


Scaling AI adoption in finance: modelling framework and implementation study

Thomas Sepanosian ¹[0009-0001-7342-3798], Zoran Milosevic²[0000-0002-1364-7423], and Andrew Blair³[0009-0000-0254-5061]

¹ University of Twente, Enschede, the Netherlands,
`t.sepanosian@student.utwente.nl`

² Deontik, Brisbane, Australia, `zoran@deontik.com`

³ Westpac, Sydney, Australia, `andrew.blair@westpac.com.au`

Abstract. There is an increasing potential for using AI applications in finance, ranging from simpler Generative AI applications to more complex, agent oriented solutions. This paper reports on our experience in applying early AI solutions in an Australian fintech landscape. We first present a framework developed to support industry experts and practitioners in adopting AI solutions in a scaleable manner, to ensure the adoption of fit-for-purpose AI systems. We then focus on a longer term research dimension, which addresses more complex business problems for which the emerging multi-agent AI technologies may offer more value. We experimented with these technologies, including their integration with more mature approaches such as RAG. Our proof of concept for retirement planning application, highlights benefits and directions for LLM-powered AI agents, and also identifies limitations of current technologies. Specifically, deploying multi-agent technologies on low-powered infrastructure presents challenges. These limitations can hinder the implementation of solutions that require reliable reasoning and collaboration. Our proof of concept highlights both the potential of multi-agent technologies, and the limitations that need to be addressed.

Keywords: LLM powered agents · Multi-agent AI · RAG · Fintech.

1 Introduction

The financial services industry is undergoing a radical transformation fueled by Artificial Intelligence (AI). From fraud detection to algorithmic trading, AI is automating tasks, improving efficiency, and generating valuable insights. For example, David Walker, the Chief Technology Officer at Westpac, one of Australia’s leading banks, predicts that these advancements will help revolutionize the banking industry [8]. His insights highlight the potential for AI to provide personalized interactions and recommendations through context-aware systems, potentially enhancing engagement with both customers and employees.

This paper addresses the need for practical AI applications in finance, specifically wealth management tools like retirement planning. We aimed to understand the value of current AI capabilities while accommodating future developments.

There are two key contributions of this paper. The first is an industry-focused framework designed to support Australian financial services professionals in adopting AI solutions in a scalable manner. Starting from simpler, more mature AI technologies and evolving into complex systems capable of addressing advanced business challenges, this framework empowers industry experts and practitioners, such as risk managers, data scientists, and solution and enterprise architects to make informed decisions throughout the AI adoption process. It addresses key considerations such as consumer value proposition and regulatory compliance, ensuring the selection of fit-for-purpose AI systems.

Our second contribution delves into a longer-term research dimension, exploring the possibilities of emerging multi-agent AI systems, demonstrating their potential to deliver more efficient solutions for existing use cases and tackling more complex financial challenges. In particular, we utilized a multi-agent orchestration framework, crewAI [13], wherein agents are powered by Large Language Models (LLMs) and capable of utilizing techniques such as Retrieval-Augmented Generation (RAG), to develop a proof-of-concept retirement planning assistance application. This exploration not only highlighted the significant benefits of LLM-powered agents, such as improved productivity and personalization, but also revealed limitations related to explainability, consistency, biases and computational demands.

The remainder of this paper is organized as follows: Section 2 presents related work. This is followed by providing the description of the problem of scaleable adoption of AI within a finance organisation and a framework we developed to facilitate this, Section 3. Section 4 describes in detail our proof of concept implementation for a retirement application, utilizing the crewAI multi-agent orchestration framework. Section 5 discusses lessons learnt in developing this proof-of-concept. Finally, Section 6 concludes the paper and outlines future directions.

2 Related Work

There are a number of technological, regulatory and commercial efforts that influenced our work, which enabled us to develop a more focused solution approach while reflecting the specific business case and business environment we have addressed.

Our use case required specific access to relevant retirement information, consumer specific information, and regulatory information, such as relevant retirement policies, as a way of enhancing the semantic context and functionality of the LLMs powering the agents. We initially identified RAG to augment the LLMs' capability and provide contextually relevant and accurate information. By integrating RAG, LLMs harness the power of not only their pre-trained data, but also dynamically retrieved information, helping prevent hallucinations [10]. This comes with the added benefit of not having to retrain a model when new information becomes available. A more advanced, Agentic RAG extends these capabilities further by enabling LLMs to adjust retrieval strategies based on the

evolving context and goals within a conversation. By integrating Agentic RAG, these agents move beyond static information retrieval to a more proactive, agent-driven approach [23].

The term 'agentic' above reflects the ability of agents to exhibit their agency, in terms of their autonomy of decision making. These advanced AI-driven programs, designed to independently pursue defined goals, have the potential of helping industries including healthcare, finance and more [9]. With the emergence of LLMs such as OpenAI's GPT series and Meta's open-source Llama models, LLM powered agents specifically have become more prominent. These are agents where the core controller is an LLM, instead of classical techniques such as rule-based systems. Using an LLM enables the agent to have memory, the ability to plan to achieve its goals, and the ability to use tools, as opposed to merely performing an action [25]. The process of constructing autonomous AI agents involves the consideration of which architectures to use, to achieve optimal results [24].

Pivotal advancements in this area include Microsoft's AutoGen, which provides a framework wherein customizable agents can converse with each other, offering the opportunity for multi-agent LLM application development [26]. They propose that multi-agent conversations are feasible due to the abilities of LLMs to respond to feedback, handle a wide range of tasks, and to perform well on complex tasks by simplifying them into simple subtasks. Another recently emerged framework is crewAI [13]. Built on top of LangChain, the crewAI framework provides the capability of defining agents, tasks, and tools to rapidly compose a crew of agents which fulfill given tasks autonomously.

The increased interest in adopting AI within fintech organisations has also led to new regulatory efforts in Australia. The Australian Banking Association (ABA) has proposed several recommendations to the Australian government for integrating AI in the banking industry [1]. These include building upon existing sector-specific practices and maintaining legislative neutrality to ensure that AI-driven and human-made decisions adhere to the same regulatory standards. Furthermore, King & Wood Mallesons, a leading international law firm, acknowledges Australia's adoption of AI and its regulation, and highlights the importance of transparency, security, and associated risks when employing AI [16].

There are also some commercial providers that have recently starting offering Generative and Conversational AI solutions for the Fintech industries. Such solutions range from intelligent digital assistants, contact center support and generative AI, which were specifically developed for finance industry, based on curated financial knowledge sources [15,22].

Despite these advancements, challenges remain in the scalability and integration of LLM applications within several industries, such as the financial industry. This study aims to address these gaps by proposing a novel modeling framework and conducting an implementation study. Our approach seeks to enhance both the scalability and efficacy of AI in financial environments, thereby overcoming existing limitations and harnessing the full potential of autonomous agents.

3 Background and Problem

Scaling AI adoption within financial organizations presents significant hurdles. Implementing AI ethically and responsibly is paramount due to the potential for biased outcomes and reputational risks. Additionally, the multifaceted nature of financial challenges necessitates a diverse toolkit of AI techniques, from traditional statistical models to advanced deep learning. Finally, successfully integrating AI across an organization requires a strategic approach that builds upon foundational applications, such as automating routine tasks, before progressing to more complex endeavors like personalized financial advice.

To this end, we developed a comprehensive framework to guide financial institutions in their AI journey. The framework categorizes AI applications into distinct problem areas, so that organizations can tailor their AI strategies accordingly. This framework encompasses **intelligent information retrieval**, **content generation**, **data analysis**, **decision support**, and **task automation**. Progressing from foundational to advanced AI capabilities is crucial. For instance, **intelligent information retrieval** using techniques like natural language processing can support basic customer queries. Building upon this, **content generation** with AI can create tailored financial reports. More complex **data analysis** tasks, such as fraud detection, require sophisticated machine learning models and data science methodologies. As AI applications become more intricate, techniques like **RAG** (Retrieval-Augmented Generation) can enhance system performance by incorporating relevant information.

To address complex, multifaceted financial problems, **multi-agent systems** and **expert systems** can be employed, while **conversation AI** can facilitate human-machine interaction. By strategically combining these techniques, financial institutions can develop sophisticated AI solutions that drive business value and improve customer experiences.

Figure 1 presents a spectrum of AI problem complexities, ranging from simple to more complex. Each level demands different AI techniques and capabilities. For instance, automating routine tasks requires relatively straightforward AI applications, while developing sophisticated AI-powered advisors, such as those for retirement planning, necessitates advanced techniques like RAG, knowledge graphs, and potentially multi-agent systems

4 Implementation Study

This section described how we used a specific multi-agent AI solution, crewAI [13] to implement functionality common to many retirement planning applications typically used by finance organisations in Australia. We also demonstrate the progression of using RAG tools as part of this multi-agent framework.

4.1 Use Case: Retirement Planning Support

The retirement planning industry is multifaceted, encompassing regulatory policies, industry trends, retirement product offers and individual customer cir-

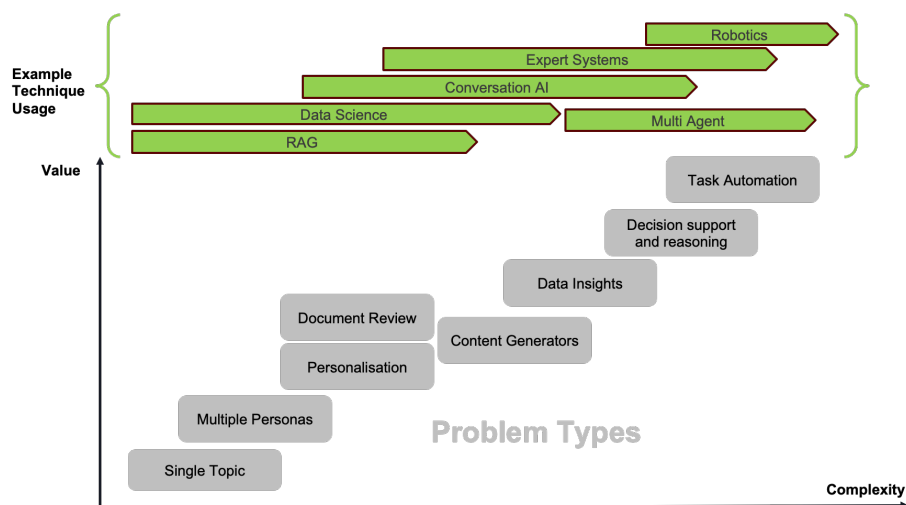


Fig. 1. AI adoption framework

cumstances. Effective retirement advice relies on understanding these elements. Therefore, an organization’s customer support staff are typically required to understand the following domains of knowledge:

- **Retirement Policy Knowledge:** This includes legislative aspects such as the maximum contributions permissible by age, and specific regulations such as the Superannuation Industry Act [2].
- **Retirement Industry Knowledge:** This encompasses knowledge regarding the current state of the industry, such as average performances of investments, and how much people need to save to live comfortably.
- **Customer Specific Knowledge:** Providing advice to customers requires careful consideration of their situation. This includes general information, such as their current investment choices, their savings situation, and how much they are being charged, but could also extend further, such as their personal risk tolerances.

Equipped with this comprehensive knowledge, support staff can address common customer inquiries, such as:

- *Can I increase my contributions by x amount per year?*
- *How is my retirements saving investments performing compared to others?*
- *Am I on track for retirement?*

However, ensuring consistent and effective utilization of these knowledge domains can be challenging due to continuously evolving regulations and industry trends. Furthermore, personalizing advice requires a thorough understanding of the customer and their situation, which could be time-consuming and error-prone. We explored the potential of applying AI for this case, specifically through a multi-agent system, powered by LLMs.

4.2 Implementation

We employed crewAI, a multi-agent orchestration framework designed to manage a team of specialized LLM-based agents that have the ability to delegate tasks, and utilize tools to solve complex problems collaboratively. We created four agents, three of which are related to the associated domains previously mentioned. The fourth agent is responsible for quality assurance. These agents, with their assigned tasks, tools and grouping in multiple crew compositions, support activities associated with retirement planning. A visual overview of the implementation is provided in Figure 2. In the following sections, we discuss how we have used crewAI [12] to achieve this ⁴.

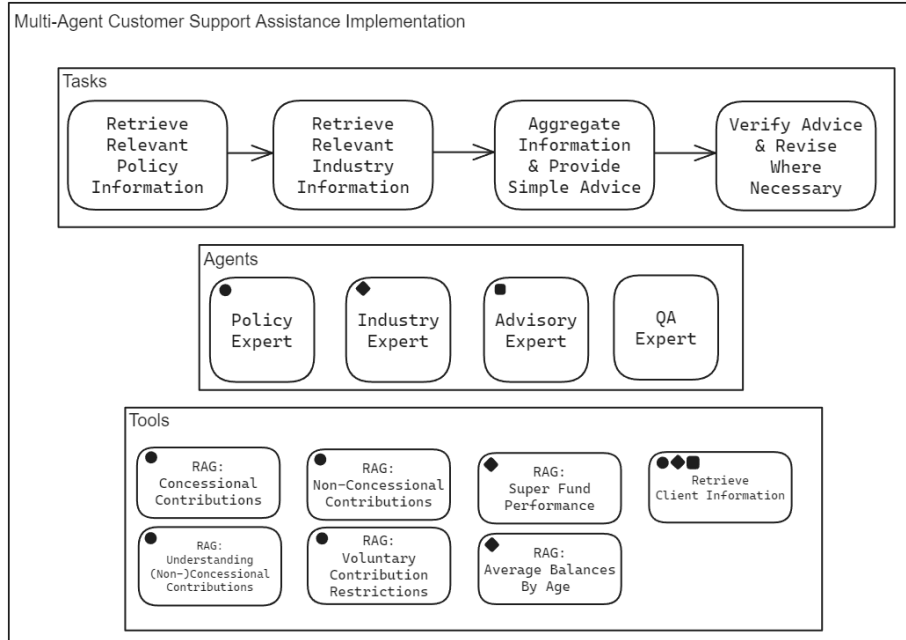


Fig. 2. Overview of the implementation in crewAI - Showcasing the Tasks and Agents, including the Tools available them, as indicated by distinctly shaped icons

Agents In the crewAI framework, agents are autonomous units which perform tasks, make decisions and communicate with other agents. Each agent is defined by a specific **role**, which captures its function in a structure called **crew**. The agent’s expected behavior, including decision making, are specified using a LLM.

⁴ Implementation available at GitHub: <https://github.com/Thomas-mp4/Multi-Agent-Retirement-Planning>

More specifically, each agent has a **goal**, and **backstory**, which serve as a way to provide further context to the agent, such that it can potentially exhibit more desirable behavior [27]. For this case study, we defined four agents:

- **Retirement Policy Expert:** Specializes in retirement policies and regulations, ensuring that relevant policy information is accurately provided in response to the client’s inquiry.
- **Retirement Industry Expert:** Expert in industry data and trends, with the responsibility to highlight relevant information in the context of the client’s inquiry.
- **Advisory Expert:** Primarily responsible for aggregating information received from the experts, and delivering clear and concise advice tailored to the client’s needs.
- **QA Expert:** Acts as a final fact-checker, reviewing the proposed advice for inconsistencies or errors, and assigning tasks to the appropriate expert if any discrepancies are found.

The first three agents are based on the aforementioned knowledge domains and do not have the ability to delegate tasks themselves. The fourth agent, the quality assurance agent, focuses on the validity of the advice, and delegates tasks accordingly as it deems fit to achieve this, based on its defined identity. However, this final check is limited by the capabilities of the LLM that powers the agent, and the definition it has been assigned within the framework, and should thus not be considered equivalent to a thorough review.

An example definition of an agent is provided in Figure 3. We experimented with several LLM providers to power the agents including Ollama [18], and Groq [11], utilizing Meta’s llama models. However, due to compatibility issues and rate limiting issues, the final iteration utilizes OpenAI’s GPT-4o [19] instead, to power the agents.

Tasks Tasks in crewAI are assignments that agents must complete. Each task requires a **description**, an **expected output** and, if the crew is running sequentially, a pre-assigned **agent** responsible for fulfilling the task. In case the crew runs hierarchically, a managing agent, either declared explicitly, or implicitly by the crewAI framework, becomes responsible for assigning tasks to agents. In our application, we employed a sequential process, where we only allow the QA agent to delegate tasks, such that revisions can be made where necessary in collaboration with other agents. Additionally, we enable the crew to utilize a memory system, implemented in the crewAI framework, such that coherence over a sequence of actions is maintained for all agents. We defined four tasks, each corresponding to an agent for our system:

- **Policy Task:** Retrieves relevant policy information, such as applicable contribution caps
- **Industry Task:** Retrieves relevant industry information, such as average balances

```

advisory_agent = Agent(
    role="Advisory_Expert",
    goal=("Understand_the_client's_inquiry,_and_based_on_the_information_
gathered_by_colleagues,_
    provide_the_best_retirement_advice_in_a_simple_manner_"
    "that_is_easy_for_the_client_to_understand."),
    backstory=("With_a_background_in_client_advisory_services,_
    you_specialize_in_understanding_client_inquiries_and_synthesizing_
    complex_information_into_clear,_
    actionable_advice._"
    "Your_communication_skills_ensure_clients_feel_confident_in_their_
    financial_decisions."
    "Make_sure_to_provide_full_complete_answers,_and_make_no_
    assumptions."),
    tools=[retrieve_client_information],
    allow_delegation=False
)

```

Fig. 3. Example of the Advisory Expert Agent (Python)

- **Advisory Task:** Aggregates information from the results of the previous tasks, and provide simple advice
- **Quality Review Task:** Reviews the information provided by the other agents, and ensure they are correct and meet the client’s needs.

Figure 4 provides an example of task definition. The formatted string literals refer to a hypothetical client’s full name and the query they have provided to customer support. By inserting this information into the task description, it allows the designated agent to be aware of the necessary context, and helps keep agents’ behavior relevant to the client and their query.

Tools Tools can help agents fulfill their tasks effectively. The crewAI framework offers a toolkit with ready-to-use tools [14] such as RAG. Note that the crewAI framework is built on top of LangChain [7], and thus LangChain tools are fully compatible. Additionally, if no existing tools satisfy the application’s requirements, custom tools can be implemented as well. Importantly however, the usage of tools by an agent is only possible when the LLM powering the agent is capable of function calling [17].

In our application we used tools to facilitate obtaining relevant and up-to-date information by the policy expert and the industry expert agents. We selected a collection of openly available articles and webpages, and converted them into plain text representations, in order to utilize crewAI Toolkit’s `TXTSearchTool`, which employs RAG. These include information about the following, where notably the policy related information is based on articles from the Australian Taxation Office [4,5,3,6], primarily concerning contribution policies:


```

advisory_task = Task(
    description=(f'Understand_the_client\'s_inquiry,_and_based_on_the_information_
gathered_by_\'
        f'the_policy_and_industry_experts,_provide_the_best_retirement_
advice_in_a_simple_manner_\'
        f'that_is_easy_for_the_client_to_understand.\'
        f'The_client\'s_request:_{query}\'
        f'The_client\'s_name:_{client}\'
        f'Consider_which_tools_you_actually_need,_and_also_consider_
whether_this_task_is_necessary_given_the_client\'s_request. '),
    expected_output="A_clear_and_concise_retirement_advice_report_that_
addresses_the_client\'s_inquiry"
                    "_and_integrates_insights_from_policy_and_market_analysis.",
    agent=advisory_agent
)

```

Fig. 4. Example of the Advisory Task (Python)

- (Non-)Concessional contribution caps
- (Non-)Concessional contribution information
- Voluntary contribution information
- Industry average performance return
- Average super balances by age

Additionally, we defined a custom tool to retrieve client information, given their full name. In our proof-of concept, this function merely returns the contents of a text file corresponding to the given full name, but in a deployed environment, this could be implemented as a call to a retirement application or its underlying database. The agents adapt to the utilization of custom tools by inspecting its Python docstring, and by inspecting thrown runtime exceptions, if applicable. For instance, if an agent attempts to retrieve a client’s information using the custom client retrieval tool without proper formatting of the parameter, a `FileNotFoundError` would be thrown specifying the appropriate format, which allows the agent to adapt accordingly, akin to how a developer would handle such an exception.

4.3 Results

To test this system, we employed a fictional client John Doe, who’s query is as follows: *"Am I on track for retirement?".* John Doe is aged 55, is male, has a current balance of \$530,000 AUD, has made \$20,000 AUD of contributions for the current year, and experienced a performance return of 6% and 7% in 2022-2033, and 2021-2022, respectively.

The application produced a report with several interesting sections. One key section includes the comparison between the client’s performance to the industry average, depicted in Figure 5. By using RAG and a text representations of a web

articles that also incorporate tables, the agents were able to gather accurate and relevant information. Additionally, the agents compared the client’s age and balance to industry statistics, and revealed that the client’s investments were performing well, as can be seen in Figure 6. Beyond these comparisons, the report discusses relevant policies and recent adjustments, such as changes to (non-)concessional contribution caps, provides recommendations to the client, and includes projections and calculations, providing the client with a comprehensive report, autonomously.

Based on industry data from ..., the performance of super funds over various periods is as follows:

1 Year:
Growth (61 – 80% growth assets): 9.9%

3 Years
Growth (61 – 80% growth assets): 5.9% per year

5 Years:
Growth (61 – 80% growth assets): 7.3% per year

10 Years:
Growth (61 – 80% growth assets): 7.0% per year

John's portfolio return of 6% for 2022–2023 and 7% for 2021–2022 is within the range of industry averages, showing that his investments are performing competitively

Fig. 5. crewAI Retirement Planning Output Excerpt - Industry Comparison

According to industry data, the average superannuation balances for men and women by age are as follows:

Age 55–59:
Men: \$316,457
Women: \$236,530

John's current balance of \$530,000 AUD is significantly higher than the average balance for his age group, indicating that he is in a strong financial position compared to his peers.

Fig. 6. crewAI Retirement Planning Output Excerpt - Balance Comparison

In order to generate this report, the crew executed all tasks sequentially, using tools where necessary, until the chain reached the quality assurance expert. Upon inspection, the QA expert deemed it appropriate to verify the policy related recommendations, and thus delegated this task to the policy expert. Using the tools available to them, the policy expert cross-referenced the report it was given with its own information, and verified the information was correct. After

receiving the response from the policy expert, the QA expert decided that that no further verification is necessary, and that its task had thus been fulfilled, finally providing the report as the crew’s output. This workflow is depicted in Figure 7. It is important to note that this workflow, and the output of this crew are non deterministic. That is to say, even with the crew’s agents and tasks being defined in a specific way, results slightly differ from each run. This can be attributed to the nature of LLMs, such as, the temperature of an LLM. The temperature of an LLM determines how creative the model becomes, that is to say, the chance of the model selecting a less or more probable next token. A higher temperature value, above 1.0, will result in more randomness, whereas a lower temperature, below 1.0, will be more deterministic. It is however important to note that even at a temperature of 0.0, the system will still be non-deterministic, as this is inherent to LLMs [20].

Running the crew without any adjustments to the codebase results into different behavior and different results. It approximately takes 4 minutes for a crew to fulfill all its tasks, with the crew taking longer when agents decide to utilize their tools more extensively, such as making additional RAG calls, or when the QA agent delegates additional tasks.

5 Discussion

This section captures some observations from our experiments, some of which have influenced our thinking about future work.

5.1 Responsibility

Explainability & Transparency As we expected, LLMs, such as those employed in proof-of-concept to power agents, produce outputs that are less deterministic and more difficult to explain compared to traditional machine learning models, such as decision trees. For example, we noted instances where the same agents, with the same configuration would make different assumptions regarding the client’s retirement contribution (non-concessional vs concessional). This unpredictability highlights the need for maintaining transparency about how data is processed, and how the system reaches its conclusion.

Additionally, it is important to be aware of any biases present in the employed LLMs due to its computational power. Even if RAG is utilized with correct data, the LLM is still capable of misinterpreting or misrepresenting the data, potentially resulting into negative outcomes. Potential ways to mitigate these problems include auditing the decision-making process of agents, such as the interactions it makes with tools to deduct an answer, and encouraging agents to clarify their reasoning themselves, such as by adding the sources it utilized when providing an answer. This would be especially valueable in scenarios such as depicted in Figure 7, where the QA expert deemed it necessary to ask for verification of specific information in the final report.

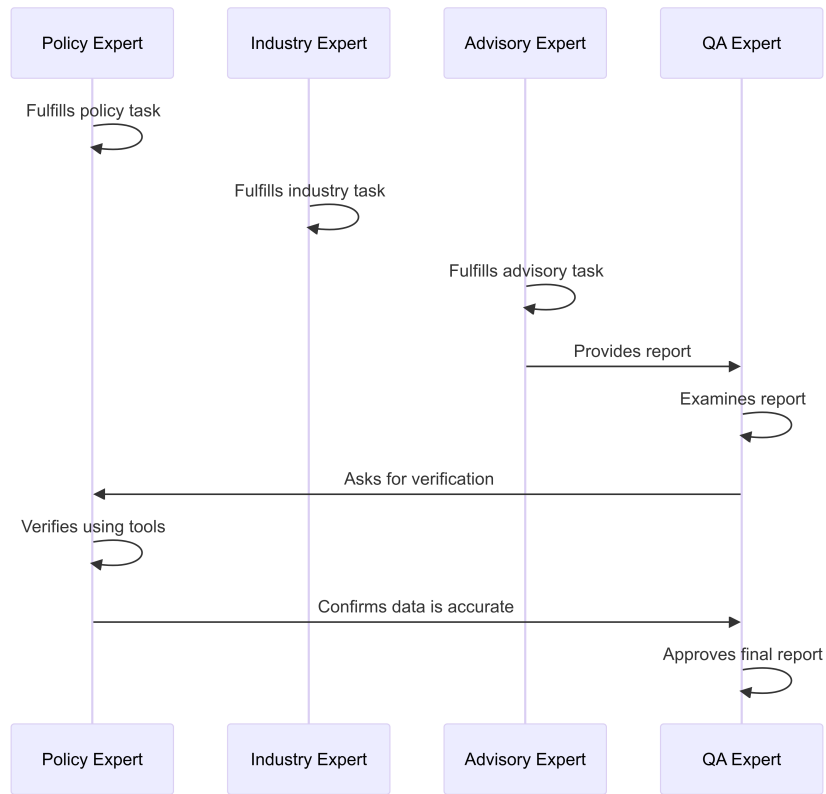


Fig. 7. crewAI Retirement Planning Output - Simplified Visualized Workflow

Security Utilizing LLMs to power a multi-agent system that assists with retirement planning also involves handling sensitive personal and financial information, which introduces specific security challenges. We encountered that lower-capacity models, such as llama-8b, may inadvertently expose parts of the crew’s internal mechanisms, such as the Python Docstring of tools. This could potentially leak sensitive information, or expose vulnerabilities which could be exploited by malicious actors. Furthermore, incorporating external technologies or services, such as utilizing the OpenAI API to utilize LLMs, or augmenting the system using web scraping tools, introduces risks associated with external data breaches and unauthorized access.

It is possible to run crewAI completely locally, but this does come with the corresponding computational costs, especially considering that smaller LLMs do not perform as well as models that require larger inference budgets. Thus, it is important to make an appropriate assessment of security risks involved when designing multi-agent applications in order to make a balanced decision.

5.2 Scalability

Preprocessing of Data Handling complex data types, such as documents containing tables, web pages containing images, or markup content requires preprocessing which adds additional overhead. The use of RAG also includes obtaining embeddings. It is essential that agents that handle different types of data are capable of accurately parsing the various mediums they are presented in. Failure to do so could potentially lead to misinterpretation of information, causing undesirable behavior.

Architecture Design One of the inherent challenges in designing systems such as the one we described in this paper is striking the balance between constructing a system effective at solving the known problems, while also maintaining enough flexibility to address unforeseen problems. This becomes increasingly important the more the system scales, as changes become more difficult and costly to implement ad-hoc. In order to overcome this challenge, it could be beneficial to apply the separation of concerns principle, especially considering this is also potentially beneficial to the behavior of the agents themselves.

Furthermore, optimizations of performance are also necessary in order to scale to increasingly complex tasks. The more tasks are involved, the more likely it is more agents will be necessary to successfully fulfill them. If agents do not make proper use of their memory, or in-effectively share information, they could make redundant calls, which consequently has a negative effect on performance, and thus also on sustainability. For instance, we observed that agents tend to exhaust all the tools available to them to acquire as much context to fulfill their tasks, without taking into consideration whether they really require these tools. We also encountered an increase of redundant calls when employing an hierarchical approach as opposed to a sequential approach. The managing agent in the hierarchical approach tends to repeats its task delegations, despite them being fulfilled before. Moreover, the stability of the frameworks utilized to create multi-agent systems have a significant impact on performance. We experienced attempts of executing the system, where it failed mid-way due to errors, or where agents got caught in a loop, trying to perform the same action with seemingly no apparent reason. As this is an emerging and rapidly developing field, with new frameworks and features being frequently released, it is important to consider these risks.

In summary, the lessons learned from these experiments should serve as input in designing new architectures for other solutions, to avoid making inadequate design and implementation choices, which can be difficult to correct.

6 Conclusions & Future Work

This paper was an attempt to bridge practical and research issues associated with the adoption of LLM-based and emerging AI technologies in financial organisations. Motivated by one specific problem, i.e. providing AI support for

a retirement planning application to help customer centre staff deliver better customer service and more efficient processing of many document sources, we decided to experiment with simpler, RAG solutions, in part, to better understand the emerging multi-agent AI techniques. In fact, we managed to establish links between the two, by letting agents make use of RAG itself. The results were quite encouraging, as for example agents provided insights similar to a human expert. On the other hand, the path to reaching this level of development had many challenges, associated with both the native LLM issues such as its stochastic nature, but also the maturity of the multi-agent solution chosen.

We believe that experiments such as this, can also help the workforce within a financial organisation to develop their own understanding of the maturity of AI based solutions, and better prepare them for liaising with technology experts, specialist solution providers and AI platforms vendors, in selecting and procuring the most cost-effective solutions. The scalability of adoption is thus capturing what technologies are best fit-for-purpose and how to move from proof-of-concept to production stage of AI adoption, and continuously applying lessons learned to adapt to higher complexity problems.

There are several directions for our future work, inspired by lessons learned so far and other use cases. One of them is exploring the value of knowledge graphs as a way of integrating them for LLM solutions. We would also like to explore the potential of a distributed multi-agent system, wherein the system can compose of agents hosted in several environments, instead of a single environment. This could allow for better performance, as well as redundancy within the system, which adds to the system’s overall resilience.

Furthermore, the topic of consistency and quality of agent output is a valuable avenue for future work. Currently, our system employs fairly generic definitions for agents and tasks, which provides more flexibility, enabling the system to handle a wide variety of client inquiries. However, this flexibility can lead to variability in the results, which might impact the reliability and predictability of the system’s output. By refining and specifying the definitions of agents and tasks, particularly aligning more closely with a specific topic in context of retirement planning advice, the system could potentially produce more consistent and high-quality results. This increased specificity might reduce the system’s ability to handle a wider range of inquiries, but it could lead to more predictable and desirable behavior. Experimentation with this trade-off between flexibility and consistency is a valuable future work avenue in order to determine what is most appropriate.

Additionally, the consistency and adherence of agents to their assigned tasks and roles, while complying with clearly defined guardrails, should be further explored. For instance, it would be desirable for developers to have fine grained control over the permission levels of agents, such that it is certain they handle data conform to a set of defined business or legal rules, despite being powered by a stochastic LLM. This is particularly of essence when handling sensitive data that could impact client privacy. One potential approach to achieve this is through reinforcement learning, enabling agents to adjust their behaviors based

on human feedback, utilizing the agents' reflection capabilities [21] or involving humans as ultimate decision maker and enforcer.

Acknowledgements. We would like to express our gratitude to the anonymous reviewers for their valuable feedback and constructive comments. Their insights contributed to the improvement of this paper during the revision process.

Disclosure of Interests. The authors declare no competing interests relevant to the content of this article.

References

1. Australian Banking Association: Positioning Australia as a Leader in Digital Economy Regulation - Automated Decision Making and AI Regulation (2022), <https://www.ausbanking.org.au/submission/automated-decision-making-and-ai-regulation/>, last accessed 2024/07/07
2. Australian Government: Superannuation Industry (Supervision) Act 1993 (1993), <https://www.legislation.gov.au/C2004A04633/latest/text>
3. Australian Taxation Office: Understanding concessional and non-concessional contributions (2023), <https://www.ato.gov.au/individuals-and-families/super-for-individuals-and-families/super/growing-and-keeping-track-of-your-super/caps-limits-and-tax-on-super-contributions/understanding-concessional-and-non-concessional-contributions>, last accessed 2024/07/07
4. Australian Taxation Office: Concessional contributions cap (2024), <https://www.ato.gov.au/individuals-and-families/super-for-individuals-and-families/super/growing-and-keeping-track-of-your-super/caps-limits-and-tax-on-super-contributions/concessional-contributions-cap>, last accessed 2024/07/07
5. Australian Taxation Office: Non-concessional contributions cap (2024), <https://www.ato.gov.au/individuals-and-families/super-for-individuals-and-families/super/growing-and-keeping-track-of-your-super/caps-limits-and-tax-on-super-contributions/non-concessional-contributions-cap>, last accessed 2024/06/23
6. Australian Taxation Office: Restrictions on non-voluntary contributions (2024), <https://www.ato.gov.au/individuals-and-families/super-for-individuals-and-families/super/growing-and-keeping-track-of-your-super/caps-limits-and-tax-on-super-contributions/restrictions-on-voluntary-contributions>, last accessed 2024/06/23
7. Chase, H.: LangChain (2022), <https://github.com/langchain-ai/langchain>
8. David, W.: David Walker's tech trends to watch in 2024 (2024), <https://www.westpac.com.au/news/in-depth/2024/02/david-walkers-tech-trends-to-watch-in-2024/>, last accessed 2024/07/07
9. Dodig-Crnkovic, G., Burgin, M.: A Systematic Approach to Autonomous Agents. *Philosophies* **9**(2), 44 (2024). <https://doi.org/10.3390/philosophies9020044>
10. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-Augmented Generation for Large Language Models: A Survey (2024), <http://arxiv.org/abs/2312.10997>

11. Groq: Fast AI inference (2024), <https://groq.com/>, last accessed 2024/07/07
12. João, M.: CrewAI documentation, <https://docs.crewai.com/>, last accessed 2024/07/05
13. João, M.: CrewAI (2024), <https://github.com/joaomdmoura/crewAI>
14. João, M.: CrewAI tools (2024), <https://github.com/joaomdmoura/crewai-tools>
15. Kasisto: Conversational ai solutions for banking and finance, <https://kasisto.com/>, last accessed 2024/07/07
16. King & Wood Mallesons: Australian government interim response on the regulation of ai: inching towards safe and responsible ai (2024), <https://www.kwm.com/au/en/home.html>, last accessed 2024/07/07
17. Mistral AI: Function calling (2024), https://docs.mistral.ai/capabilities/function_calling/, last accessed 2024/07/07
18. Ollama: Ollama (2024), <https://github.com/ollama/ollama>
19. OpenAI: OpenAI API platform (2024), <https://openai.com/api/>, last accessed 2024/07/07
20. Ouyang, S., Zhang, J.M., Harman, M., Wang, M.: LLM is Like a Box of Chocolates: The Non-determinism of ChatGPT in Code Generation (2023)
21. Park, J.S., O'Brien, J.C., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative Agents: Interactive Simulacra of Human Behavior (2023). <https://doi.org/10.48550/arXiv.2304.03442>
22. Posh.AI: Ai built for banking, <https://www.posh.ai/>, last accessed 2024/07/07
23. Takyar, A.: Agentic RAG: What it is, its types, applications and implementation (2024), <https://www.leewayhertz.com/agentic-rag/>
24. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W.X., Wei, Z., Wen, J.: A survey on large language model based autonomous agents. *Frontiers of Computer Science* **18**(6), 186345 (2024). <https://doi.org/10.1007/s11704-024-40231-1>
25. Weng, L.: LLM Powered Autonomous Agents (2023), <https://lilianweng.github.io/posts/2023-06-23-agent/>, last accessed 2024/07/07
26. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, S., Zhang, X., Liu, J., Awadallah, A.H., White, R.W., Burger, D., Wang, C.: AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework (2023)
27. Xu, B., Yang, A., Lin, J., Wang, Q., Zhou, C., Zhang, Y., Mao, Z.: ExpertPrompting: Instructing Large Language Models to be Distinguished Experts (2023)