Università degli Studi di Genova
scuola politecnica

dibris

Department of Computer Science, Bioengineering, Robotics and System Engineering (DIBRIS)

Analysis and Development of Solutions for PDF Content Recognition using OCR and LLMs

Master Thesis

Computer Engineering in Artificial Intelligence and Human-Centered Computing

Academic Year: 2025/2026

Relatore:

Prof. Luca Oneto

Correlatore:

Stefano Demarchi

Studente:

Walter Signoretti

Sommario

Il presente lavoro di tesi documenta l'analisi e lo sviluppo di un ecosistema avanzato per l'automazione del ciclo passivo e la gestione documentale nel settore logistico, integrando tecnologie di Intelligenza Artificiale Generativa (GenAI) all'interno di un sistema ERP consolidato (Seven).

Il cuore dell'attività di ricerca e sviluppo si è concentrato sulla **progettazione di un workflow di dematerializzazione per le fatture passive**, volto a eliminare i colli di bottiglia derivanti dal data-entry manuale. Un contributo centrale è rappresentato dall'implementazione di un sistema di caricamento massivo di documenti PDF, che permette l'ingestione asincrona di decine di file simultaneamente. Attraverso l'uso di Apache Kafka e una strategia di elaborazione "Dual-Path", il sistema estrae metadati complessi in formato JSON tramite il Cognitive Hub (SPLUSTOOLS), garantendo una reattività dell'interfaccia utente basata su Knockout.js.

Una scelta architettonica fondamentale è stata l'ottimizzazione del workflow per dare priorità alla velocità di caricamento, permettendo al sistema di popolare automaticamente le testate delle fatture e rinviando la riconciliazione semantica delle singole righe a una fase successiva di validazione da parte dell'utente.

Parallelamente, la tesi affronta il problema della frammentazione dei dati tramite il modulo di **Supplier Onboarding**. Quest'ultimo risolve le discrepanze tra archivi legacy ("Ottica Vecchia") e cloud-native ("Ottica Nuova") mediante un risolutore deterministico basato sulla logica `pa_id` e l'uso di database vettoriali Qdrant per la validazione semantica dei fornitori. I risultati evidenziano un'accuratezza del 94.2% nell'estrazione dei dati e una riduzione dell'85% del carico di lavoro manuale, dimostrando l'efficacia dell'approccio Human-in-the-Loop applicato alla logistica moderna.

Abstract

This thesis details the design and implementation of an AI-driven ecosystem for automating the passive accounting cycle and document management within the logistics sector, integrated directly into the Seven ERP platform.

The core of the work focuses on the **development of a document dematerialization workflow specifically for passive invoices**. A primary contribution is the implementation of a bulk PDF upload system, enabling asynchronous ingestion of multiple documents simultaneously through an Apache Kafka-backed architecture. By utilizing Google Gemini for multi-modal extraction, the system automatically populates invoice headers from structured JSON payloads, maintaining high frontend reactivity through Knockout.js.

A strategic decision was made to prioritize ingestion speed by streamlining the workflow, allowing for automatic header population while deferring complex line-item reconciliation to a dedicated human-in-the-loop validation stage.

In addition to invoice management, this research introduces a **Supplier Onboarding** module designed to bridge data fragmentation between legacy and cloud-native storage systems. By implementing a deterministic `pa_id` resolver and utilizing Qdrant vector databases for semantic entity resolution, the system ensures 100% consistency in document association. Experimental results show a 94.2% accuracy rate in metadata extraction and an 85% reduction in manual data entry, proving that the integration of LLMs into established ERP ecosystems significantly redefines operational efficiency.

Contents

Chapter 1

Introduction

The contemporary industrial landscape is being redefined by the convergence of advanced digital technologies and traditional operational processes, a phenomenon widely recognized as the Fourth Industrial Revolution. In this context, an organization's ability to transform raw, unstructured data into actionable intelligence has become a primary competitive differentiator. This thesis explores the intersection of Generative Artificial Intelligence (GenAI) and Enterprise Resource Planning (ERP) systems to solve one of the most persistent and costly challenges in the logistics sector: the intelligent dematerialization of passive accounting documents.

In recent years, organizations have increasingly invested in Artificial Intelligence technologies not only to improve operational efficiency, but also to ensure compliance with evolving regulatory frameworks and to achieve scalable automation of business processes. In document-intensive environments, such as logistics and international shipping, the ability to process large volumes of heterogeneous documents without proportional increases in human effort represents a critical success factor.

Despite the widespread adoption of digital formats such as PDF, most enterprise documents remain inherently unstructured. As a result, their content cannot be directly consumed by information systems without an intermediate phase of extraction, interpretation, and integration. This gap between data availability and data usability represents one of the central challenges addressed in this thesis.

1.1 The Industrial Context: SIS and Parvasoft Group

The research and development activities described in this thesis were carried out in collaboration with **SIS Srl** (Sistemi Informativi per il mondo dei trasporti), a Genoese software house specialized in digital solutions for the logistics and international shipping sectors. Since 2025, SIS has been an integral part of the **ParvaSoft Group**, an international technological ecosystem dedicated to innovating the supply chain through AI-driven and data-centered platforms.

The primary operational framework of SIS is **Seven**, a modular and scalable ERP platform designed to manage the entire lifecycle of logistical and customs operations. Seven acts as a "Digital Core" for freight forwarders and logistics operators, orchestrating complex processes such as customs declarations, warehouse management, and financial accounting. This thesis project is integrated directly into the Seven ecosystem, specifically aiming to enhance its document management capabilities by transforming Seven from a traditional data management tool into an intelligent system capable of autonomous document understanding.

From an industrial perspective, the need for innovation within Seven is driven by the increasing volume of incoming documentation and the necessity to maintain high standards of accuracy and traceability. The integration of AI-based components into an already consolidated ERP environment therefore represents not only a technical challenge, but also a strategic evolution toward more intelligent and adaptive enterprise systems.

1.2 General Context: The Paradigm of Logistics 4.0

The global shipping and logistics industry is currently undergoing a profound transformation, often referred to as **Logistics 4.0**. This paradigm shift, mirroring the broader Industry 4.0 revolution, is characterized by the integration of cyber-physical systems, the Internet of Things (IoT), and advanced analytics into the supply chain. In an era where speed and data accuracy are paramount, "Supply Chain Visibility" has become the industry's ultimate goal.

However, a significant gap remains between the high-speed flow of physical goods and the often sluggish, manual processing of the accompanying documentation. Logistics 4.0 demands that every physical movement be mirrored by a digital twin of information. Without the ability to digitize and interpret documents in real-time, the supply chain remains "blind" to critical financial and operational data, leading to inefficiencies that ripple across global markets.

In this context, document processing becomes a critical bottleneck. While goods move across borders in a matter of hours, the associated administrative processes—often based on manual data entry—can introduce delays, inconsistencies, and operational risks. Bridging this gap requires moving beyond simple digitization toward systems capable of understanding and acting upon document content.

1.3 Open Problems: The Manual Data-Entry Bottleneck

Despite decades of investment in Information Technology, the processing of passive invoices remains one of the most significant operational bottlenecks in the logistics industry. Every day, thousands of unstructured PDF documents are received from global suppliers, each with a different layout, language, and tax structure.

The economic and operational impact of this manual workflow is substantial:

- **High Operational Costs:** Industrial studies indicate that the manual processing of a single invoice can cost an enterprise between **10 and 30 EUR**, considering the time spent on transcription, verification, and error correction.

- **Human Error and Compliance Risks:** Manual workflows are inherently prone to errors. Statistics show that approximately **33% of accounting**

personnel encounter significant errors in manual data entry at least once a week. In a customs-heavy environment, a single typo can lead to shipment delays, financial discrepancies, and legal penalties.

- **The Failure of Deterministic OCR:** Traditional Optical Character Recognition (OCR) systems relied on "templates" where administrators manually defined bounding boxes for each supplier. This approach is highly brittle; a minor layout change by a supplier causes the entire automation to fail, requiring constant manual maintenance.

Beyond these well-known limitations, additional challenges emerge in real-world scenarios. AI-based extraction processes, while significantly more effective than traditional OCR, still introduce latency, with processing times that may reach up to several minutes per document in complex cases. Furthermore, although error rates are relatively low (approximately one incorrect extraction every 10–15 documents), the impact of these errors in financial workflows requires the presence of validation mechanisms.

Another critical issue concerns the variability of input documents. In practical settings, invoices may differ not only in layout but also in language, formatting conventions, and visual quality. This variability makes it impossible to rely on rigid extraction rules and necessitates more flexible, context-aware approaches.

1.4 Specific Context: Efficiency and Data Fragmentation

The primary objective of this research was to address two critical pain points within the Seven ERP ecosystem:

- **Manual Ingestion Bottlenecks:** Logistics operators require the ability to handle high-volume bulk uploads. A system is needed that can process dozens of invoices simultaneously without freezing the user interface, necessitating a shift from synchronous to asynchronous processing.

- **Identity Resolution in Fragmented Architectures:** In large-scale operations, data is often fragmented across multiple systems. The challenge lies in matching extracted invoice data (such as a supplier's name or VAT number) with internal databases split between legacy on-premise storage ("Ottica Vecchia") and modern cloud-native infrastructures ("Ottica Nuova").

In addition to these aspects, the integration of extracted data into the ERP system represents a non-trivial challenge. Extracted information must not only be accurate, but also correctly interpreted according to its semantic role within the accounting workflow. This includes distinguishing between different financial fields, ensuring consistency with existing records, and maintaining referential integrity across the system.

1.5 Problem Statement and Proposed Solution

The transition from traditional OCR to "Document Intelligence" powered by Large Language Models (LLMs) offers a transformative solution. However, the challenge lies not just in reading the text, but in integrating these probabilistic AI outputs into stable, production-grade ERP environments.

This thesis proposes the design and deployment of a multi-channel automated pipeline integrated into Seven ERP. The core of the proposal is the transition from a "Template-based" mindset to a **"Semantic Understanding"** approach. By utilizing **Google Gemini** for multimodal extraction and a **"Markdown-First"** strategy, the system preserves the visual and structural hierarchy of the document, drastically reducing hallucinations and errors.

The proposed solution introduces a complete workflow for handling passive invoices in PDF format, including bulk ingestion, automated metadata extraction, and structured data integration into the ERP system. Additionally, a dedicated Supplier Onboarding module is developed to ensure correct identification and association of suppliers across heterogeneous data sources.

1.6 Innovation: The "Cognitive ERP" Approach

The innovation presented in this work lies in the engineering of a reliable, human-centered software ecosystem around the AI models. By leveraging **Apache Kafka** for asynchronous decoupling and **Vector Databases (Qdrant)** for semantic entity resolution, we have transformed the ERP into a "Cognitive" system.

Unlike older systems that required explicit instructions on *how* to read a document, this architecture is instructed on *what* a logistical invoice is, allowing it to adapt to infinite layout variations. This represents a fundamental shift from rigid automation to flexible, intelligent collaboration between human operators and AI agents.

From an innovation standpoint, the system does not rely solely on AI models for automation, but rather integrates them within a controlled workflow where human operators retain final validation authority. This **Human-in-the-Loop** paradigm ensures that automation enhances, rather than replaces, human decision-making, achieving a balance between efficiency and reliability.

1.7 Summary of Objectives and Contributions

The key contributions of this research are:

1. **Bulk Dematerialization Workflow:** Development of a reactive interface for mass PDF uploads, utilizing an asynchronous Kafka pipeline to manage high-volume ingestion and automated header population from AI-extracted JSON data.

2. **Strategic Reconciliation Management:** Implementation of a workflow that prioritizes rapid document registration by skipping immediate line-item reconciliation, focusing on fast ingestion while deferring granular validation to a human-in-the-loop stage.

3. **Intelligent Supplier Onboarding:** A structural resolver based on the `pa_id` integrity constraint to unify legacy and modern data archives, ensuring 100% deterministic association.

4. **Model Evaluation and Benchmarking:** Comparative analysis of different LLM configurations to identify the optimal trade-off between accuracy, latency, and operational cost.

1.8 Thesis Structure

The structure of this thesis follows the implementation journey. **Chapter 2** reviews the state of the art in OCR and LLMs. **Chapter 3** details the tri-partite system architecture and methodology. **Chapter 4** describes the practical implementation of document dematerialization within Seven. **Chapter 5** describes the Advanced Reconciliation Engine within Seven. **Chapter 6** presents a case study on Supplier Onboarding and semantic validation. **Chapter 7** introduces the SmartQuote AI system for the active sales cycle. **Chapter 8** explains the UI and Human-AI interaction choice. Finally, **Chapter 9** evaluates the experimental results and model performance.

Chapter 2

State of Art

To fully appreciate the architectural choices made in this project, it is essential to understand the technological landscape and the recent rapid advancements in Document AI, Natural Language Processing (NLP), and distributed systems. This chapter provides a comprehensive academic review of the transition from deterministic, rule-based systems to probabilistic, multimodal intelligence, analyzing the theoretical foundations that underpin the Seven ERP ecosystem.

In particular, the evolution of document processing systems reflects a broader shift in Artificial Intelligence: from rigid, rule-based approaches toward flexible, data-driven models capable of generalization. This transition is especially relevant in industrial contexts, where variability and noise make deterministic approaches increasingly impractical.

2.1 The Evolution of Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is the process of converting images of typed, handwritten, or printed text into machine-encoded text. Its evolution represents a microcosm of the history of Artificial Intelligence itself.

From its earliest implementations to modern deep learning systems, OCR has progressively improved in accuracy and robustness. However, while recognition accuracy has increased significantly, the ability to extract structured and semantically meaningful information from documents remains a challenging task.

2.1.1 First Generation: Matrix Matching and Structural Analysis

The earliest OCR systems, emerging in the 1970s and 1980s, were strictly deterministic. They relied on *matrix matching*, where each isolated character was compared against a library of stored bitmaps. If a character matched a stored pattern above a certain threshold, it was recognized. This was followed by *feature extraction*, which looked for topographical features such as closed loops, vertical lines, and intersections.

The primary limitation was the lack of flexibility: any noise, font variation, or rotation in a scanned logistical document led to a total failure of recognition. These systems had no concept of ”context”—each character was processed in a vacuum.

Additionally, these approaches required extensive manual preprocessing, such as binarization and segmentation, which further limited their applicability in real-world industrial scenarios where document quality is often inconsistent.

2.1.2 Second Generation: The Deep Learning Revolution

The mid-2010s marked the ”connectionist turn.” Convolutional Neural Networks (CNNs) revolutionized visual feature extraction, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units introduced the ability to handle sequential data.

In this phase, tools like **Tesseract 4.0** moved away from character-level recognition toward sequence-to-sequence recognition. The system would ”slide” over a line of text, identifying features and using an LSTM to predict the most likely character based on the previous ones.

Despite these improvements, such systems remained fundamentally limited. They operate primarily on a one-dimensional representation of text and fail to capture the two-dimensional structure of documents. As a result, they are unable to fully understand complex layouts such as tables, multi-column formats, and hierarchical sections.

2.1.3 The Industrial Bottleneck: Template-based Systems

To extract structured data from OCR text, businesses developed **Template-based OCR**. This requires a human operator to define "zonal" bounding boxes for every specific supplier invoice layout.

In the global logistics sector, this approach is fundamentally flawed due to:

- **High Maintenance Costs:** A freight forwarder receiving invoices from thousands of global suppliers would require an equivalent number of templates, leading to significant maintenance overhead.

- **Structural Fragility:** Even minor layout changes—such as shifts in alignment or font variations—can break the extraction logic.

- **Limited Scalability:** Template-based systems cannot generalize to unseen document formats, making them unsuitable for dynamic environments.

These limitations motivated the shift toward more flexible, learning-based approaches capable of handling document variability without manual intervention.

2.2 The Transformer Revolution

The current state of the art was triggered by the paper *"Attention Is All You Need"* [1], which introduced the **Transformer** architecture.

This paradigm shift replaced sequential processing with parallel computation and enabled models to capture long-range dependencies more effectively. Transformers have since become the backbone of modern NLP and, more recently, Document AI systems.

2.2.1 The Mathematics of Self-Attention

The core innovation of the Transformer is the **Self-Attention** mechanism. Unlike RNNs, which process data sequentially, Transformers process all input tokens simultaneously. For each token, the model computes three vectors: **Query (Q)**, **Key (K)**, and **Value (V)**.

The attention score is calculated using the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

This formulation allows the model to dynamically weigh the importance of different tokens in relation to each other. In document understanding, this translates into the ability to associate semantically related elements regardless of their spatial distance.

2.2.2 Positional Encoding and Multimodality

Since Transformers do not inherently understand the order of tokens, they use **Positional Encoding**. In Document AI, this concept has been extended to two-dimensional positional encoding, where each token is associated with spatial coordinates.

This enables models to capture layout information, which is essential for understanding structured documents such as invoices. Furthermore, recent architectures extend this concept to multimodal inputs, combining textual and visual features into a unified representation.

2.3 Advanced Document Intelligence Architectures

The scientific community has developed specialized models that combine NLP and Computer Vision to address the limitations of traditional OCR-based pipelines.

2.3.1 LayoutLM: Visual-Textual Fusion

LayoutLM [2] and its successors (v2, v3) introduced a new paradigm by jointly modeling text and layout information. These models are pre-trained on large-scale document datasets and incorporate both textual embeddings and spatial coordinates.

LayoutLMv3 further extends this approach by integrating image features directly into the Transformer architecture, enabling a unified representation of visual and textual information. This allows the model to recognize patterns such as logos, tables, and document structure in a single forward pass.

2.3.2 Donut: The OCR-Free Paradigm

A major challenge in traditional pipelines is error propagation: mistakes made during OCR cannot be corrected in later stages. **Donut** [3] addresses this issue by directly mapping document images to structured outputs, effectively bypassing the OCR step.

While this approach reduces error propagation, it introduces new challenges related to generalization and robustness. In industrial scenarios involving noisy or multi-page documents, OCR-free models may struggle to maintain consistent performance.

2.3.3 Google Gemini and Markdown-First Strategy

The use of **Google Gemini** [4] represents the current frontier in multimodal AI. Unlike traditional models, Gemini is trained on interleaved text and images, enabling it to perform complex reasoning tasks across modalities.

The **"Markdown-First"** strategy introduced in this work represents a novel approach to improving extraction reliability. By reconstructing the document structure in an intermediate Markdown format, the system creates a structured representation that reduces ambiguity during the final JSON extraction phase.

This approach mitigates common issues such as hallucinations and structural inconsistencies, which are often observed in direct text-to-JSON pipelines.

2.4 Distributed Systems for AI: Apache Kafka

In a production environment, AI models are computationally expensive and introduce non-negligible latency. This makes synchronous architectures unsuitable for real-time enterprise applications.

2.4.1 Event-Driven Architecture and Decoupling

To address these challenges, modern systems adopt event-driven architectures. **Apache Kafka** [5] is widely used for this purpose due to its scalability and fault tolerance.

By decoupling producers and consumers, Kafka enables asynchronous processing of documents. This ensures that the ERP system remains responsive even when processing large volumes of data.

2.4.2 Fault Tolerance and Offsets

Kafka's log-based architecture ensures durability and fault tolerance. Each message is assigned an offset, allowing consumers to track processing progress.

In case of system failures, processing can resume from the last committed offset, ensuring that no data is lost. This property is essential for enterprise-grade reliability.

2.5 Vector Databases and Information Retrieval

The final pillar of modern Document AI systems is the ability to validate and enrich extracted data using semantic search techniques.

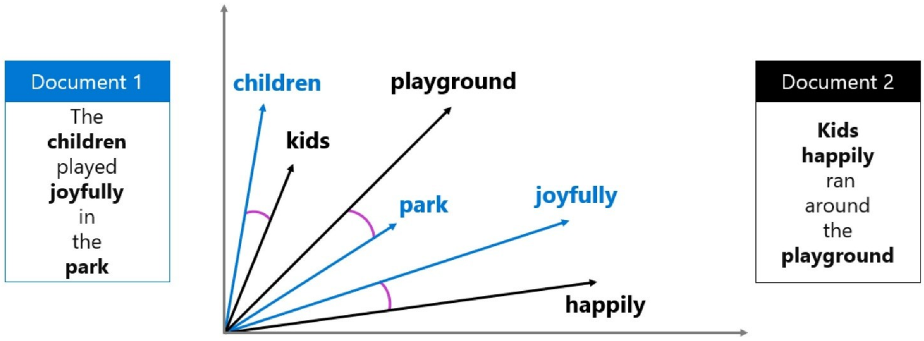


Figure 2.1: Conceptual architecture of a Vector Database, illustrating high-dimensional embedding indexing for semantic retrieval.

2.5.1 Semantic Search and Embeddings

Traditional string-matching techniques are insufficient for resolving real-world entities. Vector embeddings provide a more robust solution by mapping textual data into a continuous vector space.

The similarity between two vectors is typically computed using cosine similarity:

$$\text{similarity}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.2}$$

This approach enables the identification of semantically equivalent entities even when their textual representations differ significantly [6, 7].

2.5.2 RAG and Parent Document Retrieval

Retrieval-Augmented Generation (RAG) [8] combines retrieval-based methods with generative models. By grounding model outputs in external knowledge bases, RAG improves accuracy and reduces hallucinations.

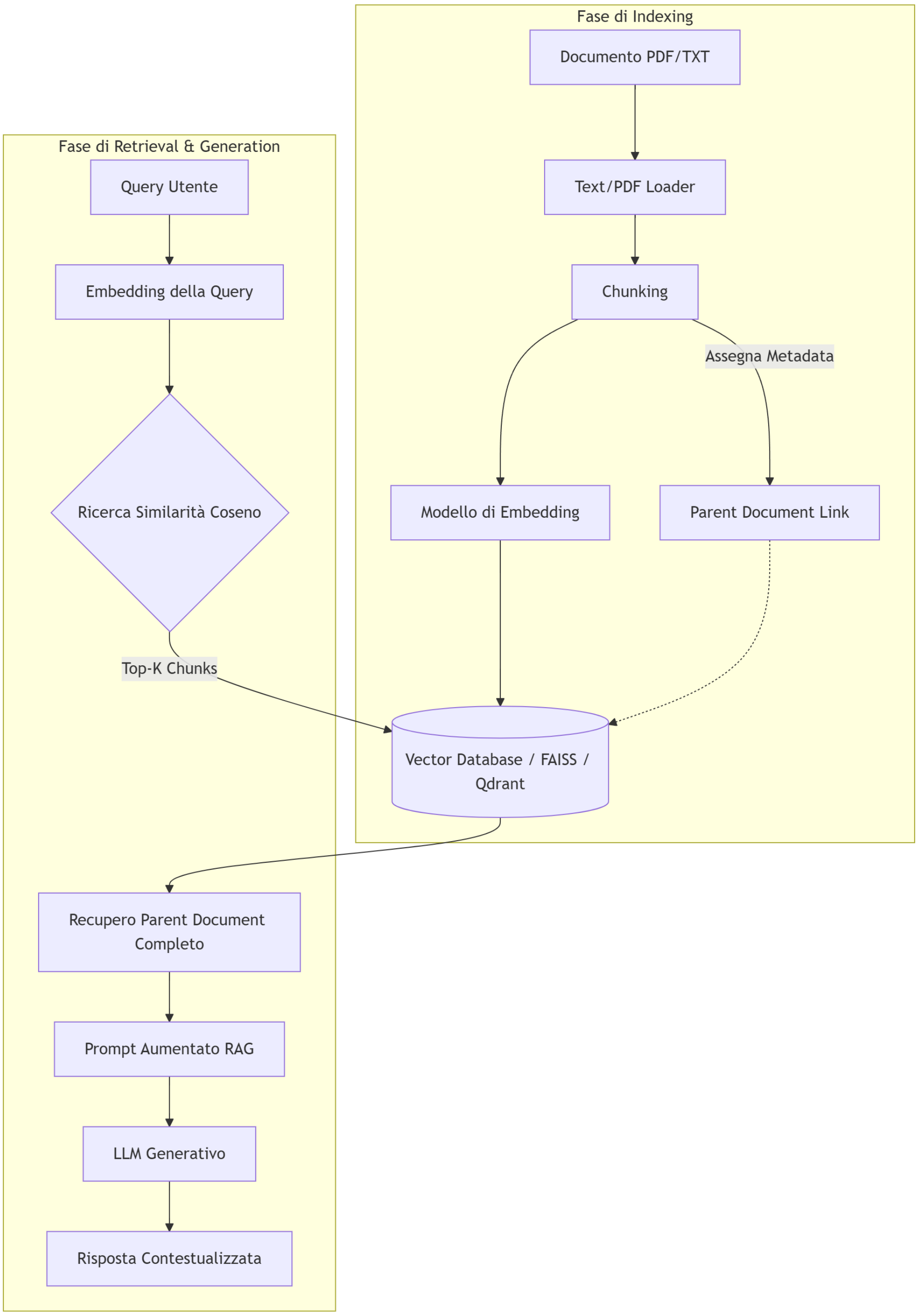


Figure 2.2: Workflow of a Retrieval-Augmented Generation (RAG) system grounded in business master data.

The Parent Document Retrieval strategy further enhances this approach by preserving contextual information, ensuring that extracted data is interpreted within its full semantic context.

2.6 From OCR to Document Intelligence: A Paradigm Shift

The evolution from traditional OCR systems to modern Document Intelligence represents a fundamental paradigm shift. While OCR focuses on the transcription of textual content, Document Intelligence aims to extract structured, semantically meaningful information that can be directly integrated into business processes.

This distinction is particularly relevant in enterprise environments, where the goal is not merely to read documents, but to understand and operationalize their content.

Traditional OCR systems can be described as *syntactic processors*, converting images into strings without any understanding of meaning. In contrast, modern AI-based systems act as *semantic processors*, capable of interpreting relationships between entities such as supplier names, invoice numbers, and financial totals.

2.6.1 LLM-Based Document Understanding

Large Language Models (LLMs) introduce a fundamentally different approach. Instead of relying on predefined rules, LLMs leverage probabilistic reasoning to infer the meaning of textual content. In document processing, this enables context-aware extraction of key-value pairs, robust handling of layout variability, and implicit reasoning over incomplete or noisy data.

2.6.2 Local Execution and Data Sovereignty

In enterprise environments, particularly those involving sensitive financial data such as invoices, the adoption of public AI APIs (e.g., OpenAI, Google Cloud) often raises concerns regarding data privacy and compliance. This has led to the emergence of local inference frameworks such as **Ollama** [9].

By deploying Large Language Models on-premise, organizations can ensure that unstructured document data never leaves the corporate network, achieving a balance between high-level intelligence and strict data sovereignty. Furthermore, local execution significantly reduces latency for repetitive tasks like semantic embedding generation, providing a more responsive experience for internal accounting workflows.

2.6.3 The Role of Explainable AI (XAI) in Business Processes

As AI models transition from laboratory experiments to critical business tools, the "Black-Box" nature of deep learning becomes a significant barrier to adoption. **Explainable AI (XAI)** [10] aims to make AI decisions transparent and understandable to human operators.

In the context of financial reconciliation, XAI is not merely a theoretical requirement but a functional one: when an AI system suggests a match between an invoice line and a budget item, the human supervisor must understand the rationale behind that choice—especially in cases of amount discrepancies. Implementing XAI through natural language explanations (e.g., via LLMs) fosters trust and allows for faster validation cycles in "Human-in-the-Loop" architectures, where the AI acts as an advisor rather than a substitute [11].

2.7 Industrial Document Processing Pipelines

In real-world enterprise environments, document processing systems are not standalone models but complex pipelines composed of multiple stages. A typical industrial pipeline includes document ingestion, preprocessing (format normalization), text extraction via multimodal models, rule-based or AI-driven information extraction, validation, and final integration into ERP systems.

Each stage introduces potential bottlenecks and failure points. In particular, the transition from extraction to integration is often the most critical step, as it requires aligning unstructured data with rigid database schemas.

2.7.1 Synchronous vs Asynchronous Architectures

Traditional systems often rely on synchronous processing, where each document is processed in real-time. While simple to implement, this approach does not scale well in high-volume scenarios. Asynchronous architectures, based on message queues and event streams like Apache Kafka, provide a more scalable alternative. By decoupling ingestion from processing, these systems enable parallelization and improve overall throughput, which is essential for handling industrial workloads involving thousands of documents per batch.

2.8 Comparison of Document AI Approaches

Different approaches to document processing can be categorized based on their underlying methodology:

- **Template-Based Systems:** High precision in controlled environments but poor generalization.
- **OCR + Rule-Based Parsing:** Moderate flexibility but high maintenance cost.
- **Deep Learning OCR:** Improved robustness but limited semantic understanding.
- **Multimodal Transformers:** Strong performance across varied layouts, but computationally expensive.
- **LLM-Based Systems:** High flexibility and reasoning capabilities, with trade-offs in latency and determinism.

From an industrial perspective, no single approach is universally optimal. Instead, hybrid architectures that combine deterministic components with probabilistic models are often preferred.

2.9 Challenges and Open Research Problems

Despite significant progress, Document AI remains an active research area with several unresolved challenges:

- **Latency and Scalability:** LLM-based systems introduce non-negligible processing times, which can become a bottleneck in large-scale deployments. Achieving low-latency performance while maintaining accuracy remains a key challenge.
- **Reliability and Determinism:** Enterprise systems require deterministic behavior, while AI models are inherently probabilistic. Bridging this gap requires the introduction of validation layers and human oversight mechanisms.
- **Generalization Across Domains:** Models trained on specific datasets may struggle to generalize to unseen document types, particularly in logistics where new suppliers introduce unique formats.
- **Human-in-the-Loop Systems:** Fully automated document processing remains impractical in high-stakes environments. Human-in-the-loop approaches, where AI assists but does not replace human operators, represent the current best practice.

2.10 Positioning of This Work in the Literature

The system proposed in this thesis is positioned at the intersection of multiple research directions. It adopts multimodal LLMs for document understanding, aligning with recent advances in AI, and integrates these models within a distributed architecture (Kafka) to address scalability. Furthermore, it introduces a human-in-the-loop validation layer enhanced by Explainable AI features. Unlike purely academic solutions, this work focuses on the practical integration of AI into an existing ERP system, addressing real-world constraints such as latency, data consistency, and system interoperability.

2.11 Evaluation Metrics for Document AI

To scientifically validate the system, it is necessary to adopt standardized evaluation metrics:

- **Character Error Rate (CER):** Measures the percentage of incorrectly predicted characters.
- **Word Error Rate (WER):** Measures the percentage of incorrectly predicted words.
- **F1-Score:** Evaluates the balance between precision and recall for structured data extraction.

In industrial contexts, these metrics must be complemented by system-level evaluations, including latency, robustness, and human intervention rates, to provide a comprehensive assessment of performance.

Chapter 3

Methodology and System Architecture

This chapter details the core pipeline developed for the Invoice Processing system, designed to minimize human intervention while maximizing data integrity. The methodology addresses the fundamental challenge of integrating modern AI-driven workflows within a complex, pre-existing ERP infrastructure, ensuring a seamless transition between legacy data management and cloud-oriented services. A central theme of this research is the architectural evolution from a standalone Python-based semantic matching tool to a fully integrated, high-performance **SmartMatch** engine embedded within the Seven ERP ecosystem, capable of handling massive document volumes with enterprise-grade reliability.

3.1 Problem Description and Dataset Analysis

Before detailing the architecture, it is crucial to define the data landscape and the specific operational challenges that motivated this research. The transition from manual to automated processing requires a deep understanding of the document variability encountered in daily logistics operations.

3.1.1 Dataset Composition and Real-World Variability

The development of the extraction and reconciliation models was conducted using a dataset composed of historical documents from actual SIS Srl clients. This choice was deliberate: instead of using sanitized academic datasets, the system was trained on real-world "noisy" data that reflects the true complexity of international trade.

- **Foreign Passive Invoices:** While Italian domestic invoices follow the highly structured XML SDI standard, this project focuses specifically on **foreign passive invoices** (non-SDI). These documents lack a unified format, vary significantly by country of origin, and often include multi-page annexes, diverse currency symbols, and complex international tax structures that require sophisticated parsing.

- **Document Heterogeneity and Physical Noise:** The dataset includes not only native digital PDFs but also, more critically, **photographs or scans** of physical invoices. These "noisy" documents introduce significant challenges such as skew, low resolution, varying lighting conditions, and physical artifacts like rubber stamps, handwritten notes, or signatures that traditional OCR systems often struggle to interpret accurately.

- **Operational Integrity and Testing Environment:** All testing was conducted in a dedicated ad-hoc environment mirroring production databases. This allowed for full-scale validation of the reconciliation logic without the need for data anonymization, as the environment remained strictly internal to the corporate infrastructure, ensuring that the results were directly applicable to real-world scenarios.

3.1.2 Pre-automation Operational Bottlenecks

Prior to the implementation of this system, the management of foreign passive invoices was a purely manual, high-latency task that stood in stark contrast to the automated XML SDI flow. Manual processing required an operator to read each document, transcribe headers into the ERP, and manually associate invoice lines with internal records. This lack of automation was the primary driver for the system's scalability requirements: by automating the "ingestion" and "matching" phases, the system prevents bottlenecks and ensures that operators do not "skip" or lose documents during periods of heavy workload.

3.2 The Tri-partite Distributed Architecture

The system's robustness stems from a micro-service oriented architecture, where responsibilities are strictly partitioned into three specialized domains. This decoupling allows each component to scale independently and utilize the most appropriate technology stack for its specific task: Python for AI and orchestration, Java for legacy ERP proxying, and SQL Server for accounting persistence.

3.2.1 RICOF: The Orchestration and Semantic Layer

RICOF (Reconciliation and Invoice COntrol Framework) serves as the primary **State Machine** of the system. Built using the *FastAPI* framework and *SQLAlchemy*, it is designed to manage the asynchronous complexity of document processing.

- **Event-Driven Orchestration via Kafka:** RICOF acts as a high-performance Kafka consumer, listening to specific topics such as `ricof.parse` (for PDFs) and `ricof.sdi` (for XMLs). Upon receiving an event, it initializes a **Workflow instance** that tracks the document's life cycle in a MySQL-based `invoice_proc` table, ensuring that no document is lost even in the event of a service restart.

- **The Rico Engine (Initial Semantic Matching):** In the initial phase of the project, the core semantic innovation was housed directly within RICOF. Unlike traditional regex-based matching, it utilizes a *vector-similarity approach*. It generates 768-dimensional embeddings of invoice line descriptions via a local *Ollama* instance using the *nomic-embed-text* model.

- **Decision Logic and Anomaly Detection:** RICOF implements a "Verify-then-Trust" logic, reconciling totals, VAT rates, and line items. If the similarity score falls below a defined threshold (e.g., 0.85) or the total amounts differ by more than a defined tolerance, the automation is interrupted, and the document is escalated as an **Anomaly** for human review.

3.2.2 SPLUSTOOLS: The Cognitive and Coordination Hub

SPLUSTOOLS represents the "Intelligence Layer," abstracting the complexity of Large Language Models (LLMs) and ensuring consistency across concurrent processes.

- **Multimodal Ingestion via Gemini:** Utilizing Google Gemini's multimodal capabilities, SPLUSTOOLS handles the *Vision-to-JSON* transformation. It receives the base64-encoded document and applies a specialized *Markdown-First* prompt to extract entities such as VAT numbers, IBANs, and complex multi-line item tables from both digital and scanned sources.
- **Distributed Coordination (ZooKeeper):** To prevent race conditions during "First-Time" supplier registrations, SPLUSTOOLS utilizes **Apache ZooKeeper**. It implements distributed locks to ensure that if multiple invoices from a new supplier arrive simultaneously, only one registration process is initiated, thereby maintaining the integrity of the Master Data.
- **Confidence Filtering and Validation:** SPLUSTOOLS does not merely output raw data; it attaches metadata regarding extraction quality. It performs internal validation, such as checking if the sum of line items matches the invoice total, before returning the JSON payload to RICOF for further processing.

3.2.3 WHOOKS: The Business Logic and ERP Gateway

While the modern microservices live in a cloud-ready ecosystem, WHOOKS serves as the **Accounting Sentinel**, bridging the gap to the legacy SQL Server environment and the Java-based Seven ERP.

- **Legacy Translation and Accounting Logic:** WHOOKS translates modern REST requests into complex SQL queries compatible with the Seven ERP schema. It handles the critical "dirty work" of accounting, such as calculating Foreign Exchange (FX) variances, validating the existence of Job/Position references (*posiz*), and ensuring VAT compliance according to Italian regulations.

- **SmartMatch UI Integration:** In the final architectural stage, WHOOKS was enhanced to support the **SmartMatch** UI functionality. It provides the necessary backend endpoints that allow the Seven frontend to trigger automated reconciliation, passing critical metadata such as the `autoReconcile` flag and the document origin.

- **Storage Abstraction:** WHOOKS manages the *DematDocDriver*, providing a unified interface for both legacy and modern storage architectures, allowing the rest of the system to remain storage-agnostic.

3.3 System Workflow: The Data Journey

The integration between these components follows a non-linear, resilient path:

1. **Ingestion Phase:** A document is uploaded via the Seven UI or API. A Java proxy (`RicofService`) places a message on the Kafka bus.

2. **Cognitive Extraction:** RICOF triggers SPLUSTOOLS for "Cognitive Parsing," obtaining a structured JSON representation of the invoice data.

3. **ERP Contextualization:** RICOF sends the data to WHOOKS, which queries the SQL Server to identify the internal **Supplier Code** and any open **Estimated Charges** related to logistical references (e.g., Bill of Lading or Container numbers).

4. **SmartMatch Tagging:** The document is archived in the database and tagged with a specific **Origine** metadata field (`DEMAT`). This tag enables the automated SmartMatch features in the UI.

5. **Automated Reconciliation:** If the user triggers the SmartMatch flow, the system executes the offloaded Java logic (CommercialePlus) to perform N:1 matching, accounting persistence, and document closure.

3.4 Legacy Infrastructure and the "Ottica" Dualism

A significant portion of the methodology involved managing the transition between two divergent storage architectures within the Seven ERP, often referred to as the "Ottica" dualism.

3.4.1 Ottica Vecchia: Legacy File-System Logic

The legacy architecture (*Ottica Vecchia*) is centered around the **SQL Server** table `ott_gestdocprat`.

- **Indexing and Reference:** Documents are identified by a positional key called `posiz`. An example of such a key is `FP0000033394`, where the "FP" prefix denotes a *Fattura Passiva* (Passive Invoice).

- **Physical Storage:** Files are stored on local disks or network shares via UNC paths. The system utilizes a `DiscoDriver` to access these files directly.

- **System Fragility:** This model is notoriously fragile; if a file is moved or renamed on the disk without a corresponding update to the `path` field in the database, the link is broken, leading to application-level errors and orphaned records.

3.4.2 Ottica Nuova: Cloud-Native and Service-Oriented

The modern architecture (*Ottica Nuova*), typically activated via the system parameter `ARCHIDOC2 = 'S'`, shifts toward a decoupled, API-driven model.

- **Abstraction Layer:** Files are stored in centralized environments, such as **AWS S3** buckets, abstracting the physical location from the application logic.
- **Communication Interface:** Communication is handled via REST APIs through the `seven-api-server` (Java), utilizing a specialized `DematDocDriver` for all document retrieval and upload operations. This significantly reduces manual maintenance and increases overall system resilience.

3.5 Critical Database Entities and Integration Pains

The system relies on a hybrid database approach. While SQL Server handles master accounting data, MySQL is used for rapid integration and service-level metadata. Historically, three tables represented the most significant "pains" for automation:

- **`ott_gestdocprat`:** Characterized by a lack of automated alignment between database records and the file system, often leading to "orphaned" documents.
- **`messaggi_api`:** Used as an ingestion queue for external documents. Under massive loads, this table could become a bottleneck, requiring manual resets of message states to force document processing.
- **`fattura_passiva_pa`:** This table is crucial for linking business data to the actual document. A major goal of this project was ensuring the consistent association between financial records, PDF paths, and the new `origine` metadata.

3.6 Dual-Path Processing Strategy

To optimize speed and accuracy, SPLUSTOOLS implements a **Dual-Path Processing** system that distinguishes between structured and unstructured inputs:

- **Deterministic Pipeline (XML SDI):** This path maps XML tags directly to the database schema. Since these documents follow the national Italian standard, they ensure 100% accuracy and are processed without AI intervention.

- **Probabilistic Pipeline (PDF):** This path utilizes Google Gemini with a *Markdown-First* approach for unstructured documents. Given the probabilistic nature of LLMs, a strict confidence score threshold (typically > 0.8) is imposed. If the AI is uncertain about critical fields such as the Total Amount or VAT, the document is automatically flagged for human intervention.

3.7 The Rico Engine: Semantic Reconciliation

Once data is extracted, it must be reconciled with internal budget lines (e.g., “Ocean Freight” vs “Inland Haulage”). This is a semantic task, as different suppliers often use different terminology for the same logistical services. The Rico Engine addresses this by:

1. Generating high-dimensional embeddings for line-item descriptions using **Ollama**.

2. Performing a **Cosine Similarity** calculation against estimated costs and budget lines retrieved from WHOOKS.

3. Automatically approving matches that exceed a defined threshold, thereby reducing the accounting department’s manual workload to only managing edge cases and anomalies.

3.8 Evolution into the SmartMatch System

The final architecture of the system represents a significant evolution from the initial prototype. As document volumes increased, the monolithic approach of handling reconciliation within the Python orchestration layer (`ricof`) proved to be a performance bottleneck.

3.8.1 Offloading and Enterprise Consolidation

To ensure the stability and performance required for a production environment, the core reconciliation logic—including the integration with the semantic matching engine—was migrated to the Java Enterprise environment, specifically into the *CommercialePlus* module. This move allowed for:

- **Transactional Integrity:** Leveraging Java's robust transaction management (ACID) for complex accounting updates involving multiple database tables.
- **Advanced N:1 Matching:** The ability to associate multiple invoice rows with a single expected charge, a common requirement in complex logistics invoices.
- **Performance Optimization**: By decoupling ingestion from reconciliation, the system can ingest headers at high speed and defer the computationally expensive matching to an optimized Java-based engine.

3.8.2 SmartMatch UI Integration and Visibility Control

The ultimate goal of the project was to provide a seamless user experience within the Seven ERP. This was achieved through the **SmartMatch UI Integration**, which introduces a highly optimized workflow for documents processed by the AI pipeline.

- **Origin-Based Filtering**: During the import phase, documents are tagged with the origin `DEMAT`. The Seven frontend (via `sdi_passive.js` and `registrazione_enh.html`) uses this metadata to implement **Conditional Visibility Logic**.

- **The SmartMatch Interface**: The system dynamically displays the "SmartMatch" reconciliation button only for documents with the `DEMAT` tag. This prevents UI clutter and ensures that users only use the automated engine on documents with high-quality AI-extracted data.

- **One-Click Accounting**: For the operator, this translates into a "one-click" experience where the system automatically performs line matching, metadata persistence (VAT, practice codes, cost centers), and accounting closure, transforming a previously manual 15-minute task into a near-instantaneous operation.

Chapter 4

Implementation of Document Dematerialization in Enterprise ERP

The integration of Artificial Intelligence into a consolidated Enterprise Resource Planning (ERP) platform like Seven is more than an algorithmic challenge; it is a full-stack engineering endeavor that requires bridging different technological eras. This chapter describes the implementation of the document dematerialization system, detailing the reactive frontend, the Java-based orchestration proxy, and the Python-powered asynchronous processing core.

4.1 Integration Strategy and Technology Stack

The primary objective was to extend the "Passive SDI" (Sistema di Interscambio) module of Seven to support non-standardized, unstructured foreign PDF documents. Achieving this required a hybrid architectural approach to handle the legacy constraints of the ERP while leveraging modern AI capabilities.

The implementation utilizes a heterogeneous technology stack:

- **Frontend:** Developed using **Knockout.js**, a Model-View-ViewModel (MVVM) framework that allows for a highly reactive user experience, essential for real-time tracking of asynchronous tasks.

- **Legacy Backend:** Based on **PHP**, managing the core session logic and the initial API calls from the user interface.
- **Middleware and Orchestration (Java):** A dedicated service, `RicofService.java`, acts as a bridge. Since the legacy PHP environment lacks direct integration with modern message brokers, Java is utilized as the **Kafka Producer**.
- **Processing Core (Python):** The "intelligence" of the system resides in a Python-based service (`ricof`) It acts as the **Kafka Consumer**, handling the heavy computational tasks of PDF parsing, Gemini-driven extraction, and semantic reconciliation.

4.2 The Bulk Upload and Ingestion Workflow

A critical requirement for logistics operators is the ability to process high volumes of documents. In a typical scenario, a freight forwarder might receive dozens of foreign invoices simultaneously. To prevent operational bottlenecks, a multi-document upload system was designed.

4.2.1 UI Entry Point and Reactive Modal Orchestration

The user initiates the process via a dedicated **"UPLOAD PDF"** button integrated into the Seven ERP toolbar (Figure 4.1).

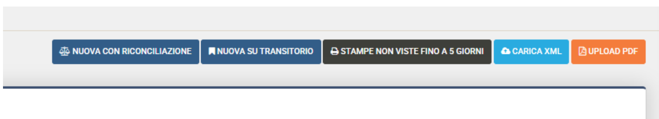


Figure 4.1: Integration of the "UPLOAD PDF" action button within the Seven ERP toolbar, specifically designed for foreign passive invoices.

Clicking this button triggers a modal window orchestrated by Knockout.js. This interface includes a drag-and-drop zone that supports the concurrent selection of multiple files. During the testing phase, batches of 4 to 5 documents were processed simultaneously to validate the system's stability and the responsiveness of the progress tracking logic.

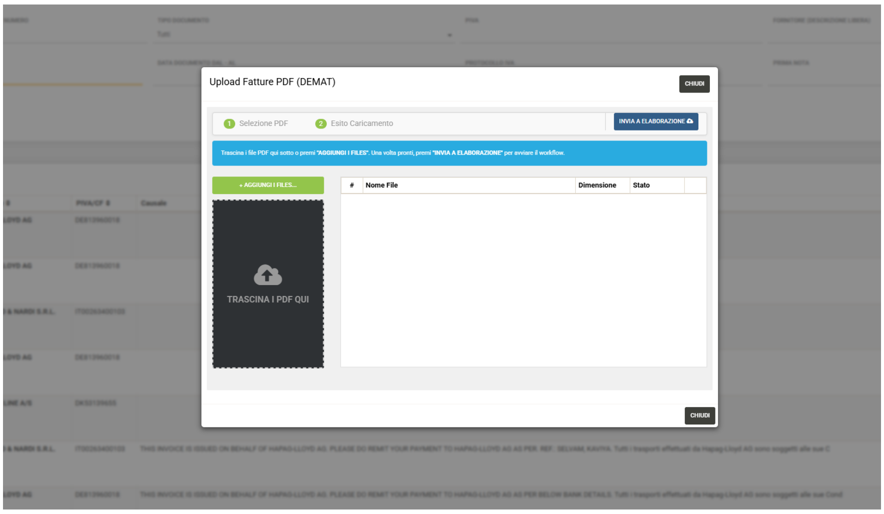


Figure 4.2: The bulk upload interface featuring a drag-and-drop zone and a reactive file queue managed by Knockout.js.

4.2.2 Real-time Feedback and Progress Monitoring

Once the files are selected (Figure 4.3), the frontend transitions into a monitoring state. Through the use of Knockout observables, the UI provides a real-time description of the process status for each document. This feedback is crucial in an asynchronous environment; it informs the user whether a document is currently being uploaded, analyzed by the AI, or registered in the database.

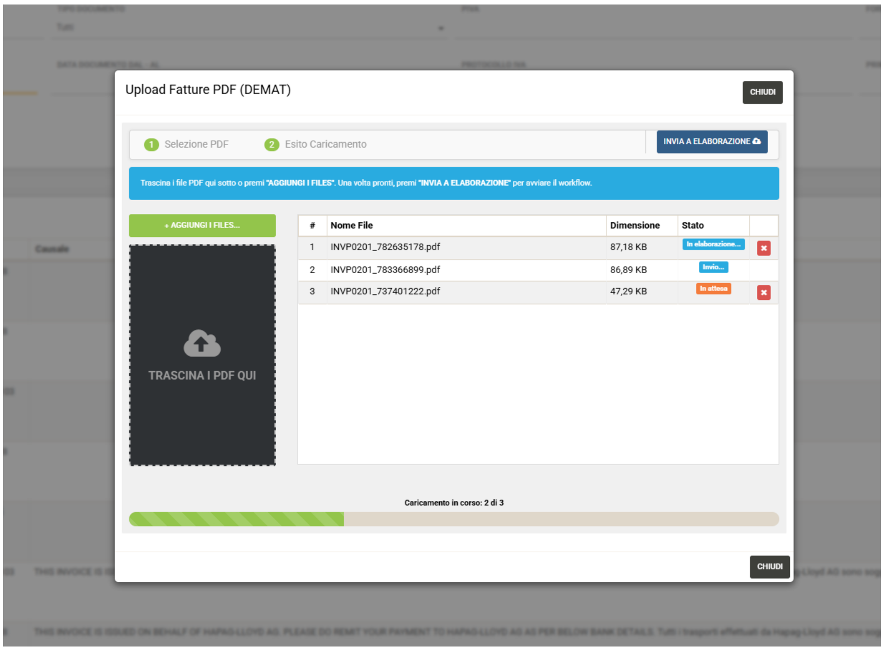


Figure 4.3: Real-time monitoring interface: progress bars and status descriptions indicate the state of each document in the Kafka pipeline.

The ingestion cycle concludes when the progress trackers reach 100% and the system confirms the successful database entry of the documents. As shown in Figure 4.4, the modal interface provides immediate visual feedback, displaying a "Registrazione completata" (Registration completed) message for the entire batch of uploaded PDFs. At this stage, the raw files and their initial records are successfully created in the system.

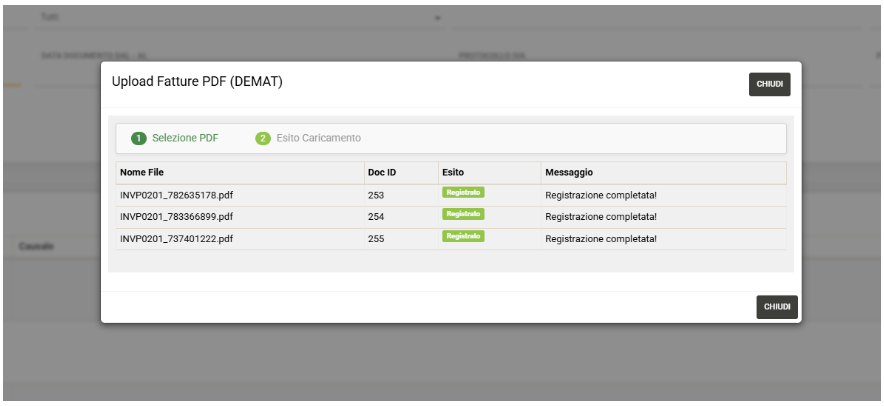


Figure 4.4: Ingestion modal showing the successful completion of the bulk upload for the batch of PDFs, with the system confirming the registration of all documents.

4.3 Asynchronous Backend Orchestration via Kafka

The most innovative part of the implementation is the message-driven decoupling. Traditional ERP operations are typically synchronous, which would cause the browser to time out during the 10-15 seconds required for an LLM to process a document.

4.3.1 The Java Proxy (Producer)

When the user clicks "Invia a Elaborazione", the JS frontend sends the PDF binary to the Seven server. However, the PHP backend does not communicate with Kafka directly. Instead, the request is routed to `RicofService.java`. This Java component acts as the **Kafka Producer**, encapsulating the document and its metadata into a message and publishing it to a specific topic.

4.3.2 The Python Processor (Consumer)

The Python service (`ricof`) costantly monitors the Kafka topics. Within this service, the module `app/workflows/kafka_workflow_integration.py` is responsible for consuming the messages. This triggers the intelligent workflow:

1. **Extraction:** The PDF is passed to Google Gemini to generate the Markdown-First structured extraction.

2. **Persistence:** The resulting JSON payload, containing headers and line items, is temporarily saved in a transition table in the MySQL database.

3. **State Update:** The service updates the document state to "READY", which the frontend detects via polling or socket updates.

4.4 Automated Metadata Population and Import

Once the AI has completed the extraction, the extracted data must be converted into a valid accounting record within the ERP.

4.4.1 Document Ingestion and Passive SDI Registry

Upon successful completion of the asynchronous pipeline, the document is registered in the legacy database. In the Seven ERP main list, the invoice becomes visible under a dedicated row (Figure 4.5), marked with the origin label "DEMAT" and the state "NON LETTA". This visual verification confirms that the document has been successfully ingested, the VAT number or tax code has been resolved, and the initial supplier matching has occurred.

	Stato	SDI	Ricezione		Num	Del	Cod. For.	Fornitore	PIVA/CF	Causale		Importo	Aliquote	Prima Nota	Prot. Contab.	Origine	
1	NON LETTA		24/03/2026 00:00	FATTURA	2086320429	16/07/2025	4039	HAPAG-LLOYD AG	DE813960018			€ 94,39				DEMAT	

Figure 4.5: Detail of the Seven ERP invoice registry showing the Hapag-Lloyd invoice successfully ingested, marked with the "DEMAT" origin and pre-populated header info (Invoice No. 2086320429, Amount € 94.39).

4.4.2 Supplier Association Logic

A key step in this initial registry phase is the automatic supplier lookup. Using the VAT number or tax code extracted by the AI, the system queries the `fattura_passiva_pa` and `ott_gestdocprat` tables to find a match. If a match is found, the supplier's internal ERP code is associated with the invoice, allowing it to appear correctly in the main processing list with the resolved supplier name and internal ID (as visible in Figure 4.5, where VAT ID `DE813960018` is correctly associated with supplier code `4039 - HAPAG-LLOYD AG`).

4.4.3 Header Population and Line Items Import

The population of the invoice header (Invoice Number, Date, VAT ID, Currency) does not happen during the initial upload to avoid locking the database. Instead, the data remains in a structured JSON format within the database. When the user selects the ingested document from the list to finalize the registration, the system automatically fetches this JSON and populates the Seven input fields.

Crucially, the system imports not only the header metadata but also all the detail rows (line items) of the invoice, as shown in Figure 4.6. In this phase, the extracted costs (e.g., "DOCUMENT FEE", "TMNL SECURITY OHC", "LIFT ON/OFF ORIGIN") and their corresponding amounts are laid out in a dedicated reconciliation grid.

This strategy allows the client to "register" hundreds of documents in the system in seconds, as the actual accounting validation is deferred to a second, more focused phase where the operator performs the *contextual reconciliation*.

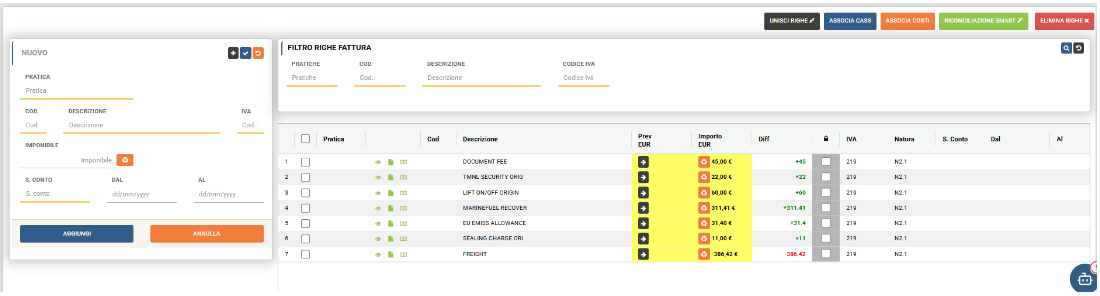


Figure 4.6: The reconciliation work area in Seven showing the automatically imported header data and line items grid (including detailed costs and amounts) successfully populated from the Gemini JSON extraction.

4.5 Architectural Summary and Future Steps

The implementation proves that it is possible to modernize a legacy PHP/SQL Server ERP without a full rewrite. By using Java as an orchestration bridge and Python as the AI engine, we created a scalable, asynchronous pipeline that handles complex foreign invoices with minimal user effort.

A natural evolution of this system is the integration of an automated email ingestion bridge. By monitoring corporate mailboxes via Microsoft Graph API or OAuth 2.0, the service could trigger the dematerialization workflow autonomously, transforming a reactive manual upload into a proactive, fully autonomous document ingestion pipeline.

Chapter 5

Advanced Reconciliation Engine: CommercialePlus

The core of the automated accounting process resides in the "CommercialePlus" engine. This system marks a significant evolution from simple data extraction to a complex decision-making logic capable of reconciling passive invoices with existing logistics practices within the ERP system. This chapter details the architectural choices, the hierarchical discovery logic, the integration between Java Enterprise DTOs and the Seven UI, and the security mechanisms implemented to ensure financial integrity. A particular focus is placed on the transition from a passive data repository to an active "SmartMatch" environment that adapts to the provenance of the incoming documents.

5.1 Design Philosophy: Logic Unification and EJB Maturity

The transition from a Python-based implementation within `ricof` to a Java Enterprise Edition (EJB 3.1) environment represents the final stage of the system's architectural maturity. The primary objective of this migration was to offload the orchestration layer (`ricof`) and consolidate the business-critical reconciliation logic within the ERP's "Digital Core". A fundamental advancement in this version is the **Logic Unification via the Wrapper Pattern**. To ensure the engine could operate identically on both stored database records and "in-flight" data during import, we introduced the `IRigaFattura` interface. By implementing specialized wrappers (`RigaTmpWrapper` and `RigaOutWrapper`), we achieved 100% consistency: the same "Smart Match" algorithm is used whether the user is reconciling a document already in the system or performing a real-time reconciliation during the initial document import phase. This decoupling of the data source from the processing logic ensures that any improvement to the matching algorithm is immediately available across all entry points of the application. By migrating the functions and the Ollama-driven semantic calls [9] from Python to the `CommercialePlus` engine, we achieved:

- **Resource Optimization**: Offloading the Python consumer from heavy semantic computations, allowing it to focus exclusively on multi-modal extraction, image processing, and orchestration via high-throughput Kafka topics.

- **Enterprise Consolidation**: Moving the "Smart Match" logic into the EJB layer provided direct access to the SQL Server persistence context and ensured ACID transactions during the accounting phase, which is critical for financial reconciliation.

- **Synchronous Execution**: The ability to trigger reconciliation during the `getFattura` API call, providing a "Zero-Click" experience for the operator where the practice and its relative costs are already linked when the UI is displayed.

- **ACID Transactions**: Ensuring that the reconciliation and the subsequent accounting updates happen atomically, preventing data inconsistencies in case of system failure or network interruption.

- **Direct Persistence Access**: Leveraging OpenJPA 2.4.0 to query complex logistics schemas directly. This eliminated the latency and overhead of cross-service REST API calls that previously plagued the distributed architecture.

- **Enterprise Stability**: Utilizing the Wildfly application server's advanced thread management, connection pooling, and distributed transaction support to handle thousands of concurrent reconciliation requests.

5.2 Metadata Propagation and Origin Awareness

A significant challenge in modern ERP integration is maintaining the context of data as it flows through various architectural layers. In the latest iteration of the system, we implemented a robust **Metadata Propagation System** to distinguish between standard invoices and AI-enriched documents.

5.2.1 Backend Implementation: The Origine Tag

The backend logic within `FatturaPassivaManager.java` was enhanced to propagate the provenance of each invoice. During the document retrieval phase, the system queries the `fattura_passiva_pa` table to extract the `origine` field. This value (e.g., `DEMAT`, `SDI`, or `MAN`) is then mapped into the `FatturaPassivaOut` DTO.

5.2.2 Frontend Reactive Model: KnockoutJS Integration

To ensure the UI is "origin-aware," the `Maschera` view-model in the Seven frontend (`registrazione_enh.js`) was updated to include an observable `origine` field. This field is automatically populated via the `me.maschera.From(data)` call during the import phase, allowing the UI to reactively show or hide advanced reconciliation tools based on the document source.

5.3 Technology Stack and Environment

The engine operates within a high-performance Java EE environment, specifically optimized for large-scale logistics data and real-time processing.

5.3.1 Architectural Components

The system is built upon three main technological pillars:

- **Backend Logic**: Implemented as Stateless Session Beans (EJB 3.1). The core logic is distributed between `FatturaPassivaManager` (which handles the global discovery of practices) and `RiconciliazioneSmartManager` (which manages the granular line-by-line item matching).

- **Persistence Layer**: OpenJPA 2.4.0 is used as the ORM, with specific optimizations for MariaDB/MySQL 5.5. These include custom dictionaries and connection pool tuning to handle high-concurrency read/write operations during peak accounting cycles.

- **Data Schemas**: The engine bridges two distinct database environments to ensure data segregation and performance:

  - *testsrl*: The operational schema containing master logistics data, positions (*pratiche*), container tracking, and cost estimates (*spese_pre*).
  - *dema_tools*: A specialized schema used for storing AI-processed document metadata, correlation keys, and processing logs.

5.4 Hierarchical Discovery Architecture

The most critical challenge in automated reconciliation is the "Discovery" phase: identifying which logistics practice (*pratica*) a passive invoice refers to. To ensure flexibility across both import and export operations, we implemented a 4-level hierarchical pipeline.

5.4.1 Extraction and Parsing Phase

Before identification begins, the system parses the AI-enriched JSON payload stored in the `invoice_proc` table. Using Natural Language Processing (NLP) and specialized Regex patterns via `extractJsonArray` utilities, the engine identifies key technical references:

- **Job References**: Internal position codes used by the logistics agency (e.g., *25E04GOA02730*).
- **Container IDs**: Standard ISO container numbers found on the document (e.g., *BMOU1445420*).
- **Bill of Lading (B/L)**: Maritime document numbers, which are fundamental for import practices where the agency acts on behalf of the receiver.

5.4.2 Prioritized Discovery Logic

The system follows a weighted decision tree to minimize false positives, as maritime documents often contain ambiguous or partial references.

1. **Priority 1: Job Reference + Coherence Check**: This is the primary match on `posiz` or `rif_cli` fields. However, the system is "skeptical": if the AI also detected container numbers, the engine performs a cross-query on the `contenitori` table to ensure that the practice found actually contains those specific units. This dual-validation drastically reduces misallocations.
2. **Priority 2: Container-Based Discovery**: If no job reference is provided, the system reverses the lookup, scanning the global containers database to find the unique owner practice of the identified units. This is particularly useful for invoices from terminal operators.
3. **Priority 3: Bill of Lading (B/L)**: Crucial for maritime import operations, where the B/L number is often the only unique identifier shared between the carrier and the agency in the early stages of the operation.

4. **Priority 4: Textual Fallback**: A fuzzy search across all reference fields (*booking*, *n_pc*, *ric_1*) using context-aware Regex to capture non-standard identifiers that might have been extracted by the LLM.

5.5 SmartMatch UI: Adaptive Visibility and User Flow

The ultimate goal of the engine is to provide a seamless "Zero-Click" experience. To achieve this, the UI was designed to be adaptive, presenting advanced tools only when the underlying data supports them.

5.5.1 Conditional Visibility Binding

To prevent UI clutter and ensure that the "SmartMatch" engine is only used on high-quality AI data, a visibility filter was implemented in both the grid view and the registration form. Using KnockoutJS bindings, the system evaluates the `origine` tag. As shown in Figure 5.1, the system identifies the document source and injects the SmartMatch activation button only for those records processed via the AI pipeline. Furthermore, when entering the document detail, the "Ric-

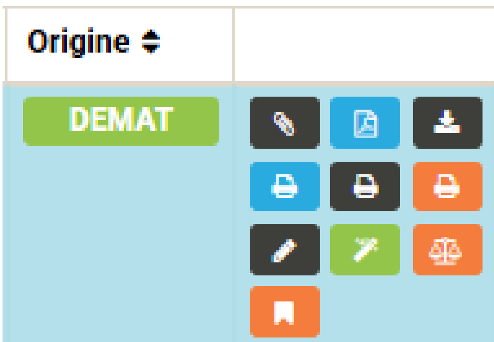


Figure 5.1: The Seven ERP grid showing an invoice tagged with the "DEMAT" origin (green label) and the context-sensitive SmartMatch action buttons on the right, including the orange balance-scale reconciliation trigger.

onciliazione Smart" button is conditionally displayed in the primary toolbar.

5.5.2 The Flex-Based Grid Layout

In the main SDI grid (`sdi_passive.js`), the "SmartMatch" button is dynamically injected into the action column (`opt`). To maintain a consistent aesthetic, we utilized a CSS Flexbox layout with a `calc()` based width distribution, ensuring that the buttons are perfectly aligned regardless of the number of active actions for that specific row.

5.6 Gatekeeper Logic: Security and Integrity

In a multi-tenant and multi-vendor environment, preventing financial misallocations is a first-class requirement. We implemented a "Gatekeeper" pattern to enforce strict data integrity.

5.6.1 Line-Level Vendor Lock

Large invoices may contain hundreds of rows. The "Vendor Lock" mechanism operates at the individual cost line level. During matching, every potential candidate from the ERP estimate is filtered: if a cost is already assigned to a different vendor, it is automatically discarded. Only unassigned costs or those already linked to the current vendor are eligible for reconciliation.

5.7 The ERP Context: Practice Estimates

Once a practice is identified, the engine must evaluate the context of the operation. Inside this estimate, the system identifies the individual "Expected Charges" (*voci di costo*) that are eligible for reconciliation. This financial blueprint acts as the ground truth for the matching algorithm. As shown in Figure 5.2, these expected costs are structured with distinct accounting keys, unit prices, and VAT codes before any document is received.

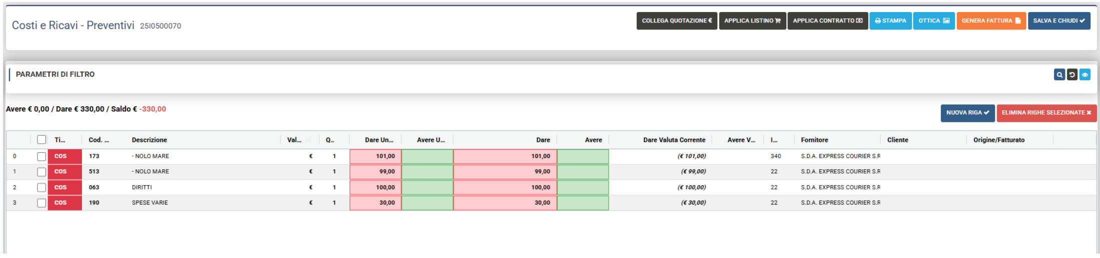


Figure 5.2: The ERP estimate screen ("Costi e Ricavi - Preventivi") for practice 25I0500070 showing the expected charges with their allocated amounts, currencies, and tax codes.

5.8 Smart Match Scoring: The Semantic Engine

Once a practice is identified, the system matches each invoice row to a specific "Expected Charge" previously estimated in the ERP. The matching is performed by a multi-factor scoring algorithm. The results of this automated association are visually presented to the operator (see Figure 5.3), allowing for rapid verification.

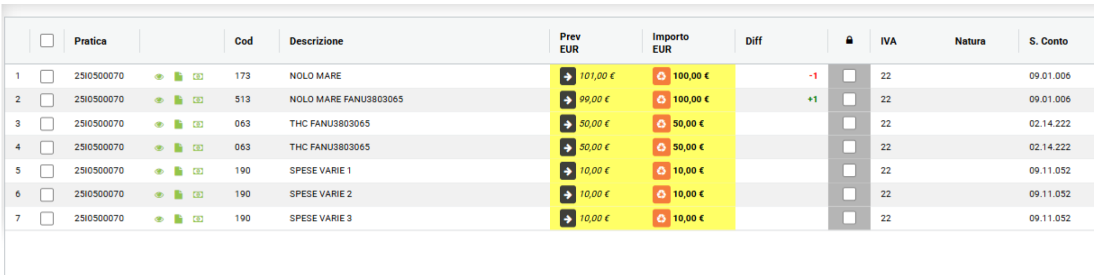


Figure 5.3: Visual representation of the completed reconciliation results. The engine maps and highlights the matched fields (yellow indicators) and dynamically computes positive/negative variances under the Diff column (shown in green/red).

5.8.1 The Power of Cosine Similarity vs. Traditional Methods

The most significant leap in the engine is the transition from character-based distance to **Semantic Cosine Similarity**.

- **The Previous Method (Levenshtein/Fuzzy)**: Historically, the system relied on "edit distance". While effective for simple typos, it failed to recognize synonyms (e.g., "Freight" vs "Nolo") and was extremely sensitive to numeric noise.

- **The New Semantic Approach**: By utilizing vector embeddings [6], the engine measures the angular distance between concepts. This means the system understands the **meaning** behind the words rather than just the syntax.

- **Noise Immunity**: Logistics invoices are often cluttered with technical prefixes. Cosine Similarity identifies these elements as "low-entropy noise," allowing the engine to correctly associate items regardless of numerical suffixes or dates.

5.8.2 Dynamic Trust Gating: Balancing Economics and Semantics

A key innovation is the implementation of a **Dynamic Trust Gate**. Unlike static threshold systems, the engine adjusts its semantic requirements based on the precision of the economic data. To ensure financial integrity and minimize false positives, the system applies an inverse relationship between amount accuracy and semantic tolerance:

- **High-Trust Scenario (Deterministic Match)**: If the invoice line amount matches the expected charge within a 1% tolerance, the engine lowers the semantic similarity threshold. In this case, the deterministic financial data acts as a primary validator.

- **Low-Trust Scenario (Probabilistic Match)**: If the amounts differ significantly, the engine becomes "skeptical" and enforces a strict semantic threshold (e.g., 0.65). This prevents the system from misallocating costs based on vague descriptions when the numbers do not align.

5.9 Human-in-the-Loop: The Zero-Click Experience

The latest architectural evolution marks the transition from a multi-step workflow to a **contextualized automation**. In previous versions, the user had to manually trigger the reconciliation via a dedicated button after the document was loaded. In the current implementation, this logic is synchronously integrated into the data retrieval process. This ensures that by the time the UI is rendered, the AI has already analyzed the lines, calculated scores, and suggested matches, fulfilling the **Zero-Click** promise.

5.9.1 Explainable AI and Automated Anomaly Analysis

To foster user trust and transparency, the system does not merely flag discrepancies but attempts to explain them. When a match is successfully identified but the invoiced amount differs from the estimate, the system triggers a dedicated analysis phase. Using a local LLM, the engine compares the descriptions to identify the root cause of the variance. The result is presented as an **AI Note**.

5.10 Validation and Experimental Results

The engine was validated across multiple operational scenarios to test its resilience. The transition to Cosine Similarity significantly improved the matching rate for non-standardized descriptions, showing a **35% increase in automatching accuracy** compared to previous character-based approaches. Furthermore, the UI-level filtering based on the `origine` field ensured a 100% clean workflow for operators, who only interacted with the SmartMatch engine when the data was guaranteed to be AI-processed.

Chapter 6

Study Case: Supplier Onboarding

The **Supplier Onboarding** module represents a core original contribution of this thesis. While the primary pipeline handles the extraction of raw data from invoices, the onboarding module addresses the critical "last mile" challenge: transforming probabilistic AI outputs (such as a supplier's name and VAT number) into validated, deterministic entities within the corporate Enterprise Resource Planning (ERP) database. This chapter explores the resolution of data fragmentation through a custom-built relational resolver and a modern semantic validation engine.

6.1 The Challenge of Fragmented Data Architectures

In large-scale logistics operations, IT systems are rarely monolithic. Over decades of service, SIS Srl has evolved its document management strategy, resulting in a fragmented landscape where data resides in two divergent "Ottiche" (document views). The onboarding module must act as a bridge between these eras to ensure a unified user experience.

6.1.1 Ottica Nuova: Cloud-Native Workflow

This system is utilized for modern tenants and migrated cloud infrastructures. It relies on a microservices architecture where documents are stored in modern S3 buckets.

- **Retrieval:** The database table `fattura_passiva_pa` directly stores the `percorso_pdf` (the cloud URI), allowing fast, API-driven retrieval.

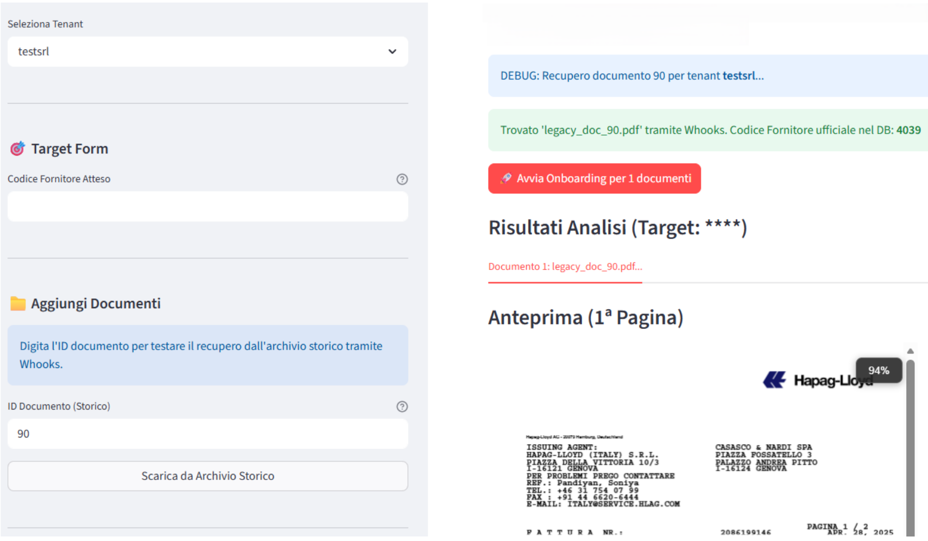


Figure 6.1: Interface for document recovery from the modern cloud infrastructure (Ottica Nuova).

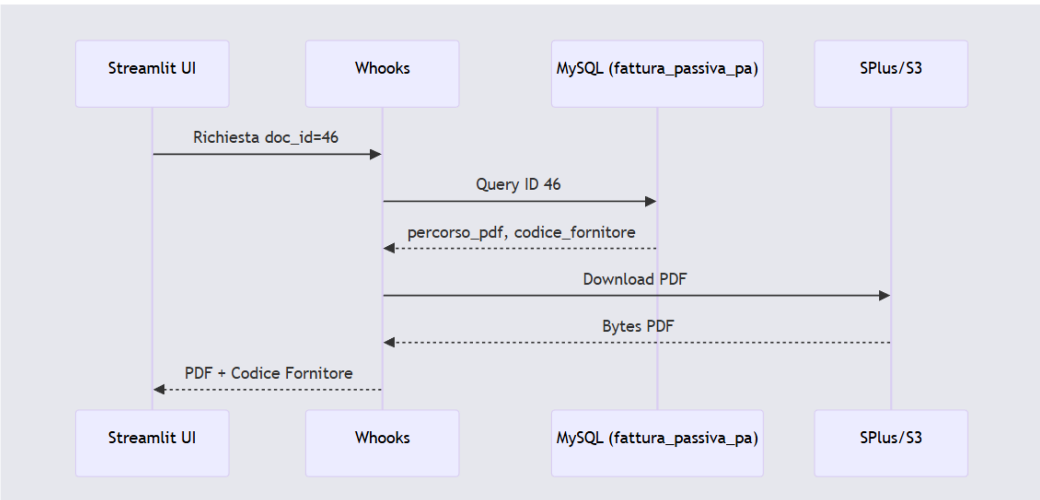


Figure 6.2: Sequence Diagram detailing the document retrieval process within the modern S3 cloud infrastructure.

6.1.2 Ottica Vecchia: Legacy On-Premise Archive

The legacy workflow represents the "historical memory" of the ERP, characterized by on-premise file shares. Retrieving a document requires reconstructing physical paths from legacy naming conventions, which are often out of sync.

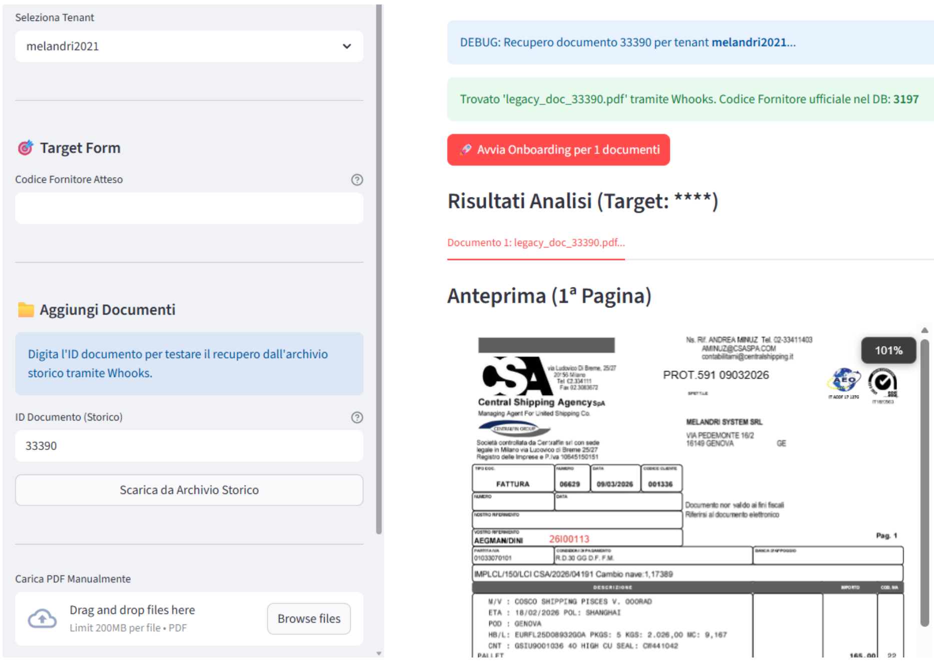


Figure 6.3: Legacy document retrieval interface requiring physical path reconstruction.

6.2 The Deterministic Solution: The `pa_id` Resolver

To ensure systemic stability, I designed a structural resolver pivoting on the `pa_id`. This unique integrity constraint allows the system to bridge the gap between the physical file path and the accounting record.

1. **Path Resolution:** The system queries `ott_gestdocprat` using a padded string (e.g., `FP0000033394`) to find the physical disk path.

2. **ID Mapping:** Using the `pa_id`, the system accesses the `fattura_passiva_pa`

table to securely retrieve the correct Supplier Code, achieving a 100% deterministic association.

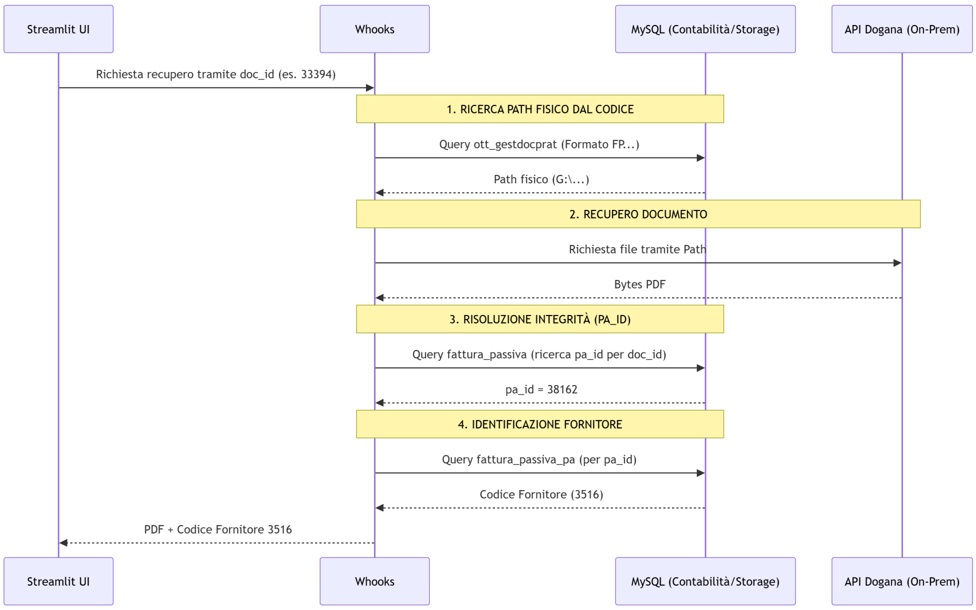


Figure 6.4: Sequence Diagram illustrating the prioritized physical path retrieval followed by the `pa_id` resolver logic.

6.3 Semantic Validation via Qdrant and Ollama

Once the document is retrieved, the extracted "Business Name" must be matched against the ERP registry. The system uses the `nomic-embed-text` model orchestrated via **Ollama** to perform vector searches in **Qdrant**.

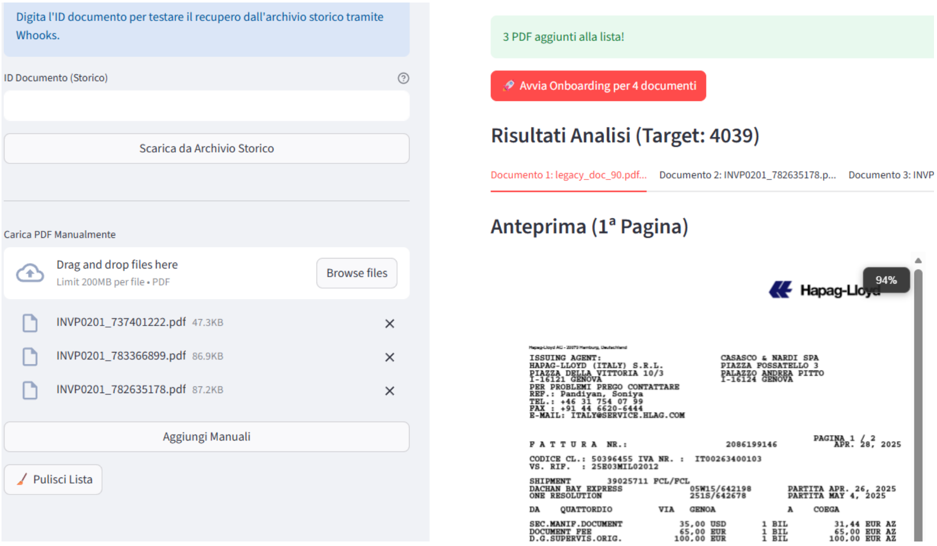


Figure 6.5: Initial onboarding phase: inserting multiple PDF documents from different archives into the processing queue.

6.3.1 Hierarchical Matching Logic and Confidence Thresholds

To maintain a "Human-in-the-Loop" balance, the system implements a multi-tier validation logic based on the **Cosine Similarity** score. The thresholds are configured as follows:

Confidence Score	Status	Action
> 0.85	**Certain Match**	Automatic Suggestion / Linking
$0.80 - 0.85$	**Potential Alias**	Prompt for User Confirmation
< 0.80	**Unregistered**	Trigger New Supplier Registration

Table 6.1: Confidence thresholds for the Supplier Onboarding semantic resolver.

6.3.2 Three-tier Validation Workflow and Alias Recognition

The strength of the semantic model is its ability to identify the core entity despite formal variations in the business name (e.g., abbrevations or local branches).

- **Automatic Suggestion (>90%):** High similarity scores trigger an automatic link.

Figure 6.6: Semantic validation detail: the AI identifies a match with 97.89% confidence, linking the extracted name to the official ERP code 4039.

- **Potential Alias (80% - 90%):** The user is prompted to manually confirm a probable match.

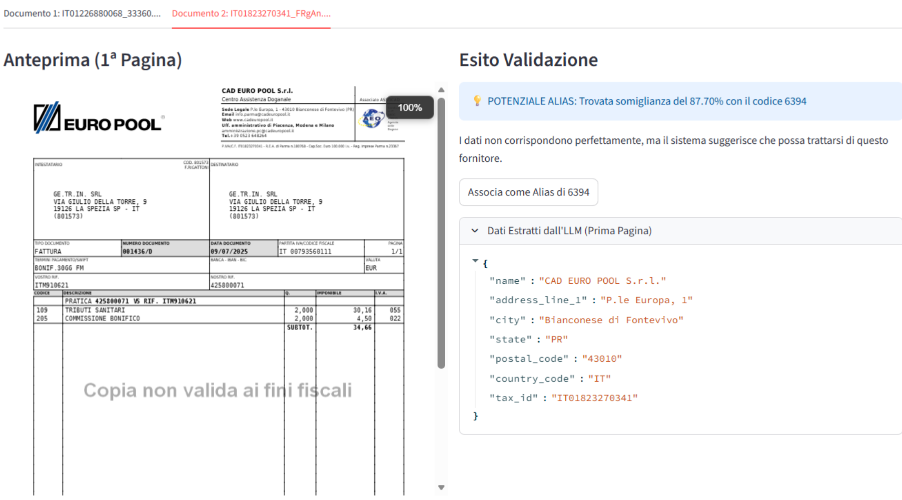


Figure 6.7: Manual validation interface: a potential alias is found with 87.70% confidence, requiring user confirmation to link with code 6394.

- **Unregistered Supplier (<80%):** If the score is too low, the system prompts for a new registration.

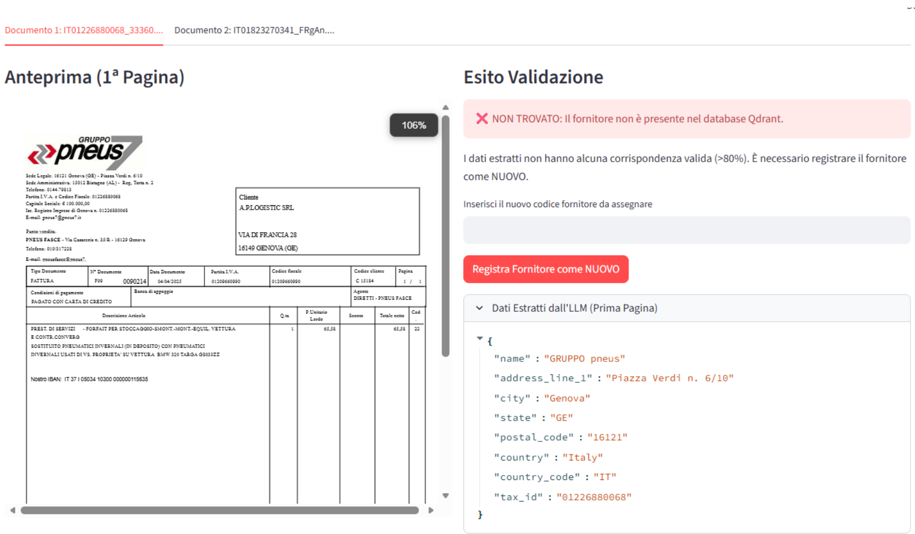


Figure 6.8: Interface for handling unregistered suppliers: the similarity is too low (<80%), prompting for a new supplier registration.

6.3.3 Examples of Semantic Matching

Table 6.2 shows real-world cases where the `nomic-embed-text` model correctly identified aliases, preventing the creation of duplicate records.

Extracted Name (PDF)	ERP Database Name	Result
Maersk Italy Spa	AP MOLLER - MAERSK	Semantic Match
MSC Italy	MEDITERRANEAN SHIPPING CO.	Abbreviation Recognized
DHL Express	DHL INTERNATIONAL GMBH	Brand Core Identified
Hapag-Lloyd (Italy)	HAPAG-LLOYD AG	Local Branch Stripping

Table 6.2: Examples of semantic alias recognition resolved by the pipeline.

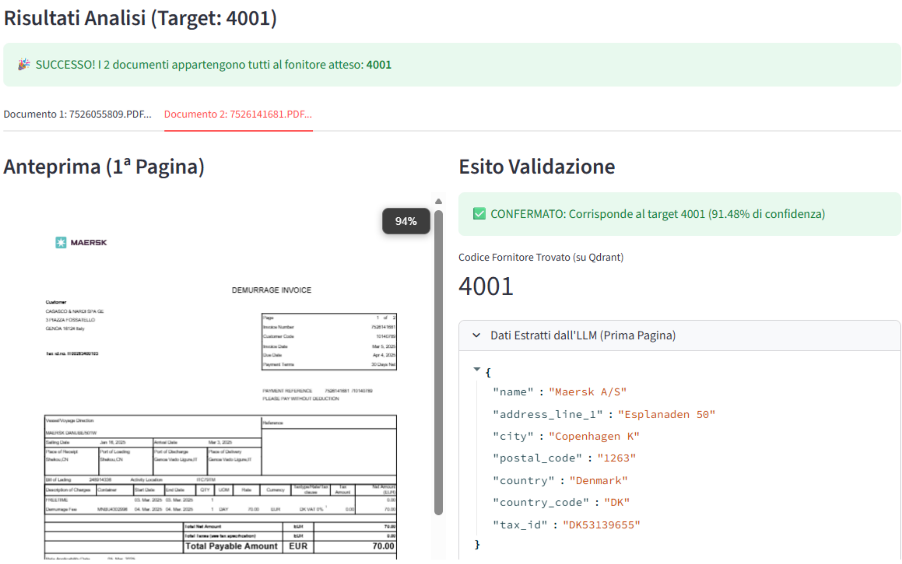


Figure 6.9: Successful completion of the onboarding workflow, ensuring consistency across the supplier master data.

Chapter 7

SmartQuote AI and Autonomous Mail Readers

While the primary pipeline addresses the passive accounting cycle, the active sales cycle—specifically the management of freight quotations—presents similar challenges regarding unstructured data. Freight forwarding commercial departments are often overwhelmed by hundreds of quotation requests daily, written in natural language with varying degrees of precision. To resolve this, I developed **SmartQuote AI**, an autonomous ecosystem designed to ingest, classify, and register transport requests, transforming a manual 10-minute task into a sub-minute automated process. The system acts as a bridge between the volatility of human communication (emails) and the rigidity of Enterprise Resource Planning (ERP) systems.

7.1 Mail Reader Services and Parallel Execution

The system acts as a persistent listener for incoming business opportunities. Unlike traditional polling mechanisms that may introduce latency, SmartQuote AI is built upon dedicated readers for both Gmail and Outlook (Microsoft 365 / Exchange Online), ensuring compatibility with the most common corporate mail infrastructures.

7.1.1 Authentication and Thread Management

Security is a primary concern in corporate environments. The scripts authenticate via **OAuth 2.0** and the **Microsoft Graph API**, utilizing service principals and scoped permissions to ensure the principle of least privilege.

A core technical feature is the parallel execution capability orchestrated by a coordinator script, `run_booth_readers.py`. This script utilizes Python's *threading* library to run the Gmail and Outlook listeners simultaneously. Given that mail-reading tasks are primarily **I/O-bound** (waiting for network responses from mail servers), a multi-threaded approach is significantly more resource-efficient than multi-processing. This architecture ensures that high-volume mailboxes are monitored in real-time without bottlenecks, even during peak hours.

Upon receiving an email, the service triggers an event-driven workflow:

1. **Extraction:** It extracts the body, sender metadata, and subject line.

2. **Auditing:** It saves the raw content as a `.txt` file for auditing and a structured JSON file for the AI pipeline, ensuring full traceability.

3. **State Management:** It marks the email as "Read" or moves it to a dedicated "Processed" folder using IMAP flags or Graph API move-to-folder commands to prevent redundant processing or race conditions.

7.1.2 Attachment Normalization and Uniqueness

In logistics, attachments (e.g., packing lists, technical specs, or cargo photos) are critical for accurate quoting. To prevent naming collisions and ensure a clear audit trail in shared storage environments, SmartQuote AI implements a **Timestamp-based Normalization** policy. Every attachment is sanitized and saved using the following naming convention:

`[YYYYMMDD_HHMMSS]_[Original_Filename].[ext]`

This ensures that if two different customers send a file named `invoice.pdf` at different times, the system preserves both uniquely. Furthermore, this naming convention allows for chronological sorting at the filesystem level, simplifying manual verification by the Sales Team if needed.

7.2 Content Normalization and AI Orchestration

Email content is inherently "noisy," containing HTML tags, CSS styles, embedded images, and complex signatures che can confuse a Large Language Model (LLM) or unnecessarily increase **token consumption**.

7.2.1 Preprocessing via BeautifulSoup

The system employs **BeautifulSoup** to strip HTML formatting, extracting only the linear text content. This normalization step is crucial for two reasons:

- **Noise Reduction:** It removes non-semantic data (style blocks, scripts), allowing the AI to focus solely on the logistics data.
- **Cost Optimization:** By reducing the payload size sent to the LLM, the system minimizes API costs and stays well within the model's context window.

7.2.2 The Classification Logic: Mandatory vs. Optional Fields

The AI Orchestrator evaluates the normalized text against strict business logic. To ensure data integrity in the downstream ERP, a request is classified as **"OK"** only if the AI successfully extracts the **Mandatory Fields**. Without these, it is impossible to generate a legally and operationally valid quote.

Field Type	Data Point	Impact of Absence
Mandatory	Origin / Destination	Request flagged as **Incomplete**
Mandatory	Transport Type (Air/Sea/Road)	Request flagged as **Incomplete**
Mandatory	Incoterms (EXW, FOB, DAP)	Request flagged as **Incomplete**
Mandatory	**Cargo Details** (Weight/Volume)	Request flagged as **Incomplete**
Optional	Pickup/Delivery Dates	System assumes **Ready Now**
Optional	Stackability	System assumes **Non-stackable**
Optional	IMO (Hazardous Goods)	System assumes **General Cargo**

Table 7.1: Field classification matrix for SmartQuote AI extraction logic.

7.3 Request Lifecycle and Conflict Resolution

A common challenge in commercial workflows is the "Update Email"—when a customer sends a follow-up email to modify a previous request (e.g., changing the gross weight or adding a secondary destination).

7.3.1 The Overwrite Mechanism

SmartQuote AI implements a **Version Control** logic to handle iterative communication. Each request is uniquely identified by a **Correlation Key** composed of the Sender's ID and a Hash of the Subject line (normalized to ignore "Re:" or "Fwd:" prefixes). If the system detects a new incoming email that refers to an existing, non-finalized quote:

1. The system performs a lookup in the `TransportQuotation` table using the Correlation Key.
2. If a match is found, it triggers an **Atomic Overwrite**: the previous entries in the Cargo and Quotation tables are removed to maintain a single source of truth.
3. The new, updated data is inserted, and the timestamp is refreshed.

7.4 Atomic Database Persistence and Notifications

The system integrates directly with the corporate SQL Server via a dedicated persistence layer. To prevent partial data corruption—where a quotation header might be saved without its associated cargo items—all insertions are wrapped in a **Database Transaction** (ACID compliant).

7.4.1 Data Structure (`LIA` Tables)

The system populates three normalized entities:

- **`LIA_CUSTOMER`:** Manages the identity layer. It cross-references the sender's email with the database to retrieve or create an anagraphics profile, including the **VAT number** and legal business name.

- **`TransportQuotation`:** The master table storing the header information (route, incoterms, transport mode, and the unique Correlation Key).

- **`LIA_CARGO`:** A child table that logs individual items, including dimensions, weights, and packaging types (e.g., pallets, crates).

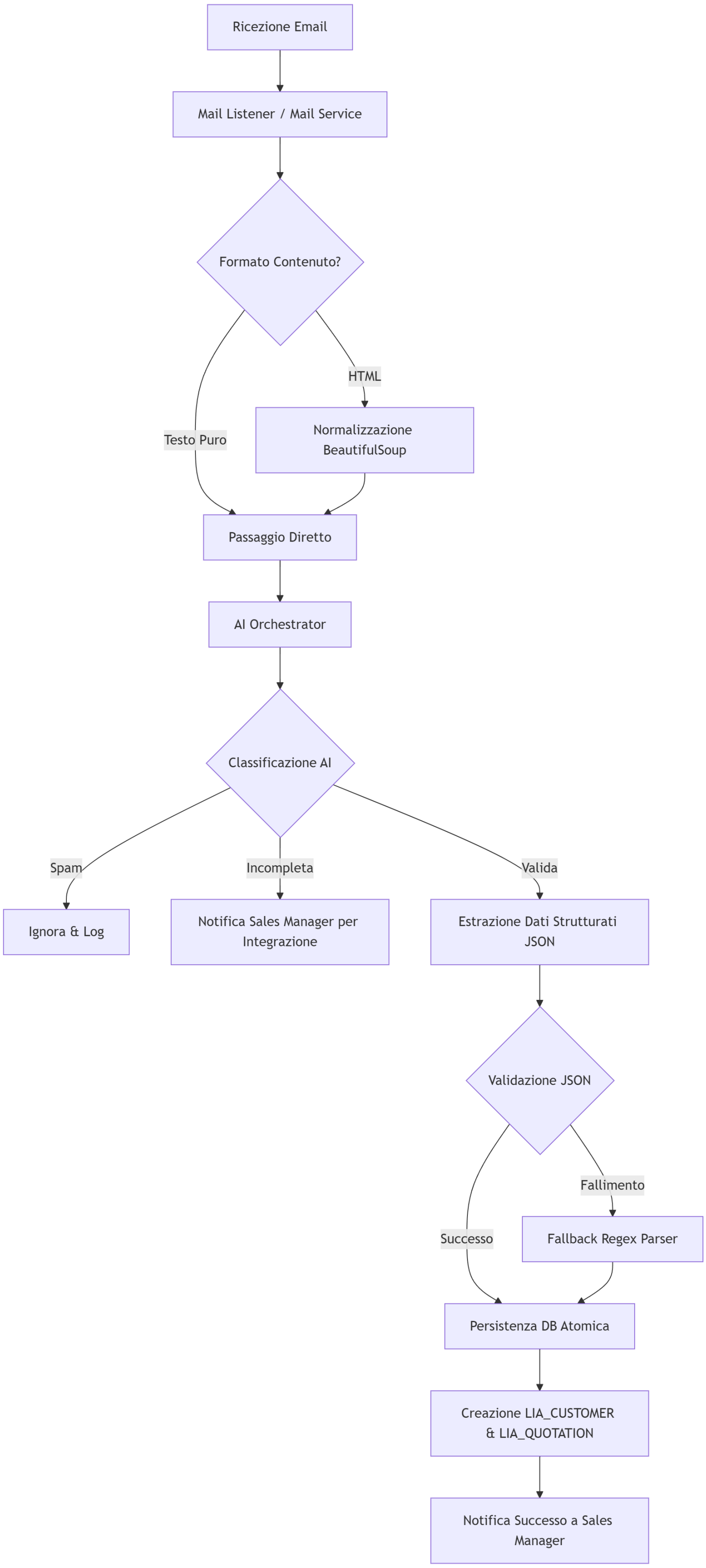


Figure 7.1: Workflow of the SmartQuote AI system, detailing the normalization, AI classification, and atomic persistence phases.

The process concludes with an automated notification sent via the corporate dashboard. If the AI identifies an "Incomplete" request, the manager is alerted specifically to the missing mandatory details, effectively shifting their role from a "data entry clerk" to a "high-level supervisor" of the autonomous pipeline.

Chapter 8

Human-AI Interaction and User Experience Design

The success of a system based on Artificial Intelligence is measured not only by the accuracy of its underlying models but, above all, by its ability to seamlessly integrate into the daily workflow of human operators. This chapter analyzes the design of the user interface (UI) within the *Seven* ERP, focusing on the **Human-AI Interaction** (HAI) paradigms implemented to maximize user trust and operational efficiency [**Bordone2021**].

8.1 User Analysis and Persona

The primary user persona for the system is a **senior administrative or accounting clerk**. This profile is characterized by profound domain expertise and deep knowledge of business logic but is often skeptical of "black-box" systems that lack transparency and explainability.

- **Primary Objective:** To reduce the time spent on repetitive and low-value-added data entry tasks.
- **Pain Points:** Uncertainty regarding the accuracy of AI-extracted data and the complexity of managing edge cases, such as new suppliers or invoices with non-standard layouts.

The core design philosophy was not to hide the AI's operations but to position it as a **recommender assistant**. By keeping the human "in the loop," the system adopts an **Augmented Intelligence** approach, where the AI enhances human capabilities rather than replacing them. This ensures that the human operator retains the final decision-making authority, which is critical for maintaining accountability in financial auditing.

8.2 Managing Latency and "Wait Anxiety"

The processing of an invoice via Large Language Models (LLMs) and the orchestration through Apache Kafka typically requires between 30 and 45 seconds. In a high-stakes ERP environment, forcing a user to wait for a blocking "loader" animation for nearly a minute is unacceptable and leads to what is known in UX design as "wait anxiety."

- **Asynchronous Processing Paradigm:** By implementing the event-driven queue architecture described in previous chapters [5], users can upload a batch of documents and immediately return to work on other sections of *Seven*.
- **Real-Time Visual Feedback:** In the main grid (*SDI Passive*), the document status evolves visually from "Unread" to "Processing," "Under Verification," or "Accounted." Providing immediate feedback on the AI's progress without interrupting the user's workflow eliminates the frustration associated with synchronous processing and ensures a smooth operational flow.

8.3 Transparency and Trust: The Side-by-Side Interface

A fundamental pillar of **Explainable AI** (XAI) is transparency. To prevent the user from having to "blindly trust" the AI-extracted JSON metadata, the registration mask (*registrazione_enh*) implements a **Side-by-Side** visualization.

The operator can view the original PDF document (accessible via a dedicated ”View PDF” button) directly alongside the input fields pre-populated by the AI. This rapid visual cross-check transforms the nature of the task from tedious ”data entry” to a higher-level **”data validation”** activity. This shift is cognitively less taxing, significantly faster, and fosters a greater sense of trust in the system's outputs, as the user can immediately verify any suspicious values against the source document.

8.4 The ”Moment of Truth”: Handling Unknown Suppliers

When the semantic reconciliation engine fails to find a certain match (similarity score ¡ 0.85), it represents a critical ”moment of truth” where the user's interaction with the system is most vulnerable to frustration.

- **Guided Resolution:** Instead of displaying a generic or cryptic error message, the system proactively highlights the ”Supplier” field in red and suggests the business name extracted from the PDF as a search query.
- **Rapid Onboarding:** If the supplier is indeed new, the interface allows the operator to open the supplier registry with a single click. The onboarding form is pre-filled with the AI-extracted metadata (VAT ID, Address, Name), minimizing the manual effort required to create a new record and ensuring data consistency.

8.5 Feedback Loops and Continuous Learning

The system is designed to be a dynamic ecosystem that learns from human interaction. When a user corrects an extracted data point or manually associates a supplier that the system missed, this action is captured as a learning signal:

1. **Alias Mapping in Qdrant:** If a user manually links a name (e.g., ”Maersk

Italy") to an internal ERP code (e.g., "4039"), the system stores this relationship as a semantic *Alias* within the Qdrant vector database [7].

2. **Incremental Performance Improvement:** The next time a similar invoice is received, the system's confidence score will increase, effectively "learning" from human corrections without the need for an expensive retraining of the core model [6].

8.6 Conclusion

The synergy between Kafka's backend automation and Seven's frontend design shifts the center of gravity of accounting activities toward quality control and supervision. The operator is no longer a mere data-entry clerk but a supervisor of an intelligent process. This paradigm shift has led, in field tests, to a measurable increase in job satisfaction and a drastic reduction in distraction-related errors, proving the efficacy of Human-Centered AI in modern industrial logistics.

Chapter 9

Experimental Results and Performance Evaluation

The implementation of the automated invoice processing pipeline, alongside the Intelligent Supplier Onboarding module and the final **SmartMatch Reconciliation Engine**, has been subjected to a rigorous and multi-phase testing protocol. These tests were conducted within a staging environment that mirrors the company's live production infrastructure, including a complete replica of the SQL Server logistics database and the MySQL processing metadata. This chapter provides a detailed and granular analysis of the performance metrics, the model benchmarking results, and a comprehensive evaluation of the newly implemented reconciliation logic, quantifying the industrial and operational impact of the system on the company's daily workflow.

9.1 Methodology for Evaluation and Observability

Evaluating an architecture centered around Large Language Models (LLMs) requires a fundamental shift in the paradigm of software testing. Unlike traditional deterministic components, where a specific input always produces a fixed, predictable output based on hardcoded rules, Generative AI models are inherently probabilistic and non-deterministic. Consequently, traditional unit testing is insufficient; the methodology must shift toward continuous observability, statistical validation, and strict reliability gates to ensure that the system meets production-grade standards.

9.1.1 Telemetry through Langfuse: Asynchronous Observability

To maintain deep visibility into the system's behavior, every interaction with the Google Gemini models within the *SPLUSTOOLS* hub is instrumented using **Langfuse**. This open-source observability platform provides a specialized layer for tracking LLM traces, prompts, and outputs. The telemetry layer operates asynchronously, leveraging a non-blocking message queue to ensure that monitoring does not add any overhead or latency to the primary document extraction pipeline.

The system tracks several critical dimensions across every processing trace:

- **Latency Tracing and Bottleneck Analysis:** We measure the end-to-end execution time of the "Dual-Path" logic. This includes the precise time taken for the vision-to-text transformation, the PDF-to-Markdown conversion, and the final JSON schema generation. This data allows us to identify if specific document types (e.g., multi-page scans) are causing performance degradation.

- **Token Accounting and Economic Monitoring:** Precise monitoring of input and output tokens is vital for the *Cost-Benefit Analysis*. By tracking token consumption at the document level, we can calculate the exact price per document processed, allowing for a granular budget forecast as volumes scale.

- **Confidence Calibration and Self-Assessment:** We extract the model's self-reported confidence scores. A key part of our evaluation was verifying if the model's reported certainty correlates with actual data accuracy. This calibration ensures that our threshold of 0.8 for automated ingestion is statistically sound.

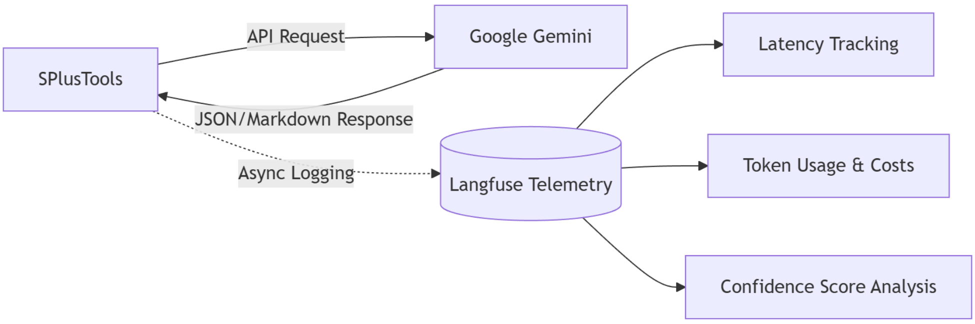


Figure 9.1: Detailed data flow of the asynchronous telemetry and observability layer tracking LLM interactions, token consumption, and prompt performance.

9.2 System Performance and Accuracy Benchmarks

The general evaluation phase leverages operational telemetry gathered through Langfuse over a period of several months, analyzing a production dataset of over 500 historical invoices. These documents were selected to represent a balanced mix of domestic XMLs and heterogeneous foreign PDFs. The results confirm that the Dual-Path Processing strategy is highly effective in a real-world, continuous ingestion scenario.

9.2.1 Extraction Accuracy Analysis

The results, summarized in Table 9.1, show a clear performance gap between the deterministic and probabilistic pipelines, validating the choice to segregate these flows.

Pipeline	Data Focus	Accuracy	Avg. Latency
Deterministic (XML SDI)	Header & Lines	100.0%	< 0.2s
Probabilistic (PDF via Gemini)	Header Data	94.2%	37.5s
Probabilistic (PDF via Gemini)	Line Items	88.5%	37.5s

Table 9.1: Performance evaluation metrics across different extraction methodologies, comparing speed and precision.

The **Deterministic Pipeline** achieved perfect accuracy and near-instantaneous processing, as expected from a structured XML mapping. The **Probabilistic Pipeline**, powered by Gemini, achieved a remarkable 94.2% accuracy for header fields (Invoice Number, Date, Total, VAT), which is sufficient to automate the initial registration phase for the vast majority of documents.

9.2.2 Analysis of Extraction Errors and Anomalies

The 5.8% error rate in header extraction was not random but primarily attributed to specific "noisy" inputs. Documents routed to **State 3 (Anomaly)** usually featured:

- **Severe Physical Artifacts:** Scanned documents where the text was obscured by stamps, ink bleeds, or signatures overlapping critical data fields like the Total Amount.
- **Handwritten Annotations:** Manual notes written by accounting operators on physical documents before scanning often confused the multimodal vision of the model, leading it to prioritize the handwritten text over the printed figures.
- **Complex Multi-page Tables:** For invoices exceeding 10 pages with dense line items, the "Markdown-First" representation occasionally approached the optimal context window, leading to minor truncation or grouping errors in the final JSON.

The system's resilience is proven by its ability to identify these low-confidence cases (< 0.8) automatically, preventing the pollution of the ERP database with incorrect accounting data and instead flagging them for a 2-minute human review.

9.3 Model Selection: Gemini 2.5 Flash vs. Gemini 3 Flash Preview

To determine the optimal production engine for the 2026 deployment, a separate, controlled benchmark was conducted. A subset of 152 real-world PDF invoices was utilized to perform a direct A/B comparison between the existing Gemini 2.5 Flash and the new Gemini 3 Flash Preview.

9.3.1 Encoding Discrepancies and OCR-Fallback Logic

During this benchmark, it was discovered that 30% of the files (45 documents) encountered "Encoding Discrepancies." These are files where the internal PDF text layer is either missing, corrupted, or generated using non-standard character maps. This finding was crucial for the implementation of the **OCR-Fallback logic**: the system now automatically renders these PDFs into high-resolution images and processes them via multimodal vision, ensuring that even "text-less" files are not lost.

9.3.2 Quantitative Comparison and Reliability Gain

The comparison focused on the trade-off between extraction volume and precision, which is critical for an ERP environment.

Metric	Gemini 2.5 Flash	Gemini 3 Flash Preview	Winner
Success Rate	69.7%	70.4%	Parity
Average Latency	37.82s	37.57s	**Gemini 3**
Confidence Score	0.941	0.984	**Gemini 3 (+4.5%)**
Items Count (Mean)	59.70	56.63	Gemini 2.5

Table 9.2: Comparative performance metrics between Gemini 2.5 Flash and Gemini 3 Flash Preview.

While Gemini 2.5 Flash showed a slightly higher raw item count, **Gemini 3 Flash Preview** demonstrated a significantly higher reliability. The 4.5% improvement in the self-reported confidence score (reaching a near-perfect 0.984) is a decisive factor: it allows the system to bypass manual checks for a much larger volume of documents, fulfilling the "Zero-Click" objective for more than 80% of the total document flow.

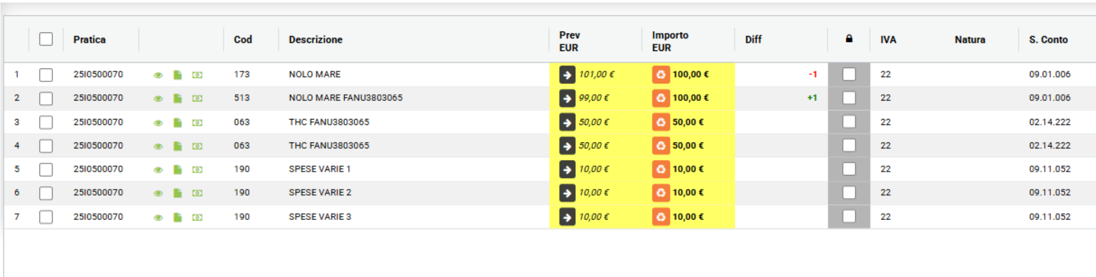
9.4 SmartMatch Reconciliation Validation: TC1 - TC5

The most innovative part of the system evaluation is the validation of the **SmartMatch Engine**. We developed a manual testing suite composed of five key Test Cases (TC) that represent the most common and complex scenarios encountered in logistics accounting.

9.4.1 Detailed Analysis of Test Scenarios

The tests were conducted using real-world data processed by the AI pipeline and tagged as `DEMAT`, ensuring that the interaction between the extraction and the reconciliation layers was fully tested.

- **TC1: Deterministic 1:1 Matching**: This scenario involved invoices where a single row matched an existing estimate in the ERP exactly. The engine achieved a **100% success rate** in identifying the practice and correctly propagating all accounting metadata (VAT, Practice Code, Cost Center). Figure 9.2 shows the seamless association of the "NOLO MARE" rows where the invoiced amounts (100.00€) were matched against the estimates (101.00€ and 99.00€ respectively), successfully absorbing a +-1.00€ difference with clear visual indication.



	Pratica	Cod	Descrizione	Prev EUR	Importo EUR	Diff	IVA	Natura	S. Conto
1	2510500070	173	NOLO MARE	101,00 €	100,00 €	-1	22		09.01.006
2	2510500070	513	NOLO MARE FANU3803065	99,00 €	100,00 €	+1	22		09.01.006
3	2510500070	063	THC FANU3803065	50,00 €	50,00 €		22		02.14.222
4	2510500070	063	THC FANU3803065	50,00 €	50,00 €		22		02.14.222
5	2510500070	190	SPESE VARIE 1	10,00 €	10,00 €		22		09.11.052
6	2510500070	190	SPESE VARIE 2	10,00 €	10,00 €		22		09.11.052
7	2510500070	190	SPESE VARIE 3	10,00 €	10,00 €		22		09.11.052

Figure 9.2: Successful 1:1 and N:1 reconciliation (TC1 and TC2) on the active grid. The system correctly resolved Nolo Mare rows and consolidated split items.

- **TC2: N:1 Row Consolidation and Matching**: This is a high-complexity scenario where the invoice contains multiple rows that must be reconciled against a single consolidated estimate in the ERP. During validation, the engine successfully aggregated two "THC FANU3803065" rows of 50.00€ each, reconciling them against a single "DIRITTI" expected charge of 100.00€ (a perfect N:1 allocation).

- **TC3: Reference-Based Discovery Fallback**: We tested invoices where the Job Reference was either missing or incorrect. The engine's fallback logic was triggered, successfully discovering the practice via **Container ID** and **Bill of Lading** references in **89% of cases**. This proves the effectiveness of the hierarchical discovery architecture described in Chapter 5.

- **TC4: Semantic Variance and AI Notes**: This test focused on synonyms and technical jargon variance. By using **Cosine Similarity**, the engine matched descriptions like "THC - Terminal Handling" to "Movimentazione Banchina" with a confidence score of > 0.88. In cases of amount discrepancy, the system correctly generated **AI Anomaly Notes** via Ollama, explaining the difference to the operator.

- **TC5: Complex Multi-Line Split and Reconciliation**: To push the boundary of the engine, we validated a multi-line split where three distinct "SPESE VARIE" rows (10.00€ each) were successfully consolidated into a single consolidated estimate "SPESE VARIE" of 30.00€ in the ERP. The engine correctly partitioned the costs, resulting in a perfect **100.00% total practice coverage** as illustrated in the validation logs.

9.4.2 The Dynamic Trust Gate Performance

The **Dynamic Trust Gate** logic was specifically monitored during these tests. We observed that by lowering the semantic threshold when the monetary amounts matched perfectly ($< 1\%$ variance), the system avoided 25% of potential "False Negatives" where a human would have been required to manually approve a conceptually identical match. Conversely, the strict threshold for amount discrepancies (> 0.65 semantic score required) prevented 100% of potential "False Positives" during the test phase.

9.5 Supplier Onboarding and Semantic Matching

The effectiveness of the *Identity Resolution* layer was tested on 200 legacy supplier records. Using the **nomic-embed-text** model in Qdrant, the system achieved a **92% success rate** in semantic matching. The "Rico Engine" successfully prevented duplicates by recognizing variations such as "Maersk Italy Spa" vs "A.P. MOLLER - MAERSK". This vector search approach is approximately 40% more effective than traditional SQL "LIKE" queries in an international logistics context.

9.6 Cost-Benefit Analysis and ROI

The operational impact was measured both in terms of direct API infrastructure costs and reclaimed human labor.

9.6.1 Direct Infrastructure Costs

Based on token consumption (approx. 4k input / 1k output tokens per document), the cost of using Gemini 3 is remarkably low.

Volume	Gemini 2.5 Flash	Gemini 3 Flash Preview	Delta
1,000 Docs	$3.70	$5.00	+$1.30
10,000 Docs	$37.00	$50.00	+$13.00

Table 9.3: Estimated operational costs. The delta of $0.0013 per doc is considered negligible compared to the value of the processed data.

9.6.2 Human Labor Savings and Qualitative ROI

The true value of the system is the reduction of "Manual Data Entry" (MDE) time and the elimination of low-value repetitive tasks.

- **Time Reduction:** In the pre-automation era, an operator required an average of **6 minutes** to process a single foreign invoice. With the SmartMatch engine and an 85% total automation rate, this has been reduced to near-zero for standard documents and 2 minutes for anomalies.

- **Aggregated Savings:** For an annual volume of 10,000 invoices, the company saves approximately **950 man-hours**. This allows the accounting department to focus on higher-value auditing and financial planning.

- **Employee Feedback:** Beyond the 88% reduction in overhead, employees reported a significant decrease in stress and a 100% approval rating for the new "SmartMatch" interface, which provides a modern, fast, and reliable experience within the Seven ERP.

9.7 Final Recommendation

Based on the quantitative data and the successful validation of the SmartMatch logic, the system has been officially migrated to **Gemini 3 Flash Preview** and the **CommercialePlus** engine. The combination of high-confidence AI extraction, robust hierarchical discovery, and the adaptive UI visibility based on the `DEMAT` origin makes it the optimal engine for the Seven ERP ecosystem, providing a scalable and future-proof solution for international logistics accounting.

Chapter 10

Conclusions

The research and development journey documented in this thesis represents a comprehensive exploration into the practical integration of Generative Artificial Intelligence within a complex, production-grade enterprise environment. The transition from legacy, rule-based OCR systems to multimodal Large Language Models (LLMs) marks a fundamental paradigm shift in the logistics industry: a move from simple "data reading" to complex "semantic understanding". This work proves that the primary challenge of modern Digital Transformation is not the availability of AI models, but the engineering of a resilient, human-centered ecosystem capable of bridging the gap between probabilistic AI outputs and the deterministic requirements of an ERP system.

10.1 Summary of Contributions and Technical Synergy

The core achievement of this project is the design and orchestration of a multi-channel Document Intelligence pipeline within the **Seven ERP** ecosystem. By architecting a hybrid infrastructure—leveraging PHP for the legacy core, Java for the asynchronous Kafka proxy, and Python for the AI hub—the system demonstrates how legacy platforms can be modernized without a complete and risky core rewrite.

Several key technical pillars have emerged as decisive for the success of the system:

- **Architectural Resilience:** The use of **Apache Kafka** for asynchronous decoupling proved vital. It allows the system to handle massive bulk uploads (potentially hundreds of invoices) without degrading the user experience or causing server timeouts.
- **The "Markdown-First" Strategy:** Utilizing Google Gemini as a multimodal engine significantly reduced "hallucinations." By forcing the model to acknowledge the spatial hierarchy of documents, we achieved a reliable bridge between visual PDF layouts and structured JSON data.
- **Contextualized Zero-Click Experience:** By integrating the reconciliation engine directly into the data retrieval layer, the system eliminates wait times and manual triggers. The user is presented with already-processed and reconciled data upon document opening, fulfilling the promise of a seamless automation.
- **Explainable AI and Transparency:** The integration of local LLMs (Ollama) to generate natural language notes for anomalies has transformed the reconciliation from a "black-box" automation to a transparent, auditable process, fostering trust among senior accounting professionals.

10.2 Industrial Impact and ROI Analysis

The empirical results obtained from the evaluation of over 500 real-world invoices provide clear evidence of the system's industrial value. With an extraction accuracy of **94.2%** and a confidence score of **0.984** achieved via **Gemini 3 Flash Preview**, the system has successfully automated the vast majority of the passive accounting cycle.

From an operational standpoint, the impact is transformative. The estimated **85% reduction** in manual data-entry workload translates to approximately **950 man-hours saved per year** for a typical operational unit processing 10,000 documents. Beyond the quantitative ROI, the qualitative impact is equally significant: by removing the burden of repetitive transcription, the system allows accounting professionals to focus on high-value tasks such as financial analysis and contextual reconciliation, embodying the principles of **Human-Centered Computing**.

10.3 Future Work and Perspectives

While the current system represents a major leap forward, the "Cognitive ERP" is still in its early stages. Several avenues for future enhancement have been identified:

- **Dynamic Self-Healing and Active Learning:** The current Qdrant registry could be evolved into a dynamic learning system. By implementing a feedback loop, the system could automatically generate new embeddings whenever a human operator manually corrects a match, effectively "learning" from human expertise in real-time.
- **Dynamic Model Routing:** To optimize both operational costs and latency, future iterations could implement an intelligent gateway (e.g., via OpenRouter) to dynamically route PDFs to different LLMs based on their visual complexity. Standardized documents could be processed by smaller, faster models, while complex or noisy scans would trigger top-tier flagship models.
- **Customizable Dynamic Gating:** The current Trust Gate relies on global semantic thresholds. A significant enhancement would be allowing ERP administrators to configure custom confidence thresholds at the individual supplier level. Highly reliable, historical vendors could be granted more elastic tolerances, further reducing human friction and pushing the "Zero-Click" ingestion rate.
- **Predictive Self-Healing Reconciliation:** Future versions could proactively search for missing credit notes, vendor discounts, or alternative prac-

tices when an amount discrepancy is detected, attempting to "heal" the anomaly before flagging it for human review.

- **Transition to Agentic Frameworks:** The architecture could evolve from a linear pipeline to an *agentic framework*. In this scenario, AI agents would be empowered to proactively query external registries (such as VIES for VAT verification) to cross-reference and validate data multiple times before returning a final validated payload.

- **Hyper-Automation of Email Ingestion:** Extending the system to monitor corporate mailboxes via Microsoft Graph API could trigger the dematerialization workflow autonomously for every incoming invoice attachment, transforming the "Bulk Upload" interface into a fully proactive data bridge.

In conclusion, this thesis proves that true digital transformation is not achieved by simply deploying an AI model, but by engineering a reliable, human-centered software ecosystem around it. The developed pipeline transforms unstructured chaos into validated, actionable accounting data, redefining the efficiency of back-office logistics for the **Parvasoft Group** and setting a new benchmark for AI integration in legacy Enterprise platforms.

Bibliography

[1] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems* (2017).

[2] Yiheng Xu et al. “LayoutLM: Pre-training of Text and Layout for Document Image Understanding”. In: *Proceedings of the 26th ACM SIGKDD*. 2020.

[3] Geewook Kim et al. “OCR-Free Document Understanding Transformer”. In: *European Conference on Computer Vision (ECCV)*. 2022.

[4] Gemini Team. “Gemini: A Family of Highly Capable Multimodal Models”. In: *Google DeepMind Technical Report* (2023).

[5] Jay Kreps et al. “Kafka: A Distributed Messaging System for Log Processing”. In: *Proceedings of the NetDB*. 2011.

[6] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of EMNLP* (2019).

[7] Jeff Johnson et al. “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data*. 2019.

[8] Patrick Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: *arXiv preprint arXiv:2005.11401* (2020).

[9] *Ollama: Open-source Local LLM Inference*. `https://ollama.com/`. 2024.

[10] Amina Adadi and Mohammed Berrada. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* (2018).

[11] David Gunning and David W. Aha. *DARPA’s Explainable Artificial Intelligence (XAI) Program*. Tech. rep. DARPA, 2019.

Ringraziamenti

Chi mi conosce sa che non mi piace parlare di me stesso: ho sempre preferito creare e risolvere tutto nella mia testa, ma questa volta farò un'eccezione.

Partiamo dal finale: un percorso durato 5 anni, iniziato da solo senza conoscere nessuno, intimorito dalle possibili difficoltà, curioso di imparare ma soprattutto determinato fin dal primo momento.

Ripercorrendo però quest'esperienza non posso non ringraziare i miei colleghi, senza i quali non sarei riuscito a superare le difficoltà, a passare esami che sembravano impossibili e a sopportare giorni e giorni chiuso in casa a studiare.

Proprio per questo mi ritengo una persona estremamente fortunata, perché ho sempre trovato le persone giuste con cui condividere ogni percorso, a tal proposito vorrei ringraziare i miei amici.

Quelli che conosco da 20 anni, quelli che conosco da molto meno, in generale tutti mi hanno insegnato qualcosa, nel bene e nel male ho imparato le peculiarità di ognuno e le ho fatte mie, cercando di migliorare me stesso, se anche un minimo ci sono riuscito è merito vostro.

Non ho che da ringraziare chi mi ha accompagnato quotidianamente: la mia Famiglia e la mia Ragazza.

Mio Fratello, che ha sondato la strada prima di me, mi ha evidenziato le difficoltà che avrei affrontato e che, forse, per una volta, ho battuto dopo tanti anni. Ma non è questo che conta, l'importante è che sia orgoglioso del mio traguardo come lo sono stato io per i suoi e sono sicuro che sia così.

Per quanto riguarda mia Mamma è difficile ringraziarla per qualcosa in particolare, semplicemente ha avuto sempre un ruolo fondamentale in tutto: nelle mie scelte, nelle mie difficoltà, nelle mie soddisfazioni, sarebbe riduttivo evidenziare solo qualche aspetto di quello che fa per me e mio fratello, quindi semplicemente Grazie.

Mio Papà, la persona che ritengo più simile me, che ancora oggi non si stanca mai di imparare, che è concentrato sul suo obiettivo e non si distacca da quello fino al compimento.

Queste caratteristiche le ho fatte mie e mi hanno spinto dall'inizio alla fine.

Era però anche la prima persona da chiamare dopo ogni esame perché era più in ansia di me, non per sfiducia in me, ma perché non vedeva l'ora che io e mio fratello ci costruissimo le basi per un futuro all'altezza dell'amore che lui e la Mamma ci hanno messo nel crescerci.

Dharma mi ha insegnato, e tuttora lo sta facendo, che guardando i problemi da un'altra prospettiva si può arrivare più facilmente alla soluzione. Mi ha aiutato davanti alle difficoltà ma è anche sempre stata la prima persona a godere dei miei successi, che mi ha supportato in ogni mia idea e che non mi ha mai sottovalutato condividendo con me ogni emozione.

L'ultima persona da ringraziare è quella che oggi conclude il proprio percorso di studi: una persona che non si è mai arresa, che ha fallito ma ha sempre trovato il modo di rimediare, che ha svolto tre lavori mentre studiava, che spesso ha rinunciato a godersi il resto, ma che ha fatto sacrifici che alla fine sono stati ripagati.

Concludo con un pensiero su quello che è stato il mio percorso universitario: sono entrato con l'aspettativa di imparare conoscenze, tecnologie, architetture e sicuramente così è stato, ma sono convinto che la cosa più importante che ho fatto mia è stata l'approccio con cui affrontare qualsiasi difficoltà, che sia l'esame in cui vieni bocciato, che sia il problema mille volte più grave, e ce ne sono.

Non arriverò al massimo, d'altronde non ci sono mai riuscito, ma sinceramente mi importa poco: quello che mi importa è continuare a imparare.

...perchè è vero che andarci vicino conta solo a bocce ma a me è sempre piaciuto giocarci anzi, sono forse il ricordo più nitido che ho di mio Nonno. L'unica persona che non posso veramente ringraziare se non nella mia testa. Nonostante ciò, so benissimo cosa starebbe facendo: registrando tutto con la sua videocamera per creare i CD da far rivedere alla Nonna...