MSc Data Science & AI for Business

RESEARCH PAPER

Academic Year 2024 – 2025

The Hybridization of Consultancy Work: Enabling Human-AI Partnership through Multi-Agent Systems

Gaspard Hassenforder

Under the supervision of

Vincent Fraitot

July 2025 – Oral Presentation

■ *Public Report* □ *Confidential Report*

Executive Summary

Consultancy partners face significant challenges in balancing client engagement with time-consuming administrative tasks such as proposal drafting and research. This Research Paper describes a system designed to automate these repetitive tasks using a hierarchical multi-agent system. By leveraging knowledge graphs, specialised agents, and external tools, the system generates comprehensive company profiles, conducts targeted business research, and drafts tailored proposals. The implementation shows potential efficiency gains and strategic advantages, allowing consultants to focus on strategic client work. The system's architecture, including its use of knowledge graphs and multi-agent collaboration, is detailed, along with real-world use cases and time comparisons. The findings suggest that human-AI collaboration can significantly change consultancy workflows, while also discussing current constraints and future improvements. This work is a practical example of AI augmentation in professional services and offers practical insights for implementing AI systems in consultancy practices.

Contents

1 Introduction

1.1 Context

Consultancy partners struggle to balance client engagement with time-consuming tasks like producing bespoke proposals for each opportunity. Traditionally, this process is highly manual: partners either delegate research tasks to junior staff or undertake the research themselves, often relying on a patchwork of online sources, internal documents, and past proposals. This workflow is not only time-consuming but can also create significant bottlenecks, slowing down the entire business development cycle and sometimes resulting in missed opportunities.

Recent industry research highlights the scale of this challenge. On average, small and midsize RFP (Request for Proposal) teams spend around 20 hours per proposal response, while larger firms report an average of 23 hours and the involvement of up to nine people per proposal [7]. Moreover, 62% of proposal professionals report working more than 40 hours per week on proposals, with 77% stating that their proposal process is far from ideal [12]. Despite the significant effort invested in crafting these documents, much of the content is repetitive. This repetitiveness arises from the consultancy's specialization in a few key fields, leading to similar missions being proposed across different clients. Additionally, various partners within the larger consultancy may have handled similar missions, resulting in overlapping content that partners may not be fully aware of. Ironically, clients often only scan these documents quickly, focusing primarily on pricing and key deliverables [7]. This inefficiency not only drains valuable partner time but also delays client engagement and decision-making, reducing the overall agility of the consultancy firm.

1.2 Motivation

The consultancy company where this project is being implemented is actively seeking to shift its business model to better align with the evolving expectations of both clients and partners. The goal is to enable partners, those who interact directly with clients, assess needs, and maintain relationships, to focus more on high-value activities such as strategic conversations, opportunity identification, and solution co-design, rather than on repetitive administrative tasks. By reduc-

ing the time spent on manual research and proposal writing, partners can increase the frequency and quality of client interactions, making calls more action-oriented and responsive to client needs. This shift should improve client satisfaction and enhance the job satisfaction and effectiveness of partners, who can devote more energy to building trust and delivering value.

Furthermore, as the consulting industry becomes increasingly competitive and data-driven, the ability to respond rapidly and insightfully to client needs is becoming a key differentiator. Firms that can streamline their internal processes and leverage technology to augment human expertise are better positioned to win new business and deepen existing relationships.

1.3 Objective

This project investigates the central research question: **To what extent can a hierarchical multi-agent system, integrated with a knowledge graph, reduce proposal generation time while enhancing quality in a consulting context?**

The primary objective is to design, and develop a proof-of-concept for a novel system intended to automate the administrative and research-intensive aspects of proposal creation. A secondary objective is to propose an evaluation framework and conduct a simulation-based estimate of the system's potential impact on efficiency.

Crucially, this research also evaluates the qualitative dimension of the output. The hypothesis is not merely about speed, but about enhancing, the quality, relevance, and strategic nuance of the proposals. A key aspect of the objective is to create a human-AI partnership where the system seamlessly assists the consultant's thinking process, augmenting the consultant's expertise rather than replacing it. The system's success will be measured by its ability to produce high-quality, customized proposals that are both generated efficiently and align with the firm's strategic objectives, thereby demonstrating a viable path toward a more agile and data-driven consultancy workflow.

2 Literature Review

2.1 Human-AI Partnership in Different Fields

Healthcare: AI agents can manage routine administration. A recent study found that healthcare professionals expect AI agents to cut doctors' paperwork by approximately 30%, saving one day per week and allowing clinicians to spend more time with patients [13]. Such systems handle tasks like scheduling, record-keeping, and care coordination, freeing doctors and nurses to concentrate on patient care.

Legal & Regulatory: In law firms and corporate compliance, AI agents sift through contracts, regulations, and case law. For example, AI-powered contract systems automatically flag non-standard clauses, suggest revisions, and update agreements in bulk when laws change [10]. This enables lawyers to delegate tedious document review and focus on complex negotiation and advocacy.

Education: Multi-agent AI creates personalized learning experiences for students. Intelligent tutoring systems continuously assess each learner's progress and adapt content accordingly. Educators use these AI tools to automatically grade work and track performance, allowing teachers to spend more time mentoring and addressing students' individual needs [14]. Routine tasks such as grading and feedback are automated, enabling teachers to focus on pedagogy and student engagement.

These examples illustrate a common pattern: AI excels at high-volume, rule-based tasks (data processing, monitoring, document analysis), while humans handle judgment, creativity, and emotional intelligence [1]. By dividing work this way, multi-agent systems create hybrid teams that leverage the complementary strengths of humans and AI [11].

2.2 The Use of AI in Consultancy

Automation of Routine Tasks: Tasks like drafting slide decks, writing standard reports, and updating client presentations are increasingly automated with AI. For instance, firms use tools that generate PowerPoint slides from bullet points or auto-fill charts from data. A field experiment with BCG consultants found that

access to a GPT-4–powered assistant made consultants approximately 25% faster and improved the quality of their work by about 40% on analytical tasks [8]. This allowed consultants to focus on higher-value analysis and client interaction.

Generative Content and Brainstorming: Consultants leverage LLMs and chatbots as ideation and drafting partners. A simple prompt can yield business ideas, draft email responses, or suggest negotiation strategies. For example, McKinsey's internal AI "Lilli" pulls from over 100,000 documents and interviews to assist with research and brainstorming, reportedly saving consultants up to 30% of their time [18]. Similarly, Deloitte's "Sidekick" (a ChatGPT-based tool) is used for brainstorming, summarizing documents, editing reports, and even generating code or email templates [18]. These AI companions accelerate creative work and improve consistency, but human consultants still guide the final decisions.

Overall, literature shows that AI in consulting augments rather than replaces human experts. AI platforms enable consultants to work faster and with richer data support, removing many manual bottlenecks [8, 18]. At the same time, top firms maintain that human expertise remains central: AI handles volume and speed, while consultants manage relationships, problem-solving, and strategic thinking.

While the literature confirms the benefits of AI in augmenting consulting tasks, few studies have measured the impact of integrated multi-agent systems on the entire proposal process. This research aims to fill that gap by quantifying the efficiency gains and qualitative outcomes of such a system in a real-world setting.

3 Conceptual Foundations

3.1 What Are AI Agents?

In the rapidly evolving field of artificial intelligence (AI), clarity of terminology is essential. The term *AI agent*, which surged in popularity in late 2024, has often been used imprecisely. In this section, we introduce a structured definition, provide illustrative examples, and distinguish true agentic systems from simpler automated workflows.

3.1.1 Working Definition

We adopt the following working definition:

> **AI agent**: A system that autonomously performs tasks on behalf of a user or system by generating its own workflows, leveraging external tools, and adapting to changing situations using reasoning and feedback.

The reader is assumed to be familiar with foundational AI concepts, including large language models (LLMs), which are of the main technologies that powers many agentic systems.

3.1.2 Spectrum of Automation to Agency

It is important to clarify the distinction between different types of systems that incorporate AI, particularly when introducing the concept of AI agents. Not all systems that use LLMs or even perform multi-step operations qualify as agents. We propose a spectrum ranging from basic automation to full agentic behavior:

- **Automated Workflows (Non-Agentic)**: These systems carry out predefined sequences of tasks involving LLMs or other AI models, without adaptive reasoning or planning. Despite multiple steps or the use of multiple tools, the logic is entirely scripted and lacks autonomy.
  - *Example*: A client onboarding pipeline. A consultant uploads a client's preliminary data sheet to a system that uses OCR to extract key fields (e.g., company name, industry, key contacts). A scripted rule then assigns a preliminary client category based on industry codes. If the industry is not recognized, an alert is generated for manual review. Although multi-step and AI-enabled, the flow is rigid and non-adaptive.
- **Tool-Using LLMs (Borderline Cases)**: These systems allow LLMs to interface with external tools (e.g., code interpreters or plotting libraries), but only in a single-turn context. They do not construct long-term plans or modify their behavior based on outcomes.

  - *Example*: A client data visualization assistant. A consultant uploads a CSV file containing client financial data and asks the LLM to generate three visualizations (e.g., revenue trends, expense breakdowns, profit margins). The LLM uses a plotting library to produce the charts in a single step. It does not ask clarifying questions about specific data points of interest, update its plan based on the data content, or revise the visualizations in response to feedback from the consultant. Despite using tools, it lacks any autonomy or feedback loop.

- **True AI Agents**: These systems are designed around a reasoning engine capable of planning, decomposing goals, interacting iteratively with tools and environments, and adapting based on outcomes. They exhibit autonomy and goal-driven behavior across multiple steps.

  - *Example*: A market research agent. Given a query such as "analyze competitive threats in the European luxury fashion market", the agent autonomously plans subtasks including competitor identification, financial benchmarking, and trend analysis. It then iterates across various tools and revises its plan when knowledge gaps are detected. This behavior illustrates agentic reasoning with feedback and adaptation.

Why These Categories Matter. Distinguishing between automated workflows, tool-using LLMs, and true agents is essential for setting expectations and assessing capabilities. Tool integration alone does not constitute agency. Many current systems market themselves as "agents" despite lacking any autonomy, iteration, or goal adaptation. By clarifying this spectrum, we can better evaluate the degree of intelligence and flexibility required for real-world deployment and avoid overestimating what current models can achieve without agentic scaffolding.

3.1.3 Eliciting Reasoning from LLMs for Agentic Behavior

A core requirement for true agentic behavior is structured reasoning: the ability to decompose a task into steps, apply logical rules, and reflect on intermediate outcomes. While early large language models (LLMs) often produced fluent but shallow completions, recent advances in prompting techniques have enabled LLMs to make their reasoning explicit and verifiable.

Techniques such as **Chain-of-Thought (CoT)** prompting and **ReAct (Reasoning and Acting)** have proven effective in eliciting multi-step, verifiable reasoning from LLMs. These methods guide the model to "think out loud," breaking down a complex query into a sequence of intermediate steps and actions. This ability to externalize the reasoning process is a necessary (though not sufficient) condition for agency, as it allows for planning, tool use, and adaptation. This paper's agents are built by leveraging such techniques to ensure goal-directed and auditable behavior.

> **Illustrative Example:** Consider a task such as: "Summarize the current state of research on battery degradation in electric vehicles (EVs)."
>
> A typical reasoning LLM may proceed as follows:
>
> 1. Identify subtasks: search for relevant papers, extract findings, identify common themes, and synthesize a summary.
> 2. Retrieve documents using scholarly databases or APIs.
> 3. Read and summarize each paper individually, highlighting methodology, results, and limitations.
> 4. Compare results across papers to identify agreements, contradictions, and open questions.
> 5. Reflect on gaps in the information and decide whether additional sources are needed.
> 6. Compose a final synthesis organized by theme, methodology, or chronological development.

This is not just multi-step text generation, it's goal-directed reasoning that can span multiple tools and interactions. The LLM must infer structure, revise hypotheses, and resolve ambiguity within its **plan–execute–adapt** loops.

3.1.4 Core Components of AI Agents

At their core, AI agents combine the reasoning power of LLMs with:

- **Tool Integration**: APIs, calculators, search engines, databases, etc.
- **Memory Systems**: Episodic and long-term memory for context retention.
- **Planning Modules**: Capabilities to decompose tasks into steps.
- **Feedback Loops**: Self-evaluation and error correction mechanisms.

These elements enable interaction with both virtual (e.g., web browsers, APIs) and physical environments (e.g., robotics).

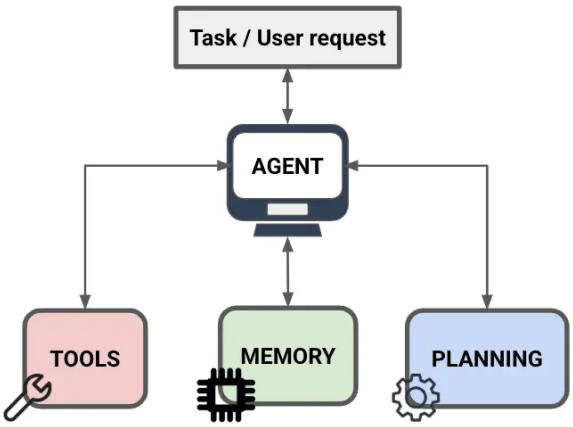

Figure 1: Core components of an AI agent

3.2 From Single Agent to Multi-Agent Systems

The choice between single-agent and multi-agent systems (MAS) is a defining question in modern AI system design. Single-agent systems offer simplicity and lower resource demands, making them ideal for focused, well-defined tasks. In contrast, MAS distribute intelligence across multiple agents, enabling specialization, scalability, and resilience but at the cost of increased coordination and complexity [5, 4].

MAS can be classified along several axes: homogeneous vs. heterogeneous (identical vs. specialised agents), cooperative vs. competitive, and centralized vs. decentralized. This work focuses on *heterogeneous, cooperative, centralized* MAS, where specialised agents, each with unique tools or expertise, collaborate under the direction of an orchestrator. This architectural choice is central to our hypothesis. We argue that a heterogeneous, cooperative MAS is better suited than a single monolithic agent to handle the distinct sub-tasks of proposal generation (e.g., web research, database querying, content synthesis), thereby improving both the efficiency and the quality of the final output.

Recent debates highlight the trade-offs. Critics like Cognition AI's Walden Yan argue that MAS can be fragile due to context fragmentation and conflicting decisions, advocating for "context engineering" in single-agent systems [17]. Others, such as Anthropic, have demonstrated that MAS can deliver significant performance gains in research and synthesis tasks by leveraging modularity and specialization [5].

As LLMs improve in context length and tool use, the performance gap between MAS and single-agent systems is narrowing, motivating hybrid approaches that dynamically select the best architecture for each task [5].

3.2.1 Why Use Multi-Agent Systems

While a single, powerful agent with a large context window can be engineered to call multiple tools, this monolithic approach is vulnerable to "contextual degradation" and "chain-of-thought drift," particularly in complex, multi-step tasks like proposal generation. As a reasoning chain lengthens through multiple tool calls, research queries, and synthesis steps, the agent's focus can drift from the original strategic intent. A hierarchical Multi-Agent System (MAS) is architected specifically to mitigate this risk.

By delegating sub-tasks to specialised agents, the Orchestrator ensures that each component operates with a fresh, narrowly-defined, and short-lived context. The Graph Database Agent, for example, is exclusively concerned with the graph schema and query optimization; it does not need to retain context about web search results from a previous step. This modular design creates a more structured reasoning process, preventing the kind of drift that can undermine the quality

and relevance of the final output. The MAS architecture is therefore a deliberate choice to enhance reliability and maintain strategic coherence throughout the entire workflow. The key factors justifying this choice are:

- **Reduced Cognitive Load and Enhanced Specialization:** Each agent in the hierarchy operates with a focused context. For example, the Graph Database Agent is an expert in Cypher queries and the graph schema, without needing any knowledge of web scraping protocols. A single monolithic agent, by contrast, would need to manage the context, prompts, and failure modes for all possible tools simultaneously. This specialization in a MAS reduces the "cognitive load" on any single component, leading to more reliable and efficient execution of sub-tasks [16].
- **Improved Fault Tolerance and Resilience:** The hierarchical structure allows for more robust error handling. If the Web Agent fails (e.g., due to an external API outage or a change in a website's structure), the Orchestrating Agent can log the failure and either attempt a different strategy or complete the rest of the task with the available information. In a monolithic agent, a failure within its complex reasoning chain is more likely to halt the entire process, making the system more brittle [9].
- **Superior Scalability and Maintainability:** The MAS architecture is inherently more scalable. Integrating a new capability like the ability to modify and create PowerPoint slides may simply require developing the new specialised agent and registering it with the Orchestrator. This plug-and-play approach is significantly cleaner than re-engineering the core prompt and complex logic of a single, monolithic agent, which becomes increasingly difficult to maintain and debug as more tools are added [6].
- **Enhanced Traceability for Debugging:** When a workflow produces an unexpected result, the MAS architecture makes it easier to isolate the source of the error. If the financial data in a company profile is incorrect, the logs of the specific API calls can be checked directly, rather than sifting through the entire reasoning trace of a single agent. This modularity is a key tenet of agent-oriented software engineering that simplifies debugging [6].

By explicitly distributing intelligence, the MAS architecture creates a system that

is not only modular but also more resilient, scalable, and manageable than a single-agent counterpart for the complex, multi-domain task of proposal generation.

3.3 MCP Integration and Agent Coordination

A persistent challenge in building agentic and multi-agent systems (MAS) is the seamless integration of external tools, data sources, and services. Traditionally, connecting an AI agent to resources like email, calendars, or company databases required custom code and significant manual effort, limiting scalability and flexibility.

The *Model Context Protocol* (MCP), developed by Anthropic, addresses this challenge by providing a universal, standardized interface for connecting AI agents to diverse external systems. Often described as "the USB-C of AI applications," MCP enables plug-and-play discovery of tools, model-agnostic interoperability, and secure, governed access to APIs, databases, and other resources [2].

MCP is built on a client-server architecture, where:

- **MCP hosts** (AI agents) request data or actions,
- **MCP servers** expose tools and data sources,
- **MCP clients** act as intermediaries, facilitating communication.

A key feature of MCP is that the MCP client provides the AI agent with a complete description of available tools: what each tool does, how to call it, and all the necessary context for effective use. This means agents can dynamically discover and utilize new capabilities without custom integration work. The agent receives structured information about each tool's functions, parameters, and expected outputs, enabling it to reason about which tools to use and how to sequence them in workflows.

Crucially, MCP democratizes tool integration. Anyone can create an MCP-compatible tool and make it available for others to use. As a result, organizations and developers can rapidly expand the capabilities of their AI agents, while maintaining centralized control over access and security.

For multi-agent systems, this standardization is transformative. Agents or tools can coordinate, delegate tasks, and share resources seamlessly, all through the MCP interface no matter who created them. This not only reduces development overhead but also enables more adaptive, collaborative, and scalable MAS architectures.

4 System Design

4.1 Architecture Overview

Figure 2 presents the architecture of the current version of the Multi-Agent System (MAS) I developed. The red cylinders represent the external databases the system connects to. The light blue rectangles denote individual agents, while the green rectangles represent agent-based workflows (as described in Section 3.1.2).

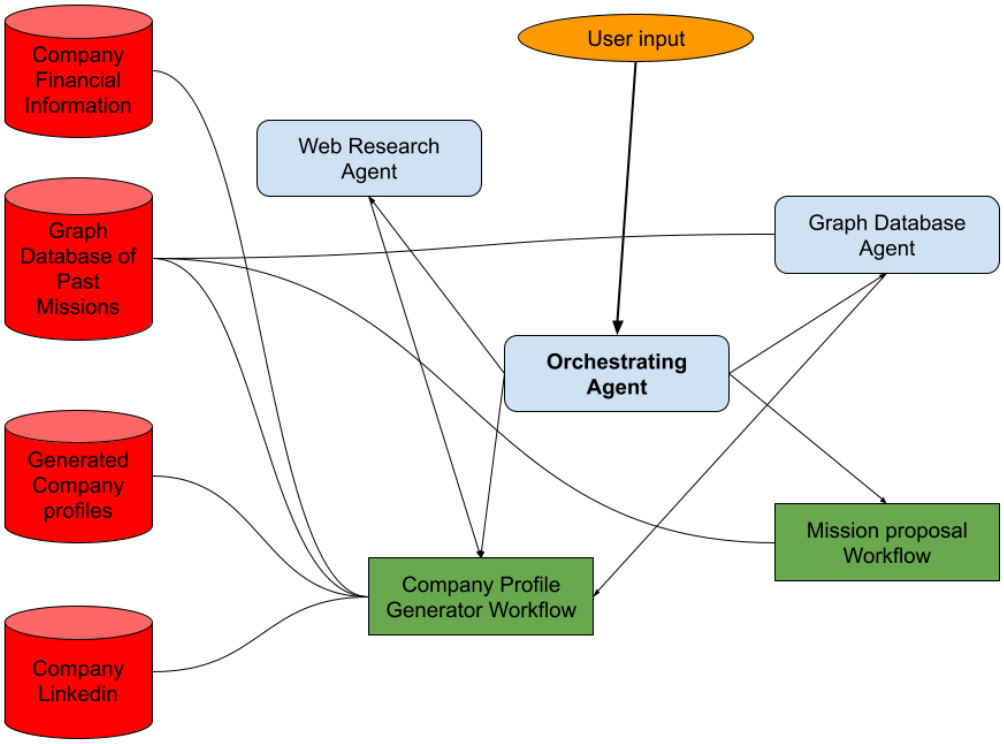

Figure 2: Overall architecture of the Multi-Agent System

The **Orchestrating Agent** plays a central role in decision-making. Based on the user's query, it determines whether to call another agent directly or initiate a predefined workflow. This agent operates under a structured system prompt that guides its behavior and tool usage. An example of such a prompt is shown below.

System Prompt for the Orchestrating Agent

```
You are a highly useful AI assistant designed to help users by
providing detailed and relevant information about clients, context,
consultants, missions, deliverables, and convictions.

Your primary goal is to interpret the available knowledge and deliver
clear, actionable insights to the user.

Always use the tools at your disposal to gather information never
rely solely on your pre-trained knowledge.

TOOLS:
------
You have access to the following tools: {tools}

To use a tool, please use the following format:

Thought: Do I need to use a tool? Yes
Action: the action to take, should be one of [{tool_names}]
Action Input: the input to the action
Observation: the result of the action

Previous conversation history: {chat_history}
New input: {input}
```

When passing the `tools` argument to the system prompt, we explicitly define the name of each tool function, the input it expects, the output it returns, and a brief description of its purpose. This structured format ensures that the Orchestrating Agent can understand how to call and use each tool effectively, the four tools currently available to the agent are:

Web Research Agent This tool can perform in-depth research on people, companies, or topics. It returns detailed results along with verifiable source links.

Graph Database Agent This agent queries a knowledge graph (see section 4.2) containing all past consulting missions. It retrieves structured insights about clients, contexts, consultants, deliverables, and more.

Mission Proposal Workflow Given a business problem provided by a client, this workflow retrieves relevant past missions from the graph and automatically generates multiple mission proposal options tailored to that client's needs.

Company Profile Generator Workflow Given a company name, this workflow aggregates and synthesizes relevant information (financials, recent news, employee data, etc.) to generate a comprehensive profile. The goal is to give consultants all the context they need before interacting with a client.

4.2 Knowledge Graphs in Business Intelligence

In consultancy, rapidly connecting expertise, client needs and past project outcomes provides a key advantage. Traditional relational databases, while robust, often struggle to capture the nuanced, evolving relationships that define consulting engagements. Knowledge graphs offer a more flexible way to model consultancy data, representing entities such as clients, consultants, deliverables, business contexts, convictions (expert methodologies), and proposals as interconnected nodes. This structure, illustrated in Figure 3, mirrors the real-world complexity of consulting work, where value is often created at the intersection of people, knowledge, and context.

Unlike rigid relational schemas, knowledge graphs allow new types of nodes and relationships to be added organically as the business evolves. This schema flexibility means that as new methodologies, client types, or deliverable formats emerge, the data model can adapt without costly redesigns or migrations. For a consultancy, this translates into greater agility and the ability to capture institutional knowledge as it grows.

A key technical advantage of graph databases is their efficiency in traversing relationships. Where a relational database might require multiple expensive joins to answer questions like "Which consultants have delivered projects in a specific business context using a particular methodology?", a graph database can answer such queries rapidly by following direct connections. This speed is not just a

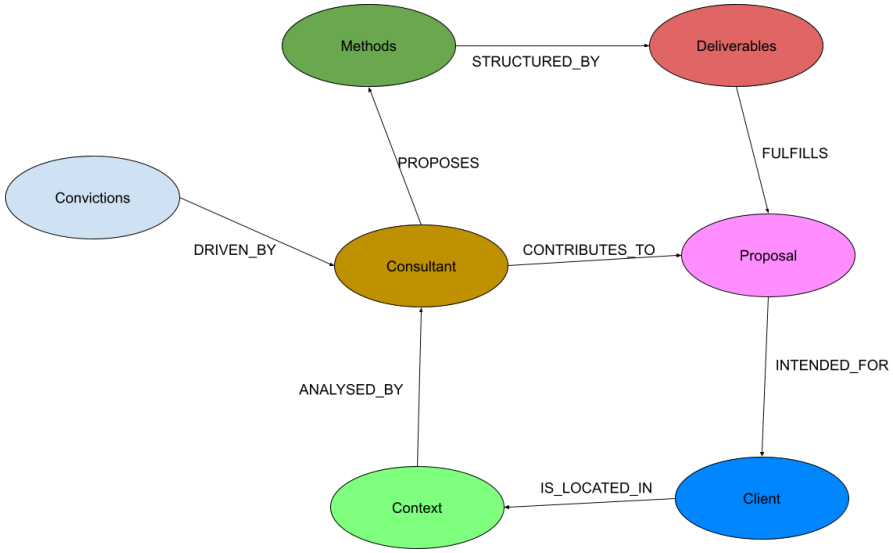

Figure 3: Schema of the Consultancy Knowledge Graph

technical benefit, it enables consultants and AI agents to surface relevant insights in real time, supporting faster, more informed decision-making.

Moreover, the semantic richness of knowledge graphs, where each relationship is explicitly labeled (e.g., `CONTRIBUTES_TO`, `ANALYSED_BY`), empowers both human consultants and AI agents to interpret the context of each connection. This clarity supports explainable recommendations and allows for more nuanced, context-aware analysis. Additionally, relationships in the graph can also store information, it isn't limited to nodes(i.e. vertices). For example, the `CONTRIBUTES_TO` relationship includes metadata such as the number of days a consultant worked on a given proposal. This information is valuable when estimating the effort required for similar future proposals (see Section 4.7).

One of the most powerful applications of this approach is community detection. Using the Leiden algorithm [15], we can automatically identify clusters of related missions, clients, or expertise within the graph. The Leiden method improves upon earlier techniques by ensuring that communities are well-connected and meaningful, avoiding the formation of loosely related or fragmented groups. In practice, this means that consultants and AI agents can discover strategic clusters, such as

emerging market themes or concentrations of expertise, without manual curation. For example, when drafting a proposal, the system can recommend similar past missions or highlight patterns in successful engagements, enabling more strategic and evidence-based recommendations. These clusters are not only actionable but also fully explainable, as each can be traced back to the underlying nodes and relationships in the graph.

Another major advantage of using a graph database is its ability to uncover valuable connections that might otherwise remain hidden. For instance, identifying internal domain experts becomes straightforward: by analyzing the strength and frequency of relationships between consultants and certain thematic areas, the system can surface individuals with deep experience in specific fields. This means that if a consultant is preparing a proposal on, say, AI for supply chain optimization, they can instantly identify and reach out to colleagues who have repeatedly contributed to related missions.

The power of graph databases is further exemplified by their query languages. Cypher, the declarative language used by Neo4j, allows for intuitive and expressive queries that would be cumbersome in SQL. For instance, to find all proposals and all clients a specific consultant has worked given a specific business context, one might write:

```
MATCH (c:Consultant)-[:CONTRIBUTES_TO]->(p:Proposal)-[:INTENDED_FOR]->
(clt:CLIENT)-[:IS_LOCATED_IN]->(ctx:Context),
WHERE ctx.name = 'Digital Transformation' AND c.name = 'Gaspard'
RETURN p, clt
```

Such a query leverages the graph's structure to surface relevant knowledge instantly, supporting consultants in building tailored, data-driven proposals. In a traditional relational database, this would require multiple joins and complex SQL, often resulting in slower performance and less intuitive query logic.

It is important to note, however, that populating and maintaining a rich knowledge graph requires a significant initial investment in data integration and curation. Yet, this effort pays dividends by enabling more strategic recommendations, improving the explainability of AI-driven outputs, and allowing both consultants and AI agents to develop a deeper, more contextual understanding of the firm's collective expertise.

4.3 Hierarchical Agent Collaboration

In a multi-agent system designed for consultancy, agents do not operate in isolation; instead, they collaborate in a structured, hierarchical manner to address complex business challenges. At the top of the hierarchy is the *Orchestrating Agent*, which receives high-level consultancy requests.

This hierarchical structure closely mirrors the organization of human consulting teams and can be seen as a digital parallel to it. The Orchestrator Agent acts as the project lead, dynamically delegating tasks to specialised agents (analogous to domain experts and analysts), while the Knowledge Graph serves as the institutional memory a senior consultant would possess after decades of experience.

A key advantage of the hierarchical agent collaboration model is its inherent support for explainability and traceability. The Orchestrating Agent acts as the system's conductor, not only delegating tasks but also recording the rationale behind each decision, such as why a particular agent or tool was chosen for a given subtask, and how the call was executed. This structured reasoning chain is invaluable for several reasons. First, it enables consultants and system designers to debug workflows and understand the root causes of errors, facilitating continuous improvement of both AI agents and business processes. Second, it provides a transparent audit trail: every recommendation or output can be traced back to its source data and the sequence of agent interactions that produced it. This is essential in consultancy and directly addresses the 'maintaining quality' component of our research question. The transparent audit trail allows a human consultant to verify the provenance of every claim in the generated proposal, ensuring the final output is not a "black box" and meets the firm's standards for quality and integrity.

While explainability is possible in other multi-agent system architectures, such as flat or fully decentralized MAS, it is often more challenging to reconstruct the reasoning process when agents operate independently and without a central coordinator. In contrast, the hierarchical approach naturally organizes decision-making and communication flows, making it easier to surface and present the logic behind each step.

Furthermore, the modular nature of hierarchical agent collaboration means that additional tools or specialised agents can be integrated without disrupting existing

workflows.(see section 7.4 for examples of these additions)

4.4 Web Agent

The Web Agent is an autonomous system designed to gather and synthesize internet-based information in response to research topics or business inquiries. As shown in Figure 4, the agent implements a modular pipeline that replicates human research workflows while leveraging the speed and scalability of automated processes.

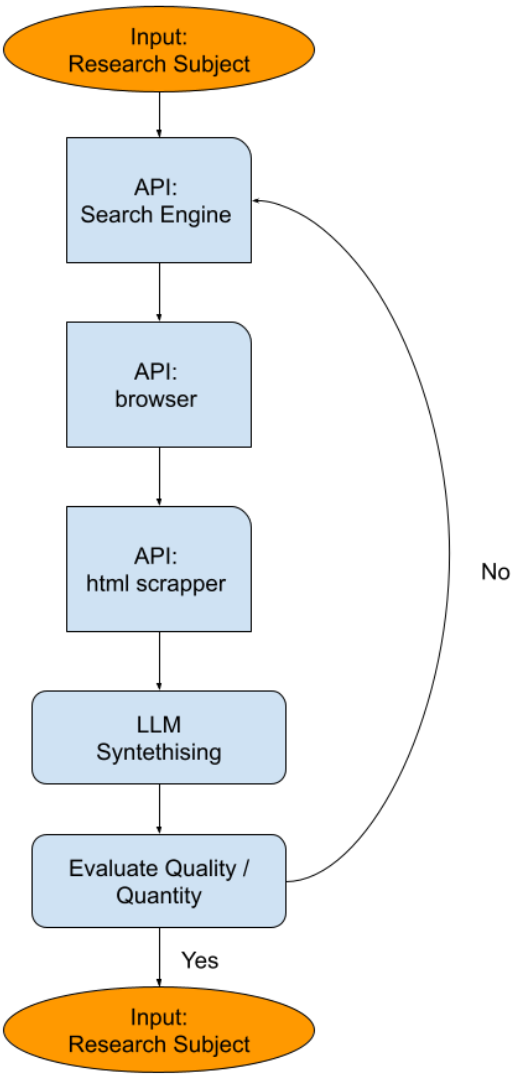

Figure 4: Architecture and workflow of the Web Agent system

The agent's operation begins when it receives a research subject. Its first step involves querying the Bing Web Search API to identify relevant web resources. To efficiently process these resources, the system implements parallel batch processing, simultaneously handling five links at a time to optimize speed and throughput.

Each retrieved link undergoes processing through Playwright, a headless browser solution that enables full rendering of dynamic web content, exactly as a human user would experience it. Following successful page loading, Selenium-based extraction modules parse and clean the HTML content, preserving primary textual information while systematically removing non-essential elements such as advertisements, navigation menus, and other page furniture.

The extracted content then flows to a GPT-4 based analysis module, which performs consolidation, summarization, and contextual adaptation to produce business-relevant outputs. The system incorporates quality control mechanisms that automatically evaluate both the quantity and relevance of gathered information. If insufficient data is obtained from a given source, the pipeline seamlessly progresses to subsequent resources in the batch.

4.5 MCP Integration and Graph Querying Agent

Building upon our knowledge graph infrastructure (Section 4.2), the Graph Querying Agent leverages Neo4j's native Model Context Protocol (MCP) tools to enable sophisticated, context-aware interactions with our consultancy knowledge base. These purpose-built tools provide significant advantages over custom implementations through their deep integration with Neo4j's architecture, ensuring optimal performance and semantic alignment with our graph structure shown in Figure 3.

The MCP toolset accelerates agent development by providing pre-optimized connectors for:

- Natural language to Cypher query translation
- Context-aware relationship traversal
- Schema-aware result interpretation
- Automated query optimization

Unlike single-attempt query systems, our agent explores the graph iteratively, similar to how a human would reason. When presented with a business question (e.g., "Find consultants experienced in healthcare digital transformation"), the agent:

1. Formulates an initial Cypher query based on the question context
2. Analyzes result quality and completeness
3. Dynamically refines the query based on graph patterns discovered
4. Expands or narrows the search scope through relationship traversal
5. Repeats until achieving confidence in the solution

This iterative approach lets the agent follow complex relationship chains that a single query would miss. For example, when searching for relevant past proposals, the agent might:

1. First locate a similar Context in past missions
2. Then trace `ANALYSED_BY -> Consultant -> CONTRIBUTES_TO` for all the Proposals for that context
3. Follow `FULLFILLS` connections to understand the Deliverables for that Prosposal

The MCP integration ensures each exploration step benefits from Neo4j's native optimizations for:

- Relationship indexing
- Pathfinding efficiency
- Community detection
- Real-time pattern matching

Moreover, the agent's reasoning process is fully transparent, every iteration, query modification, and result evaluation is logged and explainable. This traceability

is crucial for consultancy contexts where clients require evidence-based recommendations. By combining Neo4j's specialised tooling with iterative exploration, the agent delivers contextual insights that would be unattainable through either manual querying or conventional one-shot AI approaches.

4.6 Generative company profile Component

The *Generative Company Profile Component* is responsible for synthesising strategic, up-to-date company intelligence into consultant-ready profiles. These profiles support tasks such as proposal writing, opportunity detection, or client engagement preparation. Figure 5 illustrates the overall architecture.

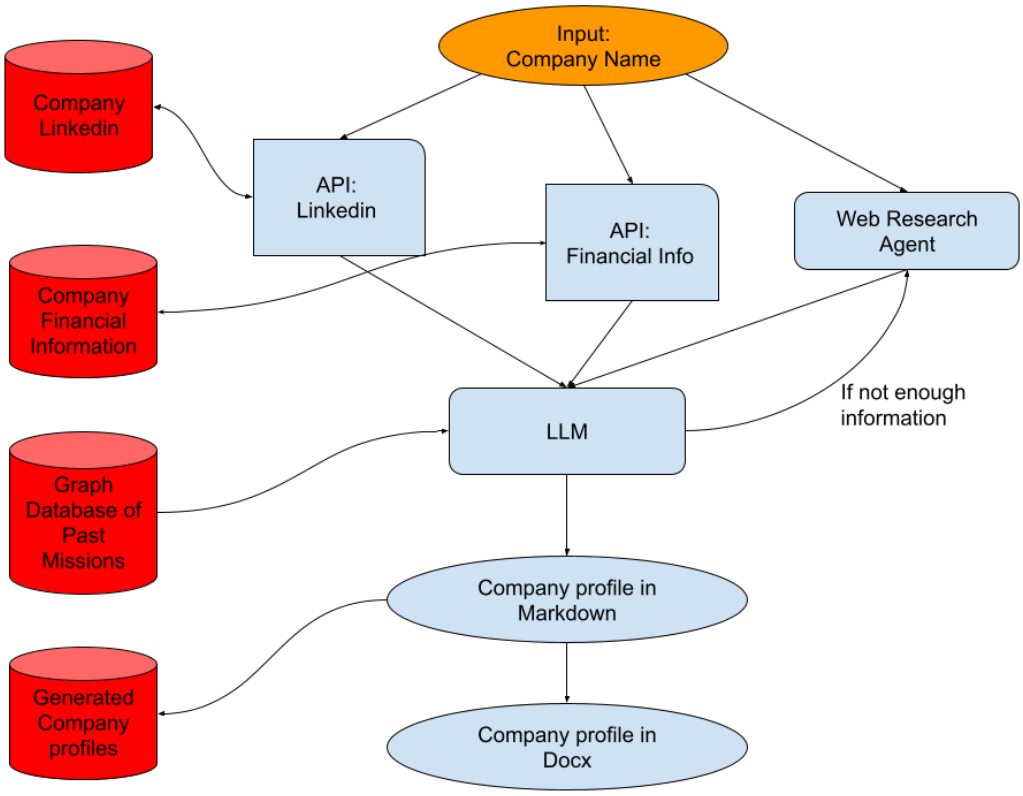

Figure 5: Architecture of the Company profile workflow

- **Input: Company Name**
  The workflow starts from a simple input: the name of the company of interest. This serves as the query key for all subsequent data collection and generation steps.

- **API: LinkedIn (via Apollo.io)**
  This API retrieves both company-level and employee-level information. It collects data such as the number of employees, headquarters address, and industry classification, as well as a detailed list of executives including their names, roles, tenure, and LinkedIn profiles. This data is cached and refreshed monthly. If new executives join or existing ones leave, the system flags these changes and notifies the consultant.

- **API: Pappers (Financial Information)**
  The Pappers API returns structured financial and legal data, including registered address, yearly revenue, EBITDA, margin, and net profit across all available years. The financial data is stored in a database and updated every six months to ensure reliability and compliance.

- **Web Research Agent**
  This agent performs live web queries to gather the latest strategic context. It retrieves the top three most recent news articles about the company, performs market research to identify major competitors, and highlights emerging challenges. Additionally, it supports the LLM by fulfilling targeted research requests whenever the model identifies gaps (e.g., incomplete SWOT sections).

- **Graph Database of Past Missions**
  This module contains a structured history of past consulting missions related to the company or its peers. These missions are injected into the LLM as variables and are also used directly in the generated SWOT analysis and business context sections to reinforce insight relevance and historical grounding.

- **LLM**
  The large language model acts as the central intelligence of this workflow, ingesting all structured and unstructured information collected by the other components. Its role goes beyond simple summarization as it actively participates in the research process. After generating an initial SWOT analysis, the LLM assesses its own output for completeness. If it identifies a knowledge gap it autonomously formulates a query and delegates a targeted research task to the Web Research Agent to fill that specific gap. This creates a dynamic, self-correcting loop where the system actively refines its own knowledge base. This agentic behavior is a powerful feature, ensuring

the final company profile is not just a summary of pre-collected data, but a comprehensively researched and reasoned document.

- **Company Profile in Markdown (Machine-readable)**
  The output is first structured in a markdown format which is valid for one week. This version is designed to be machine-readable and compatible with orchestration systems or downstream agents that automate further tasks such as proposal drafting or competitive benchmarking.
- **Company Profile in DOCX (Consultant-Ready)**
  The final version is also rendered in DOCX format for direct use by consultants. This format is preferred for its readability and ease of integration into existing proposal decks or documentation.

The generated company profile follows a consistent structure:

1. Financial Information
2. Latest News and Market Insights
3. SWOT Analysis
4. List of Past Missions
5. LinkedIn Links and Titles of Top Executives
6. Business Context Summary

4.7 Mission Proposal Component

The Mission Proposal component is responsible for generating structured consulting proposals based on a specific company context. Its goal is to automatically draft high-quality, consultant-ready documents by leveraging past missions, reusable knowledge, and iterative human feedback.

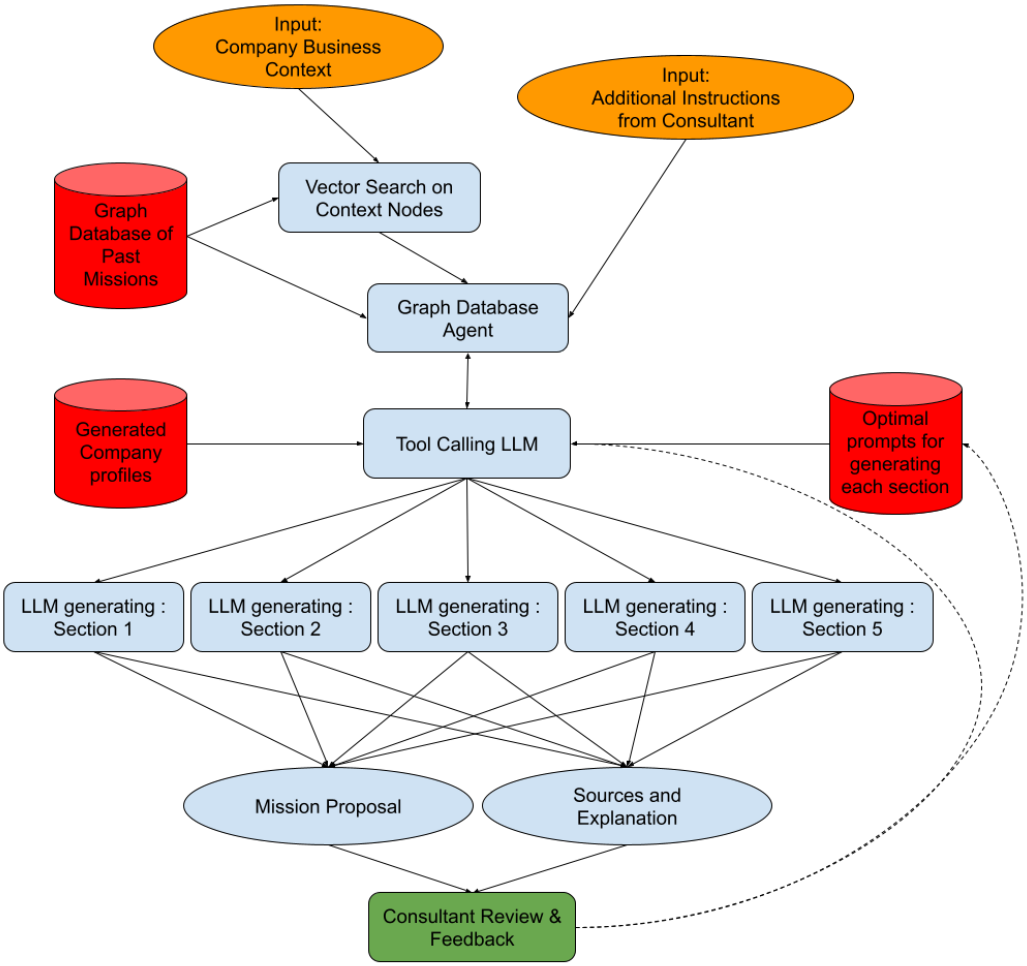

Figure 6: Architecture of the Mission Proposal workflow

- **Inputs: Company Business Context and Consultant Information**
  The process begins with a description of the company's current situation and the strategic challenges it is facing. This input can either be written manually by the consultant or automatically extracted from the structured company profile (as described in figure 5). Additionally, the consultant may contribute further insight like a preliminary hypotheses or known constraints which helps steer the proposal generation toward their intended direction. This human input complements the AI-driven retrieval of past data and ensures alignment with the consultant's business intuition.

- **Vector Search on Context Nodes**
  Each context node stored in the knowledge graph is embedded into a vector space using sentence embeddings. We then leverage Neo4j's vector indexing

to retrieve the top three most semantically relevant past mission contexts to the context given in input. These serve as starting points for deeper graph exploration and proposal generation.

- **Graph Database Agent**
  As detailed in Section 4.5, the Graph Database Agent begins by traversing the knowledge graph from the most relevant mission contexts, as identified via vector search. It collects detailed, structured information associated with these contexts, deliverables, teams, and strategic approaches. Multiple proposals may be associated with a single context node, offering a variety of solution formats.

  In addition, the agent checks whether the selected contexts belong to the same strategic cluster or community. These communities are discovered using the Leiden algorithm (see Section 4.2) and can be used to filter information, by staying within a coherent community, you avoid pulling in misleading or irrelevant cases.

- **Tool Calling LLM**
  We decided to generate each section of the proposal separately to avoid an overly complex prompt, a huge context window and misuse of given information. We equip an LLM with the ability to call another LLM and also communicate with the Graph Database Agent, it is task to determine if it has enough information from the knowledge and to segment that information into what is necessary for the generation of each section. For each section we pre-defined the optimal prompt for generating the section in question whilst following the internal guidelines of the consulting company. Altough the LLM has access to tools it does not have the required agency to be called an agent. Here are the sections that make up a standard proposal:

  1. **Context and Challenges**: A summary of the business environment and strategic challenges of the client.
  2. **Our Convictions**: Strategic beliefs derived from past missions and domain expertise.
  3. **Our Approach**: A breakdown of the consulting strategy, organized by phases (e.g., diagnostic, recommendations, implementation support), with justification for each step.
  4. **Deliverables**: A clear list of outputs the client can expect, including their form and business value.

5. **Planning, Team and Budget**: Timeline of the engagement, involved consultants and their roles, and budget breakdown if applicable.

- **Human Feedback Loop**
  Two documents are generated at the end of the proposal generation cycle:
  - **Draft Proposal**: This is reviewed by a consultant, who provides feedback on each section. If the consultant identifies missing or weakly supported sections, the LLM is prompted again, this time with the feedback integrated into the prompt. This feedback is also stored as training data to fine-tune the prompt templates and improve the system's ability.
  - **Sources and explanation**: A companion document is automatically created, detailing which pieces of information were used to generate each section of the proposal, along with the reasoning paths taken. Each insight is linked to its source ensuring transparency and explainability. Consultants can refer to this document to verify claims or enrich their own understanding of how recommendations were formed.

5 Implementation

5.1 Technology Stack

Core Technologies: The system is built in Python, with agent orchestration and memory management handled by LangChain. We chose LangChain over raw LLM API usage due to its support for modular agents, persistent memory, and seamless integration with external tools. Crucially, the system is designed to be model-agnostic. While GPT-4-turbo is currently used for its reasoning and generative capabilities, the architecture permits any comparable LLM to be substituted with minimal friction. This flexibility is a core design principle, allowing the system to adapt to the rapidly evolving AI landscape and take advantage of more efficient and smarter LLMs as new models become available. To ensure quality is maintained after any model change, the COMET evaluation framework (see Section 5.3) will be used to perform rigorous testing and performance benchmarking, guaranteeing the output remains aligned with the firm's quality standards.

We also leverage LangGraph for agent-to-agent communication (A2A), enabling agents to share memory and coordinate sub-tasks dynamically. Dynamic webpage rendering and scraping are handled through a combination of Playwright and Selenium—Playwright ensures complete rendering of JavaScript-heavy pages, while Selenium modules parse and extract structured content for further processing by Web Agent.

External APIs:

- **Apollo.io API:** Retrieves company metadata (e.g., industry, size, headquarters) and key executive information (e.g., names, roles, LinkedIn URLs). This data is refreshed monthly, with change detection mechanisms that notify consultants of executive turnover.
- **Papers API:** Provides financial performance indicators such as revenue, EBITDA margin, and net profit. These are updated biannually or manually. Although using this platform guarantees reliable, structured financial information, it has the disadvantage of only working for companies registered in France (with a SIREN number). Additionally, very large groups may be divided into many smaller entities, making it difficult to obtain

a comprehensive overview of the group as a whole, which is our primary objective.

- **Bing Web Search API:** Powers the Web Agent by retrieving relevant online content related to client context, industry shifts, or competitor movements.

5.2 Data Flows and Pipelines

Knowledge Graph Construction:
The knowledge graph is constructed by ingesting structured information from historical mission documents and company data.

Consultant activity (e.g., mission names, client associations) is extracted directly from the internal timesheet software *Silae*, where consultants record time spent on specific missions. More specific data related to a mission, such as context, deliverables, and convictions, is extracted from the final proposal documents. As described in Section 4.7, these proposals follow a predictable format, enabling reliable automated parsing. All extracted information stored in the nodes of the Knowledge Graph is linked to the original document from which it was obtained, ensuring full traceability and source verification.

Information Refresh Rates:

- **Company Profiles:** Generated weekly for all clients. Each regeneration fetches the latest news, a new SWOT analysis, and business context. The other data elements, i.e., the financial data and executive employee information, are not re-fetched unless explicitly requested.
- **Financial Data:** Fetched from the Papers API every six months. Manual refreshes are also supported.
- **Executive Team Data:** Refreshed monthly via Apollo. If changes are detected (e.g., new hires or departures), the new profiles are highlighted in the newest company report. Apollo's filtering capabilities allow us to retrieve only high-level executives by specifying seniority levels, ensuring we focus on the most relevant individuals.

Prototype Data Format:
Currently, all data is stored in JSON format within the project repository. As the system scales, we plan to migrate to production-grade database infrastructure to ensure performance, consistency, and data versioning.

Figure 7 presents the planned database schema. The central table contains a list of all `company_ids` that are present in the Knowledge Graph and serves as the primary key for linking to related data, this schema provides a solid foundation for scalability and facilitates the integration of additional databases as the system continues to evolve.

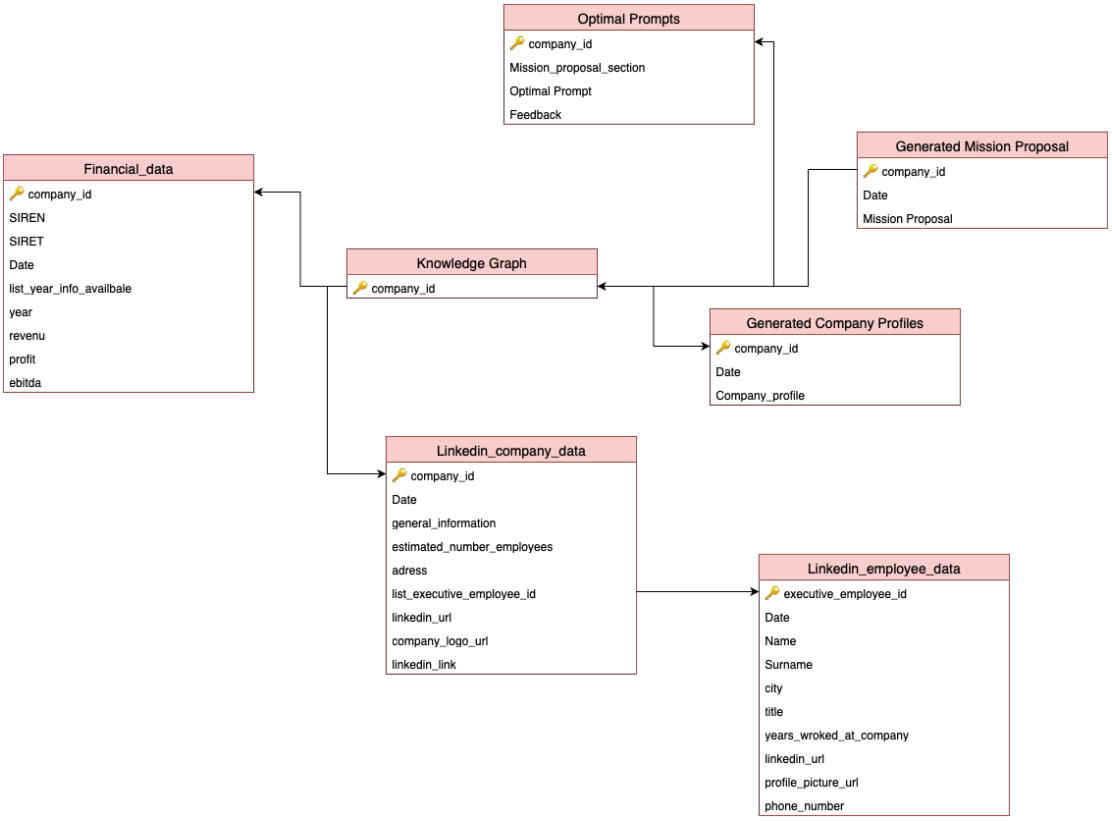

Figure 7: Database Schema

5.3 Proposed Evaluation Framework and Preliminary Analysis

To rigorously test our hypothesis in the future, we have developed a comprehensive evaluation framework using Opik COMET. While the system remains in a prototype phase, precluding a full-scale empirical study at this time, this framework

outlines the necessary methodology. We used a component of this framework to conduct a preliminary, simulation-based analysis to estimate potential efficiency gains.

To operationalize this framework, a robust evaluation test suite will be constructed from two primary sources. First, a diverse corpus of historical proposal documents from various business domains will be used to create "golden datasets." These test cases will evaluate the system's ability to accurately retrieve and synthesize information grounded in real-world source material. Second, a set of standardized, representative use cases will be developed to test the system's performance on more general tasks, ensuring its versatility and ability to handle common consultant queries effectively. This combination of real and synthetic data will allow for a comprehensive assessment of the system's accuracy, relevance, and overall utility.

Evaluation Metrics: We use COMET to run controlled test cases with predefined inputs, expected outputs, and reference contexts. It allows us to evaluate the MAS on the following metrics:

- **Usefulness Metric:** Measures whether the final output answers the user's intent.
- **Context Precision:** Assesses whether the MAS used relevant context nodes in its reasoning.
- **Context Recall:** Measures whether any key contextual elements were missed.
- **Hallucination Metric:** Evaluates whether the LLM generated information is not grounded in any retrieved or validated source.
- **Moderation Metric:** Judges tone, sensitivity, and appropriateness of the LLM output, using an LLM as evaluator.

This set of metrics provides a holistic view of the system's performance, allowing us to directly assess its ability to reduce proposal time while maintaining the high quality required in a professional consulting context. The results of this evaluation are presented in Section 6.2

5.4 Scaling Design

Scalability Roadmap:

- **Cloud-Ready Deployment:** Containerize all agents and services for deployment on cloud platforms (e.g., GCP, AWS, Azure).
- **Database Integration:** Replace local JSON storage with scalable databases.
- **Asynchronous Agent Pools:** Use message queues (e.g., RabbitMQ or Celery) to coordinate asynchronous workflows.
- **Query Caching:** Store results of high-frequency queries to reduce external API usage and speed up response time.
- **Frontend Consolidation:** Move away from independent Streamlit instances toward a centralized web interface with shared backend state.

6 Projected Impact and Workflow Simulation

6.1 Simulation Scenario

To demonstrate the system's capabilities, we walk through a realistic scenario involving a consultant preparing for a client meeting.

Suppose a consultant is scheduled to meet a contact at a company, this could result from a prior networking event, a referral, or a cold outreach that led to expressed interest. Before the meeting, the consultant uses the system to generate a comprehensive Company Profile. This document provides a deep overview of the company's situation, including its business context, latest news, SWOT analysis, and key financial and strategic indicators. 8

Upon reviewing the profile, the consultant notices that their firm has previously worked with this company. They ask the agent for details about past missions. The orchestrating agent forwards the request to the Knowledge Graph Agent, which queries the graph for all relevant information about past missions. 9

If the consultant is attending an in-person meeting or networking event, the inclusion of executive photos and job titles in the Company Profile becomes particularly useful. It provides a clear organizational snapshot, allowing the consultant to recognize individuals in person and tailor interactions based on their role (e.g., CTO for technical topics, CMO for marketing challenges).

Depending on the case, the consultant might also ask the agent to find similar past missions that address challenges like those the potential client is currently facing. The agent performs a semantic similarity search and identifies the three most relevant cases, complete with methodologies and outcomes. This allows the consultant to walk into the meeting equipped with concrete examples and immediate value propositions.

After the meeting, having validated the client's challenges and potential solutions, the consultant initiates the Mission Proposal generation workflow. The system draws from past missions and consultant input to draft a tailored proposal. The consultant can review, edit, and finalize the document, using embedded links to revisit the source context if needed.

If they require additional insight, the consultant can ask the agent who led a specific past mission or who might have domain expertise on a given subject. Once finalized, the proposal can be exported either as a Word document (automatically generated and manually refined), or used as the basis for a PowerPoint deck (which can be built quickly thanks to standardized templates).

Feedback on the generated proposal,such as gaps in project planning, tone issues, or missing business impact statements, is logged for continuous improvement. This feedback loop helps refine agent performance and improves future generations.

Comparison to Traditional Workflow:

In a typical workflow without this system, the consultant would attend the initial meeting with only a vague understanding of the company's challenges. Several follow-up meetings would likely be needed just to grasp the full context and have an idea of a solution. Proposal drafting would then start from scratch, often by a junior consultant unfamiliar with similar past work, resulting in longer turnaround times and potentially lower quality.

In contrast, this system significantly reduces the time between opportunity identification and proposal delivery. By equipping consultants with high-context insights before and during client interactions, it creates a strategic advantage: faster response time, higher relevance, and allow the consultant to focus on client interaction, giving the firm an edge over competitors who cannot act as quickly or insightfully.

When consultants enter the first meeting with a deep understanding of the client and their context, they can ask more targeted questions, establish credibility more quickly, and demonstrate a level of preparedness that often sets their firm apart in competitive situations where trust and insight are key.

6.2 Projected Time Savings and Qualitative Potential

Given the prototype stage of the system, a full-scale experiment with multiple users was not feasible. Instead, to estimate the potential impact on efficiency, we conducted a simulation of the proposal workflow. The time required for each stage was estimated based on the prototype's current capabilities versus historical

averages for the manual process.

Estimated Quantitative Impact: Time Reduction

The table below presents a comparative estimate of the time required for key stages of the proposal process. The 'Manual Workflow' times are based on firm averages, while the 'Augmented Workflow' times are estimates derived from the prototype's automated capabilities.

As this simulation suggests, the system has the potential to reduce the total time required for proposal generation by approximately 62%. The largest efficiency gains come from internal knowledge discovery and proposal content assembly, both of which are time-consuming and highly repetitive in the manual workflow. These tasks are well-suited to automation through specialised agents connected to a structured knowledge graph. Importantly, the review and iteration phase remains human-led, with the system accelerating draft preparation but preserving consultant control over final content. Further testing with real users would be necessary to validate these projections in practice.

The outputs generated by the system are designed to follow the consulting firm's visual identity and communication standards. As shown in Figure 8, the Company Profile includes the logo and name of the target company, alongside a consistent design language. While it may not be immediately visible in the Word document format, the profile contains embedded hyperlinks throughout. For instance, financial data points link directly to the source (e.g., Paper's website), and each point in the SWOT analysis links to supporting context or news.

Figure 9 demonstrates an interaction with the orchestrating agent through the chat interface. Users are not required to know exactly what they're looking for, instead, they can ask open-ended or exploratory questions and refine their queries iteratively. In this example, the user starts by asking about past missions conducted for a specific client, they can then follow up with more specific questions related to a particular deliverable or strategic conviction. The MAS allows users to ask follow-up questions and explore information step by step, without needing a predefined query

On the same platform, a second interface is dedicated to Company Reports. As

Stage	Description	Manual (hrs)	Augmented (hrs)
Company Profile Drafting	Writing a tailored company overview, with key facts and positioning.	3.0	0.5
Opportunity Understanding	Researching the client, their market, and the broader mission context.	3.0	1.0
Internal Knowledge Discovery	Searching internal documents, past proposals, and relevant expertise.	4.0	0.5
Proposal Content Assembly	Assembling methodology, deliverables, timeline, and team credentials.	6.0	3.0
Formatting & Packaging	Structuring the document or slides, applying templates and visuals.	3.0	2
Review & Iteration & Deciding Budget	Iterative feedback, edits, and alignment with partner input.	5.0	2.0
Total		**24.0**	**9.0**

Table 1: Estimated Time Comparison and Stage Descriptions for Proposal Generation

shown in Figure 10, users can select a specific week to review reports for, which is particularly useful for tracking the evolution of a company's business context over time. Reports are searchable by company name, and users can also browse them by scrolling through thumbnails that preview the company name and its business context. This interface is designed to help consultants quickly spot potential client leads or revisit evolving situations. For convenience, each company profile

1. CHANEL

1.1. Description de la société

CHANEL is a French luxury fashion house specializing in haute couture, ready-to-wear clothes, luxury goods, and fashion accessories. Its main activities include designing and selling high-end clothing, handbags, perfumes, cosmetics, and watches.

1.2. Donnée financière de l'entité suivante: CHANEL / Siren: 542052766 (source)

Année	Chiffre d'affaires	EBITDA	Marge EBITDA	Marge opération nelle	Résultat net	Croissance CA
2023	4.0 milliards €	1.5 milliards €	38.3%	35.0%	1.7 milliards €	16.3%
2022	3.5 milliards €	1.5 milliards €	42.0%	40.4%	1.0 milliards €	29.0%
2021	2.7 milliards €	869.5 millions €	32.4%	31.6%	816.5 millions €	25.3%
2020	2.1 milliards €	477.2 millions €	22.3%	15.0%	177.6 millions €	-30.0%
2019	3.1 milliards €	966.3 millions €	31.5%	28.9%	877.9 millions €	11.2%
2018	2.8 milliards €	843.6 millions €	30.6%	30.3%	606.8 millions €	2.9%
2017	2.7 milliards €	806.3 millions €	30.1%	28.8%	585.2 millions €	8.9%
2016	2.5 milliards €	686.0 millions €	27.9%	25.7%	504.7 millions €	-5.0%
2015	2.6 milliards €	753.1 millions €	29.1%	27.4%	836.1 millions €	N/A

1.3. Informations sur l'entreprise

Address : 5 Barlow Place, London, W1J 6DG, United Kingdom

Nombre d'employés estimés : 32,000

1.4. Analyse SWOT

FORCES (Facteurs internes positifs)

- **Position de leader sur le marché** : CHANEL est reconnu comme la marque de luxe la plus précieuse, avec une valorisation estimée à 38 milliards de dollars en 2025, consolidant sa position dominante dans l'industrie du luxe source: Chanel, The Most Valuable Luxury Brand According To ….
- **Fidélité des clients** : La marque a maintenu la première place dans l'indice de sentiment des consommateurs de Vogue Business, soulignant une forte fidélité et une perception positive de la marque source: What consumers want from luxury in 2025.

DOCUMENT DE TRAVAIL - STRICTEMENT CONFIDENTIEL 1
COMPANY PROFILE : CHANEL - EDITE LE 01/07/25.

Figure 8: Example of the first page of a Company Profile for Chanel

can either be viewed directly in the browser for quick exploration or downloaded as a Word document for offline reading or sharing.

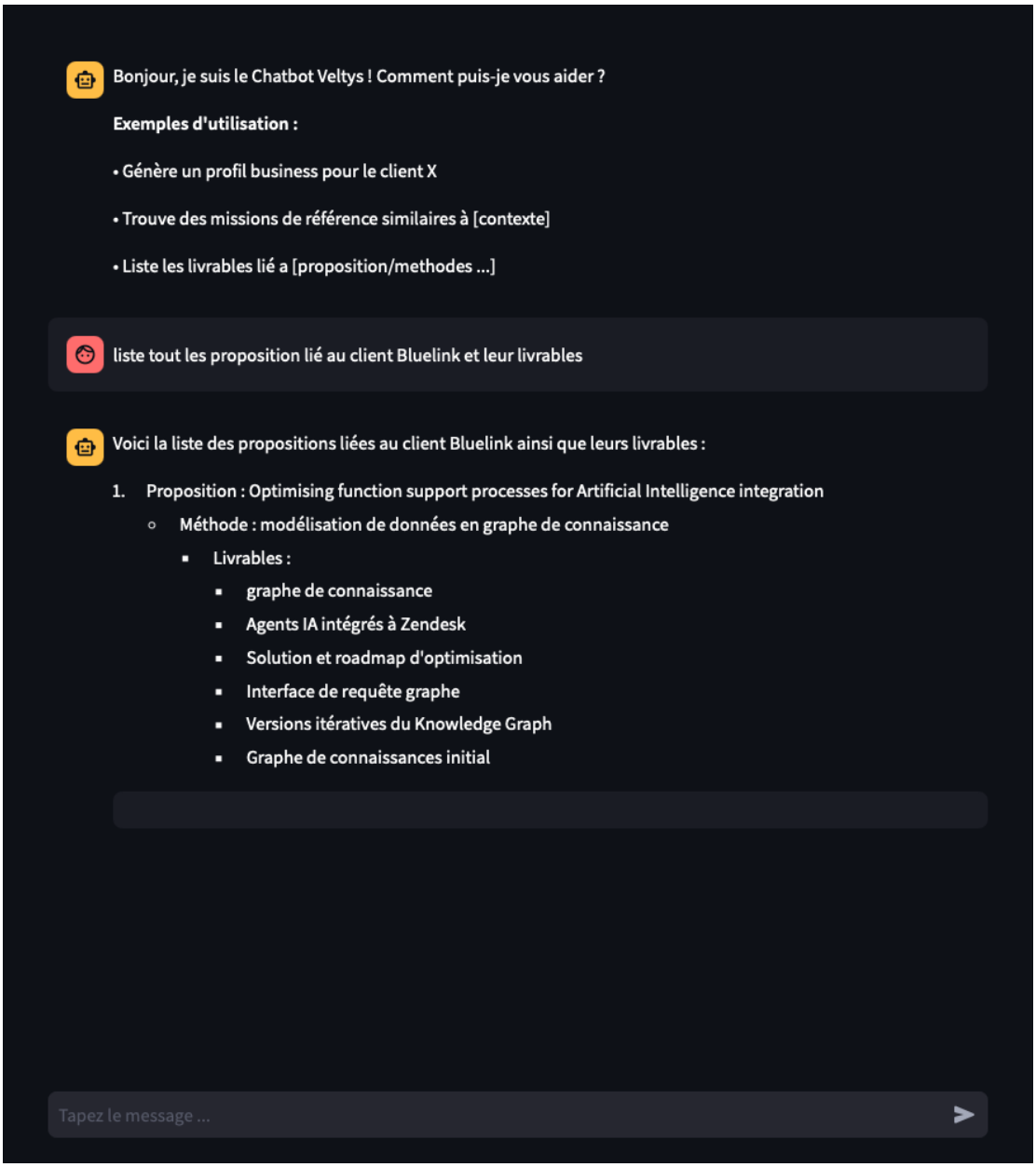

Figure 9: Example of an interaction with the MAS, querying past missions related to a client

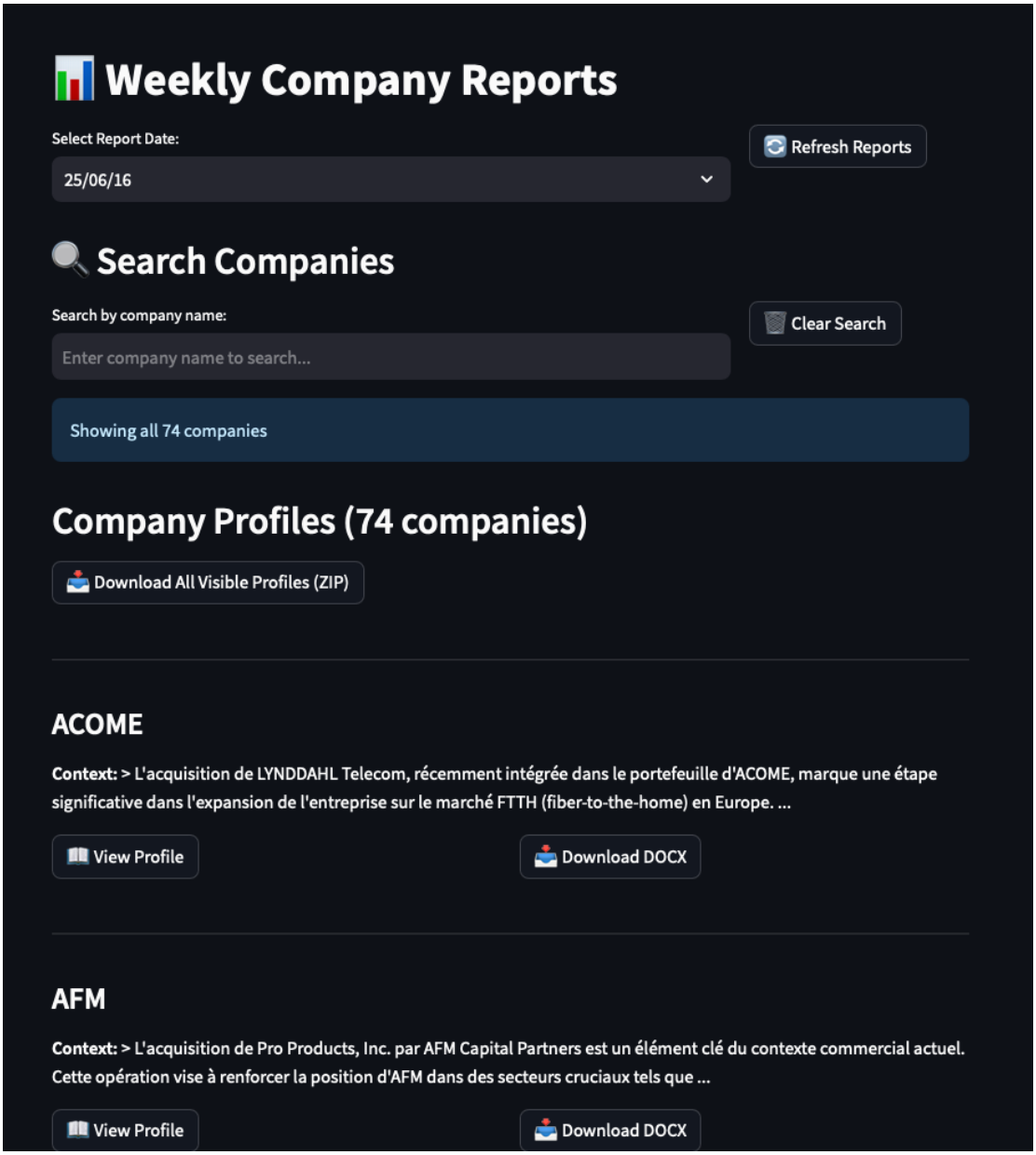

Figure 10: Interface displaying searchable and downloadable company profiles by week

7 Discussion

7.1 Limitations of the Study

The foremost limitation of this research is the lack of rigorous, large-scale empirical validation. The system was evaluated at the proof-of-concept stage, and the quantitative and qualitative results presented are therefore preliminary and simulation-based. The estimated time savings and quality assessments require validation through controlled experiments with consulting staff working on real proposals. Time constraints of the project prevented this full-scale study from being conducted.

7.2 Current Constraints

Despite promising progress, several technical and practical limitations remain. First, reasoning across multiple steps,especially in complex tasks, can suffer from *chain-of-thought drift*, where the AI agent loses coherence or fails to maintain strategic context over time. This challenge becomes more acute as workflows span multiple documents, queries, and decision points.

Another key constraint is the necessity to develop workflows for core use cases. Creating these workflows requires domain expertise; individuals must possess a clear understanding of what the workflow aims to achieve. Currently, a reasoning agent lacks the sophistication to independently manage this task

Reliable web search also remains a bottleneck. While AI agents can access the open web, filtering for trustworthy, up-to-date, and relevant sources is non-trivial. Integrating curated knowledge bases or vetted APIs may help mitigate this issue but adds implementation complexity.

Finally, data privacy poses a significant barrier to large-scale deployment. Consultancy work often involves sensitive client data, requiring strict controls around access, storage, and model inference.

7.3 Implications for Human-AI Collaboration

AI systems that handle repetitive and administrative work can free up humans to focus on high-impact tasks like strategic thinking, client engagement, and creative problem solving. When human users can speak to an AI agent to iteratively define an idea, much like thinking out loud, they externalize and refine their thought processes. This can help overcome the *"curse of knowledge"*, where individuals possess insights but struggle to recall or articulate them without prompting. In this way, AI becomes not just a tool for automation but a cognitive partner that enhances human ideation.

However, these benefits must be weighed against the potential downsides, such as the risk of over-reliance and complacency identified in recent research [3]. Our system design directly anticipates and mitigates these risks. For instance, the system is designed to prevent complacency by generating a companion **"Sources and explanation" document** for every proposal. This document, which traces every claim back to its source data, forces the consultant to verify the AI's reasoning, promoting critical engagement rather than passive acceptance.

Furthermore, the system is designed to prevent the "dulling" of strategic insight. The iterative human feedback loop ensures that the consultant's unique expertise is not replaced, but augmented. The AI generates a comprehensive and data-grounded first draft, but the consultant drives the final narrative, refining sections and injecting the strategic nuance that differentiates the firm's value. This design combines the machine's efficiency with the human expert's judgment. Human oversight is not just an option it is at the core of the system.

7.4 Potential Improvements

Looking forward, several extensions could further enhance the system's capabilities. For instance, enabling direct generation of PowerPoint slides from AI-generated content would streamline the production of client deliverables and reduce manual formatting overhead.

The existing organizational learning loop could be formalized by developing a dedicated **Prompt Management and Optimization Tool**. Currently, feedback is stored to help fine-tune templates, but a formal tool could significantly accelerate

this process. Such a tool could allow consultants to prodvide structured ratings on generated sections, enabling the system to automatically A/B test prompt variations and ientify which ones consistently produce higher-quality outputs with less need for human editing. This would create a data-driven approach to prompt engineering, treating the firm's core prompts as valuable intellectual assets with version control and performance analytics. This would not only enhance the quality of proposals but also provide clear insights into which strategic approaches are most effective, turning consultant feedback into a measurable driver of firm-wide improvement

Another opportunity lies in proactive lead generation. By continuously scanning news, industry updates, and financial reports, the agent could perform automated SWOT analyses on key clients and rank companies most likely to require consulting support. This could help consultants identify opportunities early and tailor outreach accordingly.

Finally, consulting is a relationship-driven business, and incorporating a social graph of past and current contacts could add significant value. Tracking employee movements across companies, identifying shared connections between consultants and clients, or receiving alerts when a contact changes roles could all support more strategic and personalized engagement. A graph-based system could surface hidden opportunities, improve targeting, and strengthen client relationships by tapping into the firm's collective network intelligence.

8 Conclusion and Future Work

8.1 Summary of Contributions

This research set out to investigate the potential for a hierarchical multi-agent system to improve the efficiency and quality of proposal generation in consulting. The primary contribution of this paper is the design, development, and justification of a new proof-of-concept system architecture. While a full empirical validation was beyond the scope of this project, a preliminary analysis indicates the system holds significant potential to reduce proposal generation time.

This project developed a practical and domain-aware AI system tailored to the needs of consultants. Unlike generic chatbots, which lack context and often require manual prompting, the system combines a structured knowledge graph with a hierarchy of specialised agents that follow real consulting workflows, such as drafting company profiles or mission proposals. This alignment with actual business processes is what makes it valuable in practice.

The system solves one of the bigger bottlenecks in consulting: fragmented knowledge. By connecting internal documents and past mission reports into a navigable, queryable graph, the platform allows consultants to easily find and reuse firm knowledge that would otherwise remain buried in archives. This not only saves time, but directly improves the quality and relevance of proposals.

The hierarchical multi-agent system includes specialised AI agents such as a web search agent and a graph-querying agent that collaborate to deliver high-quality outputs with minimal friction. Importantly, the system does not attempt to replace the consultant. Instead, it augments their work by handling the research, structuring, and synthesis steps, while keeping the human firmly in control.

Moreover, consulting is a high-turnover industry where valuable human knowledge is often lost when employees leave. The system directly mitigates this by functioning as a persistent, collective knowledge base. By capturing the context, methodologies, and deliverables of past missions in the knowledge graph, the firm retains critical institutional memory that transcends individual consultants. This ensures that expertise remains accessible to the entire team, reducing the impact of turnover and helping new consultants become productive more quickly.

Each proposal is generated through a modular, editable workflow: consultants can review, re-prompt, and refine any section at any point, creating a fast feedback loop that mirrors natural human iteration. This preserves strategic nuance, ensures contextual accuracy, and accelerates proposal development without sacrificing quality.

In short, the project delivers an integrated tool that fixes knowledge access, encodes domain workflows, and enhances human performance in the day-to-day realities of consulting work.

8.2 Broader Impacts

While this project focuses on the proposal phase, its implications extend much further. When proposals can be created and refined in days rather than hours, firms can respond to opportunities more quickly, engage clients sooner, and compete on both quality and speed. This favors consultants who can communicate effectively, think strategically, and iterate rapidly.

Moreover, the architecture of the multi-agent system developed for this project is inherently extensible. As AI is increasingly integrated into every tool, connecting all of them to a central orchestrating AI agent, this means the use of highly automated workflows like the ones could be applied beyond proposals to support day-to-day consulting tasks. Each consultant could work alongside a personal AI assistant that understands their context, remembers past engagements, and proactively helps manage information, tools, and deliverables. Rather than relying on disparate applications and manual work, consultants would be supported by an integrated, intelligent system that removes friction from their workflow.

Ultimately, this shift signals a broader redefinition of the consultant's role. As AI systems take over the "button-pressing" and formatting tasks, human consultants can focus on strategic thinking, creative problem-solving, and client relationships. Instead of spending time assembling information, they will spend time making sense of it. Rather than reacting to client needs, they can anticipate them. The result is not a diminished role for the consultant, but an elevated one, freed from routine labor and empowered by intelligent tools that extend their capabilities.

This shift is not unique to consulting; similar systems could be applied across

many professional fields where knowledge work involves repetitive tasks, scattered information, and the need for faster, better-informed decisions.

References

[1] Akira AI. Ai compliance monitoring and risk detection. https://www.akira.ai, 2024.

[2] Anthropic. Model context protocol (mcp). Open source specification, 2024.

[3] Andrew R. Chow. ChatGPT may be eroding critical thinking skills, according to a new MIT study, 2025.

[4] Pallavi Dadhich. Single agent vs multi-agent ai: Which one to choose in 2025? https://www.experro.com/blog/single-agent-vs-multi-agent-ai/, 2025.

[5] Mingyan Gao, Yanzi Li, Banruo Liu, et al. Single-agent or multi-agent systems? why not both? *arXiv preprint arXiv:2505.18286*, 2025.

[6] Nicholas R Jennings. On agent-based software engineering. *Artificial intelligence*, 117(2):277–296, 2000.

[7] Loopio. Rfp statistics: 2025 benchmarks to know. https://loopio.com/blog/rfp-statistics-win-rates/, 2025. Accessed June 2025.

[8] Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *SSRN Electronic Journal*, 2023.

[9] Hyacinth S Nwana. Software agents: An overview. *The Knowledge Engineering Review*, 11(3):205–244, 1996.

[10] OneReach.ai. How ai is transforming legal and regulatory workflows. https://onereach.ai/blog/ai-agents-in-legal-services-automating-review-and-contracts/#:~:text=contract%20data%20and%20consumption%20patterns,missed%20opportunities%20and%20revenue%20leakage, 2024.

[11] Relevance AI. Multi-agent systems: A new paradigm for business ai, 2024.

[12] Responsive.io. Scaling proposal management success with limited headcount, 2024.

[13] Salesforce. AI in healthcare: Doctors say it could save them a day a week, 2024.

[14] Smythos. How ai tutors are shaping the future of education. https://smythos.com/developers/agent-development/multi-agent-systems-in-education/#:~:text=Human%20teachers%20remain%20essential,solving%2C%20and%20nurturing%20students%20curiosity, 2024.

[15] Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

[16] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2nd edition, 2009.

[17] Walden Yan. Don't build multi-agent systems. https://maxpavlov.medium.com/single-vs-multi-agent-systems-when-to-build-which-1f336c676bd7, 2025.

[18] YourStory. Ai in consulting: Inside mckinsey's lilli and deloitte's sidekick. https://yourstory.com/2025/06/consulting-firms-ai-tools-mckinsey-bcg-deloitte-2025, 2024.