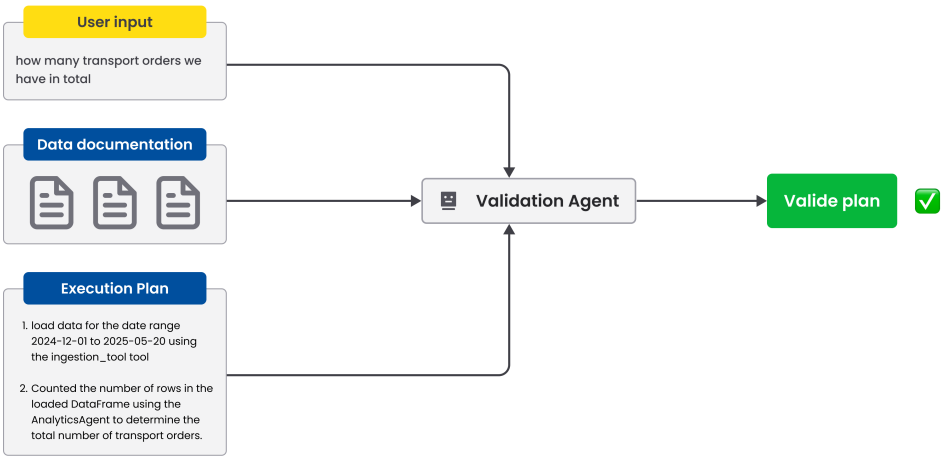


Figure 4.11: Validation Agent: Flow of plan verification using metadata and available tools

- **Analytics Agent:** A core component of the system that interprets and executes validated analytical instructions. It uses:
  - The **PythonAstREPLTool** to securely execute data transformations and statistical computations.
  - The **Documentation Retrieval Tool** to resolve ambiguities or complete missing logic by querying relevant domain-specific examples or code snippets.

  By combining these tools, the Analytics Agent can handle complex instructions like "Show a monthly comparison of average delivery times by supplier" and return accurate tabular outputs. This makes it a powerful and reliable component for delivering interpretable results.

## 4.3 User Interface

The interface serves as the primary point of interaction between users and the Analytics Agent. Built with **Streamlit**, it allows stakeholders to simply **type their queries or requests in natural language**, without the need to upload any raw data files like CSVs.

The system internally manages data ingestion and preparation, so users can focus entirely on formulating their questions and receiving actionable insights. This approach greatly lowers the technical barrier, making the analytics capabilities accessible to non-expert users.

To facilitate faster and more efficient usage, the interface provides a collection of **predefined reports and prompts** that cover the most frequent analytical needs such as supplier evaluation, inventory monitoring, or delivery performance. Users can select these templates to quickly generate relevant insights without needing to type detailed queries.

Additionally, the interface displays the **progress of the agent's processing** in real time, helping users track the status of their requests. It also incorporates a simple feedback system where users can *like* or *dislike* the responses, enabling continuous improvement based on user satisfaction.

The Analytics Agent can be triggered **manually through the interface** or automatically via scheduled events (daily, weekly), supporting both on-demand and proactive decision-making workflows.

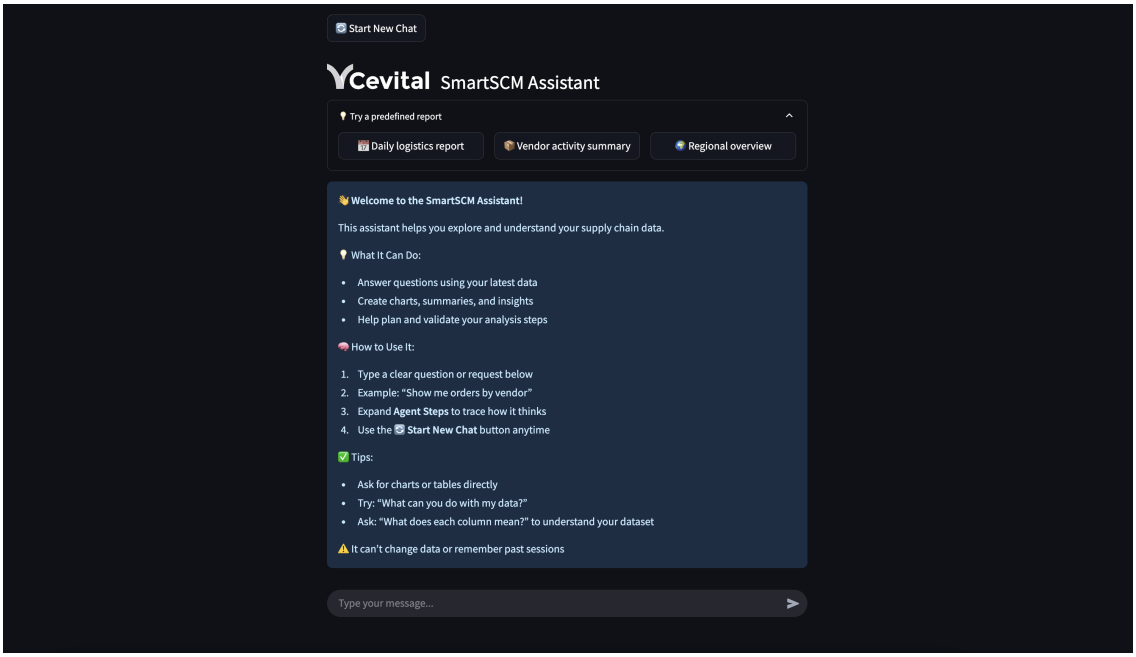


Figure 4.12: User interface built with Streamlit for interacting with the Analytics Agent

## 4.4 Extensibility and Scalability of the System

The architecture is inherently designed for extensibility and long-term scalability. Thanks to its modular and loosely coupled structure, it allows for the seamless addition of new agents and tools without disrupting existing components. This flexibility ensures that the system can evolve alongside organizational needs and technological advancements.

One of the key advantages of this architecture is its ability to grow both horizontally and vertically:

- **Horizontally**, by integrating additional agents to handle new tasks or business workflows.
- **Vertically**, by embedding advanced AI modelssuch as machine learning or deep learning pipelinesthat extend analytical and decision-making capabilities.

For example, a specialized **Forecasting Agent** can be integrated to address predictive tasks such as: "Forecast the demand for product Y over the next quarter." This agent could rely on statistical models (e.g., ARIMA, Prophet), machine learning regressors, or deep learning models such as LSTM or Temporal Fusion Transformers to analyze historical sales data, seasonality, and other relevant features.

In addition to forecasting, other AI-powered agents can be embedded to support key Supply Chain Management (SCM) functions, such as:

- **Inventory Optimization Agents**, which use reinforcement learning or genetic algorithms to minimize holding costs.
- **Pricing Strategy Agents**, which employ demand elasticity models and competitive analysis to recommend dynamic pricing.
- **Routing and Scheduling Agents**, which optimize logistics using constraint solvers or graph neural networks.
- **Anomaly Detection Agents**, which flag unusual patterns in procurement, production, or shipping using unsupervised models or autoencoders.

These AI-enhanced agents can be registered as tools within the system and orchestrated by the Supervisor Agent, which detects intent and delegates tasks accordinglyensuring that the system becomes increasingly intelligent and autonomous.

Moreover, the architecture is not limited to the internal multi-agent system. It enables the development of other **métier-specific applications** that can consume:

- Raw and enriched SCM data, or
- Insights and outputs generated by the agents (forecasts, anomalies, suggestions, etc.)

This opens up possibilities for downstream applications in business intelligence dashboards, ERP modules, TMS integrations, or decision support systemsenabling broader value creation across the enterprise.

In summary, this architectural paradigm provides a robust foundation for continuous expansion, making it well-suited for organizations aiming to embed AI incrementally and strategically into their supply chain operations.

## 4.5 Explainability of the System

Explainability has been a persistent challenge in the field of Artificial Intelligence. Traditional AI systems often behave like black boxes, producing results without offering insight into how or why specific decisions were made. This lack of transparency has limited the trust, usability, and accountability of AI solutions.

Our multi-agent system addresses this issue by making the internal workings of each agent visible and understandable. Rather than hiding the logic behind decisions, the system is designed to show exactly what steps have been taken and which step is currently being executed, in real time.

A key component of this approach is the *Plan Validation Agent*. This agent ensures that plans generated by the supervisor are coherent and aligned with system goals. While the validation logic itself is internal, the execution flowthe concrete steps the agents performis made fully transparent through the user interface. This UI component enables both developers and users to monitor the agent's progress and understand its behavior at each stage of operation.

By combining structured planning with live step visualization, our system transforms AI from a black box into a transparent, traceable, and trustworthy collaborator.

# CHAPTER 5

# BENCHMARKING AND RESULTS

The objective of this benchmarking study is to simulate the behavior of our AI agent in realistic conditions by evaluating its responses to actual user-like questions. The core aim is to assess the performance, cost, and accuracy of the agent when powered by different large language models (LLMs). This benchmarking serves to understand how well the AI agent can handle practical supply chain management queries and to determine its feasibility for integration in real-world systems.

## 5.1 Benchmarking Methodology

### 5.1.1 Evaluation Metrics

To rigorously assess the performance of the AI agent, we identified four primary evaluation metrics that reflect both technical efficiency and practical applicability:

- **Accuracy**: Evaluates how correctly the agent answers a given question compared to the expected output. This metric is critical to assessing whether the AI system produces reliable and relevant answers in a supply chain context. Accuracy is computed using a hybrid method combining semantic similarity and human-like judgment, described in the following section.
- **Token Consumption**: Captures the total number of tokens (input and output) used per query. This metric directly affects the operational cost, particularly in API-based deployments, and also serves as a proxy for the verbosity and efficiency of the model's responses.
- **Duration**: Measures the latency of the system, i.e., the time taken to generate a response for a single query. This is important for assessing the agent's responsiveness.
- **Error Rate**: Represents the proportion of queries that lead to failed or unusable responses. This includes system crashes and execution timeouts. High error rates can indicate poor robustness or weak error handling in the agent's reasoning chain.

These metrics were selected to provide a balanced view of the agent's performance in terms of *correctness* (accuracy), *cost-effectiveness* (token usage), *efficiency* (latency), and *robustness* (error rate). Together, they form the foundation for comparing multiple large language models under realistic, user-driven supply chain queries.

Quantitative metrics such as **token consumption** and **duration** were collected automatically using **LangSmith**, a tracing and observability framework tailored for LLM applications. LangSmith enables fine-grained tracking of inputs, outputs, token usage, and execution time across complex agent pipelines. This tooling was essential for ensuring consistency, reproducibility, and traceability throughout the benchmarking process.

### 5.1.2 Tracing and Monitoring with LangSmith

To ensure systematic, reproducible, and insightful benchmarking, we instrumented the AI agent using **LangSmith**, a tracing and observability platform developed by the creators of LangChain. LangSmith acts as a centralized debugging and evaluation layer, tailored for applications built on large language models (LLMs).

Figure 5.1 offers a visual overview of a complete agent execution trace. This includes the original prompt, intermediate reasoning steps, tool calls, and final output.

LangSmith enables detailed inspection of agent behavior across multi-step workflows, such as tool usage, chain execution, and internal reasoning paths. This makes it particularly suitable for complex LLM pipelines.

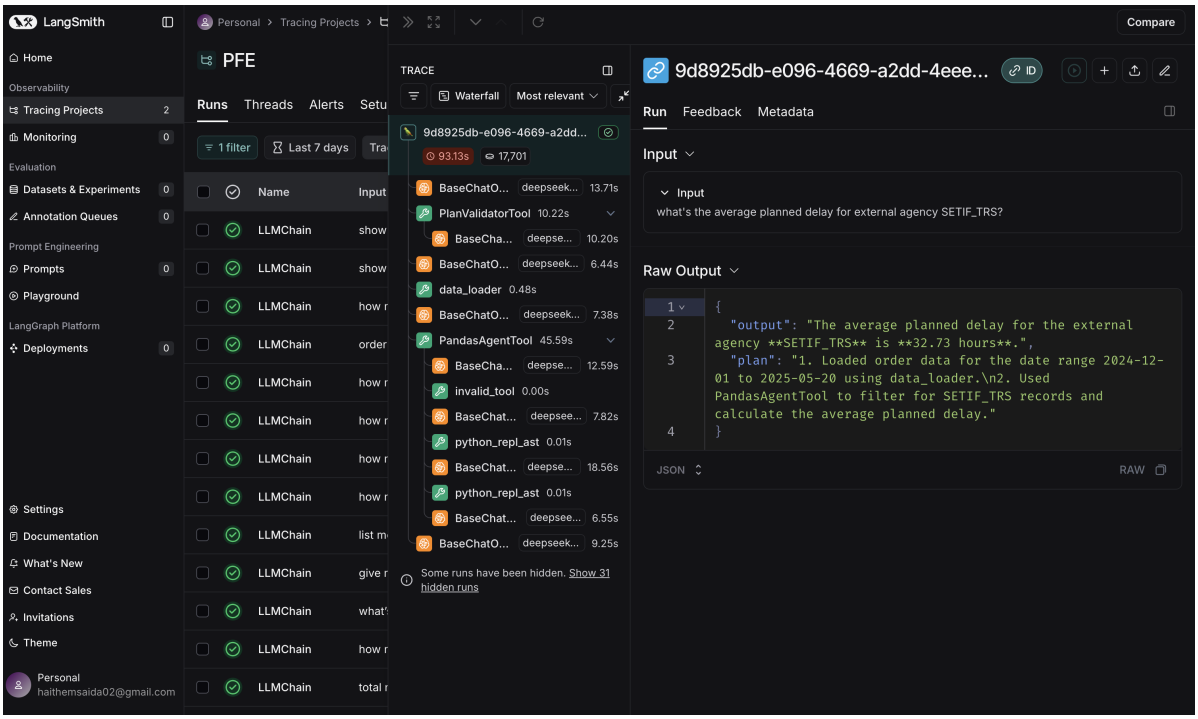


Figure 5.1: Agent trace view in LangSmith.

For each query in our benchmarking dataset, LangSmith logs:

- **Inputs and outputs**: The full prompt provided to the LLM and the response it generates.

- **Token consumption**: Total tokens used (input + output), directly linked to inference cost.
- **Duration**: End-to-end latency of the request, including both API call and tool execution time.
- **Intermediate steps**: In multi-step agents, LangSmith logs each sub-operation, including tool invocations and decisions.
- **Error logging**: Failures such as timeouts, execution crashes, or malformed responses are captured with metadata for diagnosis.

LangSmith integrates seamlessly with LangChain and can be used via a developer SDK or visual dashboard. For our benchmarking, we logged all executions programmatically and exported traces for both live debugging and statistical analysis.

This observability layer provided the following advantages:

- **Performance consistency**: Standardized metric collection across all evaluated models.
- **Robust debugging**: Immediate identification of pipeline errors, tool misuses, or abnormal latency spikes.
- **Behavioral validation**: Enabled inspection of agent reasoning, decision flow, and tool selection logic.

### 5.1.3 Hybrid Evaluation Strategy

To ensure accurate assessment of natural language answers, we adopt a hybrid evaluation pipeline that combines the efficiency of semantic similarity scoring with the contextual reasoning abilities of large language models. This strategy leverages the strengths of both SBERT and LLM-based methods in a sequential decision process.

We begin by computing the SBERT similarity score between the generated and reference answers. If the score exceeds a predefined threshold ($\theta = 0.8$), the answer is accepted as correct. Otherwise, we defer to a large language model for a deeper contextual evaluation.

The complete hybrid evaluation logic is shown in the following algorithm:

**Algorithm 1** Hybrid Evaluation: SBERT + LLM-as-a-Judge

```
procedure EVALUATE(question, reference_answer, generated_answer)
    score ← COMPUTESBERTSCORE(generated_answer, reference_answer)
    if score > θ then                                          ▷ e.g., θ = 0.8
        return True
    else
        verdict ← LLM_AS_A_JUDGE(question, reference_answer, generated_answer)
        return verdict
    end if
end procedure
```

The first technique, **SBERT-based semantic similarity**, uses Sentence-BERT to encode both the generated and reference answers into dense embeddings. The cosine similarity between these embeddings provides a quantitative measure of semantic closeness. If the similarity exceeds a predefined threshold ($\theta = 0.8$ in our case), the answer is considered correct.

This method is lightweight and fast, which makes it suitable for high-throughput evaluations. However, it has notable limitations: it cannot account for logical inconsistencies, domain-specific paraphrasing, or subtle errors in reasoning. As a result, answers that are semantically different in structure but logically equivalent may be rejected, while incorrect but lexically similar answers might be accepted.

The second approach, referred to as **LLM-as-a-Judge**, uses a large language model to assess correctness more holistically. The model is provided with the question, reference answer, and generated answer, and is prompted to return a binary decision on correctness. It does so using five evaluation criteria:

- **Factual Accuracy** Does the answer contradict or misrepresent the reference?
- **Completeness** Are the key ideas or facts adequately covered?
- **Relevance** Does the answer stay focused on the question's intent?
- **Clarity** Is the response coherent and easy to understand?
- **Terminology** Are acceptable synonyms and paraphrases used correctly?

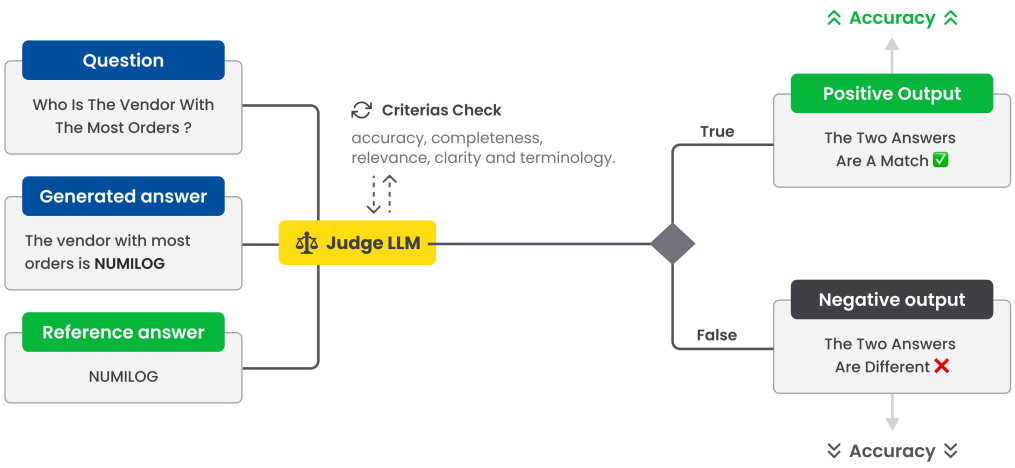


Figure 5.2: Illustration of the LLM-as-a-Judge Evaluation Process

This method excels at capturing nuance and evaluating meaning beyond surface similarity. It is particularly effective at identifying partial, misleading, or overly verbose answers that SBERT might fail to detect. However, this comes at the cost of higher computational demands and slower execution, especially when run on large-scale datasets.

This modular setup ensures reproducibility and scalability across various datasets and experimental configurations.

The following table summarizes the comparative strengths and weaknesses of the two techniques:

Table 5.1: LLM vs. SBERT: Summary of Evaluation Characteristics

| **Aspect** | **LLM Judge** | **SBERT** |
|---|---|---|
| Cost | High | Low |
| Speed | Slow | Fast |
| Output | Justified decision | Numeric score |
| Factuality Detection | Yes | No |
| Contradiction Detection | Detects | Misses |

This hybrid design provides both the efficiency of semantic similarity scoring and the depth of contextual human-like judgment, making it suitable for both large-scale and high-quality answer evaluation tasks.

### 5.1.4 Unit Testing of Tools

In our multi-agent system for supply chain data analysis, we unit tested core tools such as the **Data Ingestion** module and the **Insight Renderer**. The Data Ingestion tool loads and parses ERP and TMS data, while the Insight Renderer generates plots and visualizations to support analytical tasks.

Unit testing ensured these tools worked reliably and produced consistent outputs, reducing the risk of silent failures and incorrect interpretations. More importantly, it helped prevent agent crashes caused by malformed inputs or unexpected tool behavior.

This approach also made the system easier to maintain and extend, allowing us to introduce changes or new features with confidencecritical in a domain where data structures and business requirements frequently evolve.

### 5.1.5 Benchmark Dataset Construction

To evaluate the performance of our AI agent in realistic scenarios, we manually constructed a benchmarking dataset composed of natural language queries typically encountered in the domain of supply chain management. The dataset was designed to reflect both straightforward and ambiguous user behaviors, simulating real-world interaction patterns.

The final dataset comprises a total of **100 questions**, which fall into two main categories. The first group consists of **87 ordinary queries**, which are standard, well-formed questions related to inventory levels, order statuses, supplier data, delivery delays, and key performance indicators. The second group contains **13 out-of-scope (OOS) queries**, which are incomplete, vague, or context-switching questions designed to evaluate the agent's robustness, error handling, and contextual reasoning abilities.

Each question was manually authored based on realistic supply chain use cases and paired with a ground-truth implementation in `pandas`. The code is capable of executing against the operational dataset to return accurate, context-relevant answers. This code-annotated structure allows precise supervision and reproducibility of expected outcomes.

While the raw outputs from the code were usually in tabular or numerical format, they were reformulated into human-readable natural language answers using a local large language model, `gemma3:latest`, deployed via **Ollama**. This ensured stylistic and semantic alignment with how a real AI assistant would respond to the user.

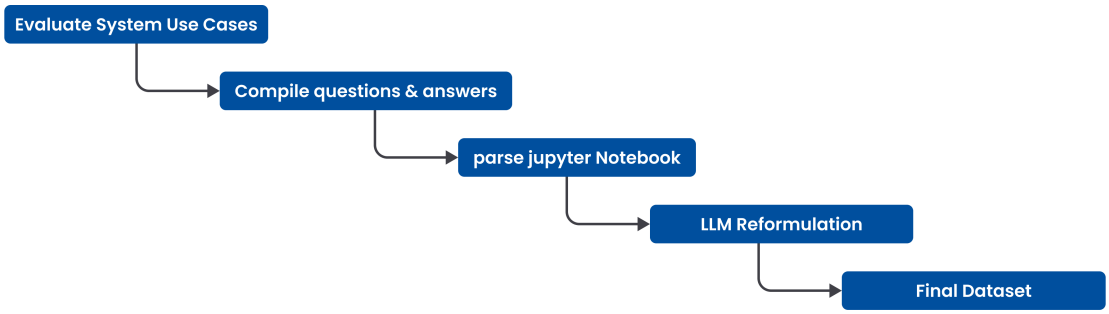


Figure 5.3: Overview of the Dataset Generation Pipeline

The final dataset was exported into a structured CSV file with the following fields:

- `questions`: the user query in natural language
- `answers`: the direct output generated from the code
- `code`: the pandas code used to compute the answer
- `type`: classification label either `ordinary` or `oos`
- `reformulated_answers`: the LLM-generated human-like version of the raw answer

The diversity and realism of this dataset enable rigorous evaluation of the agent's generalization, semantic flexibility, and error tolerance. Future extensions will incorporate multilingual queries, broader supply chain functions, and multi-turn dialogue.

### 5.1.6 Testing Pipeline

To evaluate the performance of our AI agent, we designed a lightweight, reproducible testing pipeline executed locally on a MacBook Air equipped with an **M1 chip**, 16GB of unified memory, and running macOS. All scripts were implemented in **Python 3.13.2**, ensuring compatibility with the latest machine learning and NLP libraries.

For inference, we leveraged the **Ollama** framework to run language models locally. Specifically, we used **Nous Hermes 2 - Mistral 7B - DPO**, selected for its effective reasoning ability and practical speed in local environments.

The entire testing pipeline operates on a common evaluation dataset that is generated from the notebook using a dedicated preprocessing script.

The evaluation process involved the following steps:

1. We first executed the `benchmark.py` script, which loads the generated evaluation dataset and runs the agent to produce answers. During this process, intermediate metrics were collected using the **LangSmith** tracing and logging platform.

2. Next, we ran the `run_tests.py` script, which applied a two-step hybrid evaluation pipeline: a semantic similarity check using **SBERT**, followed by LLM-based judgment for edge cases or low-confidence matches.

3. The script then aggregates all results and computes final accuracy, duration, and token usage.

The final output of the testing pipeline is summarized in the figure below.

```
Total tests: 100
Successes: 83
Failures: 17
Accuracy: 83.00%
Average duration (sec): 80.28078249494949
Average total tokens: 18667.24
Accuracy by question type:
  OOS: 92.31%
  Ordinary: 81.61%
```

Figure 5.4: Summary of Test Results: Total Runs, Duration, Token Usage, and Accuracy

## 5.2 Benchmarking Results and Analysis

### 5.2.1 Model Selection Rationale

For the initial phase of evaluation, we selected **DeepSeek V3** as the sole large language model powering the AI agent. This choice was guided by its strong performance in recent benchmarks, its alignment with cost-performance trade-offs, and its availability through API access. DeepSeek V3 is recognized for its capacity to handle complex reasoning tasks and to generalize well across diverse domains, making it a strong candidate for supply chain applications.

While future iterations of this study will incorporate multiple LLMs for comparative benchmarking, this phase aimed to establish a reliable performance baseline and to validate the overall effectiveness of the agent pipeline.

### 5.2.2 Results Overview

The system was evaluated on a curated dataset of 100 natural language queries, encompassing both standard supply chain queries and out-of-scope (OOS) prompts to assess robust-

ness. Evaluation was conducted using the hybrid scoring mechanism previously described, combining SBERT similarity and LLM-based judgment.

| Metric | Result |
|---|---|
| Accuracy | 83% overall (81.61% on ordinary, 92.31% on OOS queries) |
| Average Token Consumption | 18,667.24 tokens/query |
| Average Duration | 80 seconds/query |
| Error Rate | 0 out of 100 queries (0%) |

Table 5.2: Benchmark Results Summary for DeepSeek V3

The evaluation demonstrates that the agent consistently delivers accurate and contextually appropriate responses:

- **Accuracy:** The system achieved 83% overall accuracy, with strong performance on both ordinary queries (81.61%) and exceptional handling of out-of-scope inputs (92.31%), indicating robustness to non-standard scenarios.
- **Token Efficiency:** The average token usage per query was 18,667.24, which is acceptable for most production-level applications considering current LLM pricing models.
- **Latency:** With an average response time of 80 seconds per query, the system remains within tolerable latency thresholds for semi-interactive use cases.
- **Error Rate:** Notably, the system produced zero critical failures across all test cases, reflecting high stability and fault tolerance.

### 5.2.3 Qualitative Observations

In addition to quantitative metrics, several qualitative patterns emerged:

- The agent consistently succeeded in tasks involving tabular summarization, ratio computation, and categorical filtering.
- Minor factual hallucinations occurred when the agent misinterpreted ambiguous prompts or lacked access to clarifying context.
- For out-of-scope queries, the agent tended to generate polite but speculative answers, reflecting a lack of explicit refusal training.

Example excerpts are shown below to illustrate typical success and failure patterns:

- **Successful response:** *"The supplier with the highest average delivery delay is Supplier X with an average of 4.3 days."*
- **Failure case (hallucination):** *"There is no recorded delivery for Product Y in the last month,"* — incorrect due to misinterpretation of a missing column.

### 5.2.4 Limitations and Future Benchmarking Directions

Although DeepSeek V3 an **open-source** model demonstrated strong performance, several limitations remain in our current evaluation framework, both in scope and methodology.

First, benchmarking **AI agents** is inherently more complex than evaluating traditional single-turn LLM responses. Agents operate through multi-step reasoning, tool usage, and memory, which makes standard evaluation approaches difficult to apply. Their behavior depends not only on prompt understanding but also on decision-making processes over time.

Additionally, the current evaluation dataset, while carefully curated, is limited in size and coverage. With only 100 queries, it does not capture the full range of real-world use cases in supply chain management. Future work will expand the dataset and adopt a **multi-dataset strategy**, where each agent module is evaluated on a targeted benchmark. An example is the **Validation Agent Dataset**, designed to assess how well the system transforms user queries into executable plans.

Benchmarking agents is also **costly and operationally demanding**, especially when relying on LLMs to evaluate responses. Using an LLM as a judge requires high inference costs and slows down the testing process. To address this, we implemented a **hybrid evaluation pipeline** that uses SBERT for fast similarity scoring and only invokes an LLM when similarity falls below a threshold. This significantly reduces overhead while preserving evaluation quality.

Another key challenge is that effective agent benchmarking requires **continuous monitoring and integration**. Agents evolve frequently due to changes in models, tools, or context which can silently affect performance. To mitigate this, our system incorporates fine-grained **logging and explainability** features using LangSmith. These logs capture the full execution trace, including tool calls, reasoning paths, and final answers, enabling detailed audits and reproducibility.

A particularly notable case highlighted the limitations of rigid evaluation strategies. When asked:

> *"Give me the number of orders made during Ramadan."*

our reference label indicated that the system could not answer this query due to the lack of an explicit calendar or holiday-detection tool. However, the agent responded:

> *"There were **12,507 orders** made during Ramadan (March 1 to March 30, 2025)."*

Surprisingly, the agent internally inferred the approximate date range of Ramadan and applied the correct filter to generate a plausible result. Although the answer was factually consistent and contextually sound, it was marked incorrect by the evaluator. This example illustrates how static ground-truth labels can fail to capture emergent reasoning capabilities, and how agents can exceed predefined expectations.

Such cases reinforce the need for **more flexible and context-aware evaluation strategies**, capable of adapting to creative and valid reasoning paths not foreseen during dataset construction.

Finally, although this phase focused on a single LLM, there is a clear need to **benchmark additional models** to better understand trade-offs in accuracy, cost, and latency. Different models vary in reasoning ability, verbosity, and robustness, which can significantly affect agent behavior.

Future evaluations will help identify which models perform best for specific tasks such as planning, tool selection, or response generation. We also plan to **cherry-pick the most suitable model for each agent**, ensuring an optimal balance between performance and cost across the system. For example, lightweight models may be used for fast validation, while more capable ones handle complex analysis or dialogue.

# CONCLUSION

Throughout this project, we explored how artificial intelligence can be realistically integrated into the industrial world by focusing on Cevital's supply chain system. After studying different possibilities and evaluating multiple areas where AI could bring real value, we decided to center our thesis on one practical and impactful use case: automating the generation of business reports from historical transport data.

To approach this challenge, we chose to work with AI agents powered by large language models (LLMs), a promising and fast-evolving field. Specifically, we built a system using the pandas Agent, which allowed us to interact with structured data and generate relevant reports in real time. Our focus was on creating a solution that is both low-cost and locally hosted, making it practical and deployable in a real-world industrial setting like Cevital.

One of the most rewarding aspects of the project was applying our solution to a real use case. Unlike academic environments with artificial datasets and ideal conditions, the real world is messy. We had to deal with missing data, inconsistencies, a lack of documentation, and fragmented systems. But it is precisely in this kind of environment that innovation becomes meaningful. Building a system that works under real constraints taught us far more than any textbook scenario could have.

The current version of our reporting agent works well for generating simple and direct reports, especially those based on pre-defined queries. But this is only the beginning. Thanks to the modularity of AI agents and their ability to integrate multiple tools and protocols (like MCP), the system can evolve to handle more complex tasks. In the future, we envision adding features such as demand forecasting, ERP integration, email reporting, and even voice or chatbot interfaces making the solution not only more intelligent but also more accessible to all departments.

With the ongoing progress in the field of LLMs, we also see opportunities to improve response time, optimize resource consumption, and build a smoother, faster user experience. These improvements could eventually make this tool a core element of Cevital's internal workflow.

Finally, we want to emphasize that AI systems like the one we built are not here to replace humans. Instead, they are designed to support humans, automate repetitive and time-consuming tasks, and free up time for high-level decision-making and creative problem-solving. In that sense, our project shows how AI can be a true collaborator not a competitor within the future of industrial operations.

# BIBLIOGRAPHY


[1] Marijn Overvest, Sjoed Goedhart, and Ruud Emonds. *Supply Chain Statistics 70 Key Figures of 2025*. https://procurementtactics.com/supply-chain-statistics/. Accessed June 24, 2025. Written by Marijn Overvest, reviewed by Sjoed Goedhart, fact-checked by Ruud Emonds. 2025.

[2] Djamel Eddine Aggoune and Walid Oukachbi. "Contribution à l'amélioration de la performance de la chaine logistique de CEVITAL." Retrieved from ENP Repository. Master's thesis. École Nationale Polytechnique d'Alger, 2022. URL: https://repository.enp.edu.dz/jspui/bitstream/123456789/6392/1/AGGOUNE.Djamel%20Eddine_OUKACHBI.Walid.pdf.

[3] Vikas Misra, M.I. Khan, and U.K. Singh. "Supply Chain Management Systems: Architecture, Design and Vision." In: *Journal of Strategic Innovation and Sustainability* 6.4 (2010). Journal of Strategic Innovation and Sustainability vol. 6(4) 2010, pp. 102–108. URL: http://www.nabuspress.com/JSIS/MisraWeb.pdf.

[4] Meena Siwach and Suman Mann. "A Compendium of Various Applications of Machine Learning." In: *International Research Journal of Engineering and Technology (IRJET)* 9.7 (2022), pp. 1141–1144. URL: https://www.researchgate.net/publication/362150447_A_Compendium_of_Various_Applications_of_Machine_Learning.

[5] Aishwarya Shekhar et al. "Generative AI in Supply Chain Management." In: *International Journal on Recent and Innovation Trends in Computing and Communication* 11.9 (2023), pp. 4179–4185. ISSN: 2321-8169. URL: https://www.researchgate.net/publication/378140419_Generative_AI_in_Supply_Chain_Management.

[6] Giovanna Culot, Matteo Podrecca, and Guido Nassimbeni. "Artificial intelligence in supply chain management: A systematic literature review of empirical studies and research directions." In: *Computers in Industry* (2024). DOI: 10.1016/j.compind.2024.104132. URL: https://www.sciencedirect.com/science/article/pii/S0166361524000605.

[7] Haifeng Lin, Ji Lin, and Fang Wang. "An Innovative Machine Learning Model for Supply Chain Management." In: *Journal of Innovation & Knowledge* 7 (2022), p. 100276. DOI: 10.1016/j.jik.2021.100276. URL: https://www.sciencedirect.com/science/article/pii/S2444569X22001111.

[8] Javad Feizabadi. "Machine Learning Demand Forecasting and Supply Chain Performance." In: *International Journal of Logistics: Research and Applications* 25.2 (2022), pp. 119–142. DOI: 10.1080/13675567.2020.1803246. URL: https://www.tandfonline.com/doi/full/10.1080/13675567.2020.1803246.

[9] Yunbo Long et al. "Leveraging synthetic data to tackle machine learning challenges in supply chains: Challenges, methods, applications, and research opportunities." In: *International Journal of Production Research* (2024). DOI: 10.1080/00207543.2024.2447927. URL: https://www.tandfonline.com/doi/abs/10.1080/00207543.2024.2447927.

[10] GeeksforGeeks. *Machine Learning Model Evaluation*. Accessed: 2025-06-05. 2023.

[11] Jani Data Diaries. *Choosing the Best Model: A Friendly Guide to AIC and BIC*. Nov. 2024. URL: https://medium.com/@jshaik2452/choosing-the-best-model-a-friendly-guide-to-aic-and-bic-af220b33255f (visited on 06/05/2025).

[12] Venkat Sharma Gaddala. "Unleashing the Power of Generative AI and RAG Agents in Supply Chain Management: A Futuristic Perspective." In: *ICONIC RESEARCH AND ENGINEERING JOURNALS* 6.12 (2023). ISSN: 2456-8880.

[13] B.L. Aylak. "SustAI-SCM: Intelligent Supply Chain Process Automation with Agentic AI for Sustainability and Cost Efficiency." In: *Sustainability* 17.6 (2025), p. 2453. DOI: 10.3390/su17062453. URL: https://www.researchgate.net/publication/389736744_SustAI-SCM_Intelligent_Supply_Chain_Process_Automation_with_Agentic_AI_for_Sustainability_and_Cost_Efficiency.

[14] Evidently AI. *LLM as a judge: How to evaluate outputs of LLM applications*. Retrieved June 5, 2025. 2025. URL: https://www.evidentlyai.com/llm-guide/llm-as-a-judge.

[15] Ruman. *BERTScore Explained*. https://rumn.medium.com/bert-score-explained-8f384d37bb06. Retrieved June 5, 2025. 2024.

[16] UKPLab and Hugging Face. *Sentence Transformers*. Accessed: 2025-06-05. 2025. URL: https://sbert.net/.