Can AI Deliver Financial Advice?

The art (and science) of conversation

Exploratory research Late 2023–Early 2025

Aritra Chakravarty
Product Direction

Mykhaylo Merkulov
Solutions Architecture & Backend

Julia Earthrowl
Product & Experience Design

Yan Zubrytskyi
Front-end (Flutter)

Kavitha Vinod
Product Analysis & Documentation

JIFFY.ai, 860 N. McCarthy Blvd, Suite #210, Milpitas, CA 95035 USA.

Executive Summary

Personalized financial advice remains inaccessible to most Americans, typically reserved for those with $250,000+ in investable assets, due to the knowledge, experience, and time it requires to deliver quality advice[1]. This **exclusivity comes at a significant societal cost**: the average American loses over $1,000 annually to preventable financial mistakes—totaling $240-388 billion nationally[2]—while nearly half have no retirement savings whatsoever, and two-thirds of non-retired adults report being off-track for retirement[3].

From late 2023 to early 2025, we have been exploring the feasibility of using Large Language Models for financial advice conversations with the aim of making financial planning more accessible. We believe that a comprehensive agent architecture, deployed through advisory firms, under direct human-in-the-loop supervision of a licensed human advisor, will enable firms to serve more clients, allowing for a 1:200 ratio vs 1:20.

In this paper, we assess the tenets of a good financial advisor (communication, competence, and integrity) and develop with these in mind. **We look in-depth at solving challenges relating to Multi-Modal Conversations, Voice models, Prompt Tuning, and LLM Orchestration.** Our focus has been on the conversation: aiming to deliver an advisory conversation that feels as supportive, insightful, and responsive as talking to a human financial professional, but we have also covered the technological and regulatory challenges that would need to be overcome if this were productionized.

Over the course of this work, we have navigated a landscape that has been shifting in real time, and we have found ourselves testing the limits of a technology while its boundaries are still expanding. This research serves as documentation of that change and assessment of possibilities as they stand today.

'FinleyAI,' the name we have given to our build, can help a person understand their current financial situation, consider their goals, and make actionable plans for their future. He can converse in voice, transcript, and chat modes. He can present accurate projections and interactive widgets mid-conversation and talk through the details in however much depth the client needs. He can navigate a range of financial products topics and timelines, deal with the emotional complexities of humans and support them right through to account opening.

Core to the conversation build has been finding the sweet spot in the balance of hard numbers vs softer, human aspects. The financial-planning-optimized experience we have developed is a significant step up from what is available through existing general LLM platforms. It supports:

- **97% success rate in delivery of all goals in any conversation,** (speed and fluidity of dialogue, no errors in tool misuse or formatting mistakes, serving multi-modal elements correctly).
- **A hybrid tooling model** that balances speed, reliability, and conversational naturalness better than either native or fully structured approaches.
- **A distinctive voice interface** powered by ElevenLabs Flash v2.5 with ultra-low latency (~75 ms), offering a human-like presence.
- **A modular orchestration framework** leveraging OpenAI's Agents SDK for real-time interaction and LangChain/LangGraph for more tooling support.
- **A scalable prompt-tuning and evaluation framework** that includes LLM-based simulation and scoring across five core dimensions, allowing fast iteration and quality control.
- **Built to serve under the direct supervision of a licensed human advisor.** Designed to be a digital co-pilot, never a discretionary decision-maker. All regulated actions are subject to human-in-the-loop review, with full audit trails, observability, and user transparency.

What We Learned

- **Multi-modal conversational experiences hugely help the user's** ability to understand the complexity of financial planning conversations.

- **Data and functions integrations with LLMs must be well-architectured**. Too many tools degrade LLM performance, fully structured output leads to robotic user experience. Hybrid approaches emerged as a sweet spot between control and flexibility.

- **Voice quality and persona matter** more in financial contexts than in general customer service ones. Trust is built (or lost) in a single inflection.

- **Prompt engineering is a team sport**. Enabling product managers and designers to directly shape prompts through custom UIs resulted in stronger advisor personas and more intuitive conversations.

What work still needs to be done?

While FinleyAI demonstrates that AI can support meaningful financial conversations, several areas remain open for further investigation:

- **Dynamic Scenario Planning and Inclusion Audits**
  Tools to let users explore alternate futures (e.g., job loss, caregiving needs) are in early stages. More research is needed on how AI can support interactive scenario modelling that adapts over time. Also, how well FinleyAI can serve underrepresented groups, non-standard financial lives, or culturally specific needs.

- **Auditing and Regulation Readiness**
  Full observability is built into our architecture, but the next step will be developing standards and tooling to make auditing regulator-ready.

Our goal was to explore the feasibility of using AI (specifically large language models) to close the financial advice gap—ambitious in a sector that is both technically demanding and tightly regulated. **The process demanded continuous iteration, technological adaptation, and interdisciplinary thinking. The result is a product, FinleyAI, that not only demonstrates feasibility but offers tangible proof of what is possible today.** Deployed thoughtfully within an advisory, it could open up advice to many more people. **With the challenges the world faces, not to mention the**

pensions issues that are present in many markets across the globe, a future where more people have access to financial advice is one to feel hopeful about, and one that despite hurdles, we should seek to start delivering now.

We invite regulated institutions, wealth firms, and policymakers to engage in shaping this future with us.

Introduction

Imagine a world where, regardless of income, everyone had access to good financial advice – where the fear of unexpected expenses, the uncertainty of retirement, and the frustration of not having the funds to achieve important life goals, like homeownership, were replaced by confidence in making sound financial decisions. Today, that vision feels distant. Many people struggle to build even a modest safety net, with little to no savings to cover emergencies or to secure a comfortable retirement. The root of this struggle often lies in financial illiteracy, a widespread issue that leaves many ill-prepared to navigate their financial futures.

What if there was a way to bridge this knowledge gap, to put the power of financial planning into the hands of everyone?

With the strides in AI today, and the massive improvement of AI systems in communicating conversationally and analyzing complex information, the possibility of AI providing personalized financial advice is a real and exciting possibility. However, there are many challenges to overcome. The world of financial advice is particularly complex, highly regulated, and deeply personal.

Can AI rise to the occasion?

Section 1

The Financial Advice Gap

Despite the critical role that financial literacy plays in economic well-being, many individuals struggle with the basics. As financial illiteracy continues to drive poor financial outcomes and widen economic inequality, it begs the question: are current solutions for financial advice truly meeting people's needs? If not, where are they falling short?

The Financial Illiteracy Problem

The financial illiteracy problem is widespread, with many individuals lacking the necessary knowledge to make informed financial decisions. Studies show that a significant portion of adults struggle with understanding basic financial concepts, including budgeting, interest rates, and investments. For example, in the United States, according to the most recent National Financial Capability Study (NFCS)[1] conducted by the FINRA Investor Education Foundation in 2022, only about 37% of respondents could answer four out of five basic financial literacy questions correctly, highlighting a major gap in understanding key concepts like inflation and risk diversification. This widespread financial illiteracy leaves individuals vulnerable to poor financial choices, including excessive debt, insufficient retirement savings, and susceptibility to financial fraud. Furthermore, a 2023 report by the OECD [2] underscores how financial illiteracy contributes to economic inequality, as those lacking financial knowledge are less likely to invest or take advantage of opportunities for wealth accumulation. This lack of financial literacy can contribute to poor financial decisions, such as living paycheck to paycheck, under-saving for retirement, and accumulating debt. The lack of financial knowledge creates a market for any financial advising tool or business (whether that be an app, or a person).

Where Can People Get Financial Advice Today?

Access to high-quality financial advice would help improve people's financial wellbeing, but it is hard to come by. Financial literacy is not widely taught in schools, and finding answers elsewhere can be challenging and unaffordable.

Internet and Social Media

One can use the internet or social media for financial advice, but it is difficult to know where to start and where to find information that is digestible for a beginner, comprehensive, and accurate. Going online for financial advice is not the most interactive or personalized experience either – a user can go to a website or watch videos on social media regarding finance but would have to figure out how to apply it to their situation. If they had any questions, it would be difficult to reach the creator of the content they are consuming for advice. Additionally, the internet is rife with misinformation, with misleading or inaccurate advice often being spread through social media, blogs, and unverified websites, as well as fraudsters hoping to make money off the financially vulnerable.

Robo-Advisory Apps

There are robo-advisory apps, such as Nutmeg, Betterment and Wealthfront, but they do not offer personalized financial advice as they are unable to evaluate your bigger picture or make personalized recommendations. For example, they immediately ask the user for a target amount for a goal, or how much they are contributing their goal per month. A user without much experience in planning for financial goals would need help with these questions; robo-advisory apps assume the user already knows the answer to these, making them more of a calculation tool than an advisor. When robo-advisory apps first came out they had a lot of promise, but they have not taken business from human financial advisors.

Financial Advisors

The issues with current apps and internet advice explain why, if possible, people pay for the services of a financial advisor. Having a financial advisor can provide personalized advice about one's specific situation that they may have difficulty getting off the internet. However, they are often unaffordable. Advisors often charge a percentage of assets under management (AUM), with the industry average ranging from 1% to 2% annually, making it expensive for those with smaller portfolios. For instance, someone with $100,000 in assets might pay $1,000 to $2,000 per year in fees. Additionally, many advisors require minimum investment thresholds, typically ranging from $100,000 to $2.5m[3]. They have to charge this level of fees due to the time and complexity of the job they do; however, this excludes individuals with fewer savings. Minimum thresholds to retain a financial advisor vary, but very generally, one should have between $50,000 and $500,000 of liquid assets to invest. 63% of Americans have less than $50,000 saved up for retirement, and the median bank account balance is $5,300 (Sall)[4].

For reference, in 2023 around 60% of Americans live paycheck to paycheck, meaning that they will not meet the minimum investment threshold or have enough money in their budget to allocate for a financial advisor. In 2024, around 28% of Americans across 4 generations have less than $1000 in savings (Adam)[5]. In 2022, it is estimated that only around 35% of people in the United States have worked with a financial advisor[6].

Conclusion

Existing solutions are not going to close the financial advice gap

Looking across the landscape of current options, it is clear that existing solutions fall short of closing the financial advice gap. Online resources are often overwhelming, impersonal, and rife with misinformation. Robo-advisory apps, while convenient, assume a level of financial knowledge many users lack and primarily serve as calculators rather than true advisors. Meanwhile, human financial advisors remain financially out of reach for the majority of Americans.

With the current problems with existing financial advice solutions, the rapid advance in artificial intelligence technology in recent years begs the question:

Is AI capable of solving the financial advice problem?

Section 2

What Makes a Great Financial Advisor?

To answer the question of whether AI can effectively solve the financial illiteracy problem, it is imperative to understand what makes a good financial advisor. Is it deep technical expertise? The ability to build trust? Or something more human, like empathy and personal connection?

To define this we looked at a range of trusted sources across regulatory, research and commercial institutions: CFP [5], FPA [6], Vanguard [7], MorningStar [8], JD Power [9] and Finra [10].

❖ Good Communication

The aspects of good communication:

➔ Clarity

A financial advisor should ensure a user has a full understanding of their current financial situation, the options available to them, and the recommendations and decisions that are being made, and should translate complex financial concepts as needed. They may pull on a range of tools to help them, whether it is showing charts or graphs, providing the client with interactive tools, or following up with summary conversation emails.

➔ Attunement

A financial advisor should be consistently aware, responsive, and mindful of the client's needs, preferences, and experiences, not just in isolated moments, but across the whole relationship. They will display great empathy and rapport to build a trusting relationship with the client that makes them feel comfortable and confident.

➔ Proactivity

A financial advisor should offer advice proactively whenever the client needs it, without being asked. They will monitor the client’s financial situation, world events, and respond accordingly.

❖ Competence

The aspects of competence:

➔ Certification

A financial advisor must be certified by the appropriate regulatory body in that country and keep up to date with changes.

➔ Good Judgement

A financial advisor must display good judgement and make consistently good decisions, considering all factors.

➔ Systems Thinking

A financial advisor must be complexity-literate and able to manage multiple moving aspects of a client's financial life and future.

❖ Integrity

The aspects of integrity:

➔ Fiduciary Duty

All financial advisors must abide by their fiduciary duty and provide unbiased recommendations aligned with the client’s goals.

➔ Data security and Privacy

A financial advisor must maintain confidentiality, ensuring the client's sensitive financial information is always protected.

➔ Accountability

A financial advisor is responsible for their actions, decisions, and outcomes; and must be able to explain them if challenged. They must keep sufficient records.

Conclusion

The highest levels of communication, competence and integrity are the tenets of a good financial advisor

Great financial advice is about much more than numbers—it is a delicate balance of knowledge, empathy, communication, and trust. The core traits we have outlined form the foundation of what is valued most in advisors: good communication, competence, integrity.

Understanding this high bar is critical as we evaluate the role of AI. We need to be able to judge whether it can give financial advice in a way that meets the standards that human advisors have set.

Section 3

Could existing AI platforms solve the Financial Advice gap?

People are increasingly willing to use AI for services that once seemed deeply personal or required specialized human expertise. In recent years, AI-powered tools have made significant inroads into sectors like mental health, where apps like Woebot and Wysa provide cognitive behavioral therapy techniques through friendly, always-available chatbots. In healthcare, AI is beginning to be used to check symptoms, diagnostic support, and personalized treatment recommendations. In education, students now rely on AI tutors for everything from math help to coding and writing.

So, given the historic shift in AI's capabilities and the public's increasing openness in using services powered by it, might AI platforms such as ChatGPT start closing the Financial Advice Gap.

As OpenAI's ChatGPT is a leader in this space, we have used it as a benchmark for what a financial advice conversation would look like with today's cutting-edge AI tools.

Here are a few ways we could consider people might use ChatGPT:

- A user asks ChatGPT about their finances
- A user uses a CustomGPT built by a third party
- A user builds their own financial CustomGPT

To assess how viable each of these options is to act as a financial advisor we have tested and assessed them against the '*What makes a good financial advisor'* standards listed in Section 2. The results have been recorded in a summary table in Appendix 3 and summarized below. (*Assessment done in Q3 2024)*

A person uses ChatGPT

Since low user knowledge is required to use ChatGPT, this makes it the most viable current option as an AI financial advisor.

What ChatGPT Does Well

In terms of user experience, ChatGPT has the ability to provide genuinely good advice, remember previous conversations and keep track of conversation history. If the user were to set up multiple financial goals in the same chat at different times, and then later ask questions about their goals, ChatGPT would be able to provide that information. Additionally, it can provide recommendations for actionable steps to further one's financial goals. For example, if ChatGPT advised the user to open an IRA and the user asked for provider options, it would list some examples such as Fidelity or Vanguard. If asked where one can open a Fidelity account, the link for the Fidelity website is given along with rough instructions for how to set up an IRA on Fidelity's website.

Where ChatGPT Falls Short:

As discussed in Section 2, for ChatGPT to be a successful financial advisor, during and after a conversation it needs to:

- **Good Communication** (Clarity, Attunement, Proactivity)
- **Competence** (Certification, Good Judgement, Systems Thinking)
- **Integrity** (Fiduciary Duty, Data security & Privacy, Accountability)

Aside from the fact that ChatGPT is not regulated or certified as a financial advisor so should not be trusted as such, ChatGPT's biggest shortcoming as an AI financial advisor is its *inability to be proactive* due to its purpose as a responsive tool (see table and/or subsections below for more details).

Built as a Responsive Tool, Rather than a Proactive One (Section 2, #5)

ChatGPT is built as a responsive tool – it is great at answering a question or a task that a user prompts it with. However, a financial advisor has a proactive role, meaning the responsive nature of ChatGPT means it is not an ideal match for the role. For example, ChatGPT is not connected to accounts the user might own, making it difficult to stay up to date on a user's financial affairs. ChatGPT cannot track if the user has followed through on advice given, or if they are off track to meet their goals without the user verbally telling it. They cannot initiate conversations with the user in the form of a notification or remind the user to actually move forward on plans to meet your goals unless the user asks.

Conversationally, ChatGPT does a good job at proactively asking the user questions such as if they would like to delve into more detail of anything covered. However, it is not the best at prompting the user for all information necessary to create a personalized plan to meet their goal. For example, in our tests, the user was asked for their age and return rate for their brokerage account, but was never asked if they were employed, or what their income was. If it does not occur to the user to provide this information, the financial plan generated will not be personalized enough to their situation.

A person Uses a Third party CustomGPT

OpenAI explicitly states that ChatGPT and CustomGPTs should not be used for financial advice, and no custom GPTs on the public platform are allowed to provide such guidance, eliminating this as an option. Furthermore, financial institutions would not build a CustomGPT on the platform for advice, as they cannot monitor or audit the conversations, which is necessary for compliance purposes. This option is therefore ruled out for the foreseeable future.

A User Builds their Own Financial CustomGPT

Open AI allows people to “create their own GPTs,” meaning one can feed an LLM a financially focused prompt for a better financial planning experience than just using ChatGPT.

High User Prerequisites Eliminate this Option

The user must be tech-savvy, with an understanding of how to craft an effective prompt, as well as possessing the time and patience required to refine and tweak it. The user also would require a fair amount of financial knowledge to design a good system prompt for a CustomGPT. For instance, a retirement system prompt may specify when to introduce 401(k)s, IRAs, etc. for a good conversation flow. Due to the comfort with tech and financial knowledge required, CustomGPTs are never going to address the financial advice gap.

Though the user prerequisites rule out CustomGPTs as an AI financial advisor, we have tested conversations with a CustomGPT for good measure to see how it meets the requirements of a good financial advisor listed in Section 2. The custom GPT is much better conversationally than ChatGPT, but it could do better in terms of conversation quality and consistency. See the appendix below for sample conversations, results, and further detail regarding CustomGPTs.

Conclusion

In their current form, existing AI platforms will not solve the financial advice gap

We can see that existing general LLM platforms are not solving the financial advice gap today. They require a high level of financial and technical knowledge from the user and provide none of the proactivity and diligence required for a reliable experience.

This is not to say that things will not change in the future, but as it stands, especially with the regulatory challenges, we cannot see them solving the financial advice gap.

So next we explored whether a platform purpose built for AI financial advice could solve the problem. In the next two sections we will address both the theoretical benefits and explore the feasibility of the build.

Section 4

Would a proprietary platform improve the chances of success?

From the outset, a proprietary architecture for AI financial advice appears to offer notable benefits from both an experience and regulation perspective.

An app-based solution allows for an experience specifically tailored for the use case of financial advice

Conversational Experience Designed for Financial Advice

The conversational experience can be designed through the specific lens of a financial advisor, ensuring it can support everything needed for an advice conversation – building trust, supporting multiple modes, and dealing with complexity.

We can support the user through to Account Opening

A purpose-built app means this is transition from advice to account opening, can be managed as part of the whole experience (including human-in-the-loop checks)

Integrated Portfolio Management

When an investment account is opened, the user can manage this within the same interface, which in turn adds context to further planning conversations.

Proactive Advisor that's Always Tracking User's Finances

The app can support proactive financial guidance by monitoring and responding to a user's finances and letting a user know if they are off-track through notifications. The user does not have to initiate a conversation.

Ability Connect to User’s Bank Account for Live Data

A purpose-built app enables real-time connection to a user’s financial data across different accounts or providers. Connecting this data is highly valuable, as it allows for a more accurate and granular understanding of the user’s finances.

With our own platform, meeting regulatory standards is more feasible

Control stays with us – the implementing regulatory firm

Our own purpose-built environment means we have control over what is happening at all points in the user experience.

It will operate under an advisory firm’s regulatory umbrella

With this model we can support efficiency while retaining full human accountability. All regulated activities (e.g., investment advice, pension recommendations) can be signed off by a certified advisor in the firm. This maintains compliance with regulatory needs from FCA, SEC, and FINRA frameworks around scope of advice, fiduciary responsibility, and liability assignment.

Identity checks, etc. can be implemented

The user is onboarded into a control environment and dealt with in exactly the ways a standard financial app would support – e.g., identity and fraud check during onboarding, money laundering checks during transactions.

Visibility of conversations, ready for auditing

We can monitor all conversations, advice, decisions, and agreements. All regulated actions are subject to human-in-the-loop review, with full audit trails, observability, and user transparency.

Data security and privacy

As part of our build, we will need to ensure all data is fully protected and meets industry standards for data storage and transit, but this is more feasible with our own platform.

Conclusion

A bespoke build has a much higher chance of meeting the regulatory and experiential needs

A bespoke build has a much higher chance of delivering both the regulatory and experiential needs required to get AI financial advice into the market and start closing the advice gap.

However, there are still huge technical challenges to overcome to bring this to life.

The next section will document our journey addressing the range of requirements. Work was done across 2024–early 2025, and we are publishing this now in April 2025.

Section 5

The brief to ourselves

We used the aspects of a good financial advisor to identify key areas of research that need to happen to assess the level at which AI can deliver.

WHAT MAKES A GOOD FINANCIAL ADVISOR?		Research Area:
Good Communication	**Clarity** – Simplifies complex financial jargon into easy-to-understand terms during consultations.	*Can our AI have a human sounding conversation?* →**Voice Model Evaluation**
		Can our AI share charts and interactive tools during the conversation? →**Multimodal conversations**
		Can our AI discuss complex topics? →**Prompt engineering**
	Attunement – consistently aware, responsive, and mindful of the client's needs, preferences, and experiences, not just in isolated moments, but across the whole relationship.	*Can the AI talk to clients in a way that meets their personality and emotional needs?* →**NLP extraction**
		Does the AI stay aware of the softer client details (preferences, upcoming holidays, etc.)? →**Memory management**
	Proactivity – Offering guidance without being asked, Constant monitors the client's financial situation and acts accordingly	*Can the AI use inputs from various sources to keep track of the client's finances? For e.g.* • *Openbanking for monitoring client's financial situation* • *Investment APIs for monitoring performance of investments* • *News tracking APIs for tracking news* →**Orchestration**
Competence	**Certification** – Certified by the appropriate regulatory body in that country, kept up to date with changes	*What about regulation?* →**Legal Perimeter & Certification Positioning**
	Good judgement – Makes consistently good decisions	*Can the AI give consistently good and accurate advice:* • Calculations • Suitability →**Prompt engineering**
	Systems thinking – Must be complexity literate and able to manage multiple aspects of a client's financial life	*Can the AI maintain clarity even with a complex financial situation?* →**LLM Orchestration**
		→**Memory management**
Integrity	**Fiduciary duty** – Provides unbiased recommendations aligned with the client's goals	*Can we validate that the AI consistently acts in the client's best interest?* →**AI alignment**
	Data Security and Privacy – They maintain confidentiality, ensuring the client's sensitive financial information is always protected.	*Can we ensure data is kept secure within an LLM based architecture?* →**LLM data security**
	Accountability – Responsible for actions, decisions, and their outcomes. Able to explain to the regulator if necessary.	*Can we ensure full conversation observability is built into architecture?* →**Observability**

Section 6

Research Findings

Introduction

To explore whether AI can serve as a financial advisor, we covered the following core research areas:

- Multi-Modal Conversation
- LLM Orchestration Framework
  (including memory management, data security and observability)
- Evaluating voice models
- Prompt Tuning *(including AI alignment)*

We also made plans for:

- Client attunement
- Regulation & Compliance

Delivering Multi-Modal Conversation

Can our AI share charts and interactive tools in the midst of a human-sounding conversation?

Delivering a consistent and understandable AI-driven financial advisory experience requires more than just text output. In practice, effective conversations blend multiple modalities – explanatory text, visual graphs, and interactive dialogue prompts with predefined options – to communicate complex financial insights. For instance, an AI advisor might verbally explain a portfolio's performance, display a trend graph of investment returns, and then present the client with follow-up options (e.g., "Rebalance Portfolio" or "View Risk Analysis") to guide the next steps. The challenge lies in integrating these modalities smoothly so that the conversation remains coherent, accurate, and natural for the user. Achieving this with a large language model (LLM) like GPT-4o involves ensuring the model can handle tool usage (for calculations or graph generation) and structured prompts without confusing the flow.

Recent research on augmented LLMs suggests that combining reasoning with tool use can improve an AI's consistency and capabilities. In theory, giving a model access to tools (calculators, databases, graphing functions, etc.) should help it provide factual, context-rich advice beyond its native knowledge. Likewise, structuring the model's outputs (for example, in a JSON format for graphs or choices) can enforce consistency and correctness in multi-modal responses. Our experiments evaluated three approaches to realize these benefits in a conversational setting. We measured each approach's ability to successfully achieve the conversation's goals, the speed and fluidity of the dialogue (especially for voice interfaces), and error rates like tool misuse or format mistakes. The goal was to find a method that delivers a high-quality financial conversation – with rich content and interactive elements – without sacrificing naturalness or reliability.

Approaches Tested

1. Native LLM Tooling

The first approach kept the conversation flow entirely in natural language while letting GPT-4o invoke native tools behind the scenes as needed. GPT-4o has the capability to call integrated tools (such as calculators for math, knowledge look-up, or graph-drawing functions) to augment its responses. This technique aligns with prior research showing that LLMs augmented with a suite of external tools can solve a wide range of tasks more effectively. In our financial advice scenario, the model could autonomously decide to use these tools when appropriate – for example, computing a personalized portfolio metric or generating data for a chart – and then continue the conversation with the result.

In practice, this native-tooling approach worked well up to a point. With a moderate number of tools (around 4–5) available, GPT-4o's performance remained strong and it maintained a coherent dialog, seamlessly incorporating tool results into its answers. However, as we added more tools beyond that threshold, we observed growing integration issues. The model began to confuse or misuse tools in about 20% more cases (e.g., invoking the wrong tool or unnecessary tools), and the success rate of completing the desired conversational goals dropped sharply (around 60% beyond 5-6 tools). Giving the model too many built-in tools created uncertainty in its decision process – it struggled to choose the right tool at the right time. This finding is consistent with the idea that while multiple tools can vastly expand an AI's abilities, the complexity of managing them can also increase the chance of errors if not carefully constrained. The native tooling approach thus offered high conversational naturalness (the user experiences a fluid, human-like chat) but showed limitations in consistency and goal completion once the toolset grew large.

2. Fully Structured Output

The second approach enforced a fully structured output format for each response. In this design, every turn of GPT-4o's output followed a rigid template specifying all elements of the reply (narrative text, any graph data, and the exact dialog prompt options for the user). For example, the model might output a JSON or annotated text block containing a summary, a formatted data section for a chart, and a list of menu options for the next question. The idea was to guarantee that nothing would be omitted or malformed – effectively hard-coding consistency into the conversation. This method is akin to the “function calling” capability introduced in modern LLMs, where the model is guided to produce a JSON result that an external system can reliably use. Such structured generation ensures all needed content (text, visuals, choices) is present and can be precisely interpreted by the application hosting the conversation.

In our evaluation, the structured output approach did prove highly reliable in achieving conversational objectives. The model invariably produced the expected fields for graphs and prompts, making it easy to render multi-modal content without missteps. We saw a clear reduction in logical errors – the AI rarely went off-script or gave inconsistent answers, since the format itself kept it on track. However, this reliability came at a significant cost to user experience. The enforced structure made the dialogue feel robotic and slow. Responses were roughly three times slower (slower to generate output) than the free-form approach. In a voice interface, this proved especially problematic – the assistant would pause and output in a stilted manner, which is not ideal when speaking to a user in real-time. Users expect a conversational cadence, but the structured outputs sounded like the AI was “reading a form” rather than having a chat. This lack of natural flow undermines the interpersonal aspect of financial advising. Thus, while fully structured output met the technical requirements of multi-modal content, it was unsuitable for a natural dialog experience in our use case.

3. Hybrid Approach

The third approach combined the strengths of the first two while minimizing their weaknesses – a hybrid model. We limited the number of native tools accessible to GPT-4o (to avoid overloading its decision-making), and we introduced a custom lightweight syntax in the system prompt that the model could use to trigger external tools or structured actions when necessary. The model was trained/prompted to know that certain special tokens or tags in its output would signal the system to perform a specific action (like generate a graph or present options), without the model having to explicitly format the entire response rigidly. We also explicitly defined which steps of the conversation required a structured output segment. For example, when a graph needed to be created, GPT-4o would output a short, structured snippet (with the data or parameters for the graph) inside special tags, but for the rest of the conversation it would continue in free-flowing natural language. Similarly, for multiple-choice prompts, the model could output a list of options in a clearly marked format only at the moment it is needed. All other dialogue remained unstructured and conversational.

This hybrid approach yielded excellent results – it achieved a nearly seamless integration of text, visuals, and interactive prompts. The error rate dropped to only about 3-5%, meaning the model almost always used the tools correctly and produced the expected structured snippets without mistakes. This low error rate indicates a high consistency, matching the reliability of the fully structured method but without sacrificing natural flow. Most of the time, the AI spoke just like it normally would, maintaining a friendly and fluid conversation. Only when a structured element was absolutely required did the underlying syntax briefly surface, and those instances were handled almost perfectly. The conversation remained quick and felt human-like, since the model was not encumbered by an always-on formatting framework. In fact, this approach mirrors techniques in recent AI research where reasoning steps are combined with actions to improve outcomes. By interleaving free-form reasoning with controlled actions, the model can stay on track and accurate – our results confirm that explicitly guiding GPT-4o only at crucial junctures (and letting it be creative the rest of the time) strikes the best balance. The hybrid model effectively delivered the intended multi-modal content (explanations, graphs, option prompts) in a way that was both highly reliable and natural sounding.

Results and Comparison of Approaches

In summary, our experiments showed distinct trade-offs for each approach. The native tooling method preserved a natural conversational experience and leveraged GPT-4o's ability to use tools on the fly, but it struggled with consistency when too many tools were introduced (leading to increased errors and lower success in completing tasks). The fully structured output method excelled in driving the conversation to predetermined goals with precision, yet it did so at the expense of speed and conversational naturalness – an outcome ill-suited for interactive, especially voice-based, advising. The hybrid approach emerged as the most promising, combining tool use with selective structuring to keep errors minimal while largely retaining a human-like dialogue flow.

The table below compares the three approaches across key dimensions (success rate in achieving the conversation's goals, response speed, conversational naturalness, and error rate):

Approach	Goal success Rate	Speed	Conversational Naturalness	Tool/format Error Rate
Native LLM Tooling	High with a small toolset; drops ~60% when tool count is high.	Fast until too many tools introduce overhead.	Very high – outputs feel like a natural chat.	~1-2% (tool misuse rises ~20% beyond 5 tools).
Fully Structured	Very high – consistently hits objectives due to enforced format.	Slow – ~3× slower responses than natural output.	Low – dialogue feels robotic and not ideal for voice.	Low format errors (rigid template).
Hybrid Model	High – ~97% success (combines reliability with flexibility).	Fast – nearly on par with native conversation.	High – mostly natural flow, minor format intrusions.	~3-5% – minimal tool or formatting mistakes.

As shown above, the hybrid model delivered the most balanced performance across all criteria. It maintained a success rate close to the structured approach while keeping the speed and natural feel comparable to the native tool usage. The error rate was an order of magnitude lower than in the pure native tooling method, indicating a major improvement in consistency. Overall, this comparative evaluation made it clear that the hybrid strategy was the optimal choice for our multi-modal financial assistant scenario.

Outcomes and Next Steps

Based on these findings, we have taken several steps to implement and further develop the solution:

Adoption of the Hybrid Model: We adopted the hybrid approach as the default for our AI financial advisor's multi-modal conversations. This means GPT-4o now delivers advice using primarily natural language, augmented with graphs and choice prompts via the controlled syntax triggers. Users experience a smooth dialogue with rich content (visuals and interactive options) integrated seamlessly.

Tool-Use Evaluation in Model Assessment: We have updated our model evaluation framework to include dedicated checks for tool usage and multi-modal output, especially for voice-enabled interactions. Rather than evaluating the AI's responses on text accuracy alone, we now actively measure whether it invokes tools appropriately and how it handles the structured parts of the output. This ensures that the conversational AI not only "knows" the financial domain, but also effectively manages the graphs and prompts that are essential for a full advisory experience. This focus on tool evaluation as a core metric helps maintain high quality and consistency when the model is deployed in real-world, voice-interactive settings.

Exploration of Step-by-Step Agent Design: Looking forward, we are experimenting with a more advanced step-by-step agent architecture. The idea is to have the AI plan the advisory conversation in dynamic stages – for example, first gathering client goals, then analyzing data, then presenting recommendations – with explicit control over each step. At each stage, the system could inject specialized content or triggers (like creating a graph only when it is most relevant or summarizing options after certain analyses are complete). This design would give the model a form of dynamic step control, potentially reducing complexity by breaking tasks down and ensuring optimal timing for each modality. Early exploration suggests that this could further improve the robustness and clarity of multi-modal financial advice delivered by AI, by making the conversation flow even more intelligently structured behind the scenes, while remaining natural and user-friendly on the surface.

By combining these outcomes – an effective hybrid conversation model in production, improved evaluation techniques, and ongoing research into agent-like planning – we aim to push the boundaries of what AI-driven financial advice can achieve. The ultimate goal is to deliver an advisory experience that feels as coherent, insightful, and responsive as talking to a human financial professional, while harnessing AI's ability to crunch numbers, generate visuals, and personalize advice on the fly. With the hybrid approach proving that AI can indeed deliver high-quality financial conversations in a multi-modal format, we are confident that further refinements will only enhance this capability, bringing us closer to truly trustworthy and versatile AI financial advisors.

Evaluating voice models

Can our AI have a human sounding conversation?

The team set out to choose a text-to-speech (TTS) voice provider for our AI-powered financial advisor product. The voice needed to sound highly natural and human-like to instill user trust, and it also had to offer a unique vocal identity not commonly heard in other AI assistants. The challenge was balancing speech quality (clarity, natural intonation, and minimal artifacts) with brand differentiation (having a voice persona exclusive to the product). Achieving this would enhance user engagement and reinforce the product's brand character.

Approaches Tested

To identify the optimal solution, the team evaluated several state-of-the-art TTS approaches, both proprietary and open source. Each was tested for output naturalness, ease of integration with the AI's text output, latency, and ability to support a custom voice. The approaches included:

OpenAI TTS Models: Open AI released advanced voice capabilities on September 24, 2024. The team tested the 2024 version of OpenAI's internal TTS engines, specifically the tts-1-hd model and a new voice model gpt-4o-mini-tts. These models delivered high-quality audio with very natural prosody and virtually no glitches or artifacts in speech. Integration was seamless with the GPT-4o generated text, meaning the models could directly read the AI's responses without mispronunciations in most cases. There was support for minor adjustments in tone and pronunciation via system prompts (for example, instructing the AI to speak in a calm or enthusiastic manner). However, a major limitation was the lack of support for custom or cloned voices – the OpenAI voices, while pleasant, were generic and could potentially be similar to voices used in other products. This meant they could not easily provide the unique voice persona the team wanted. In summary, OpenAI's TTS had excellent naturalness and easy integration, but offered little flexibility in creating a distinctive brand voice.

ElevenLabs TTS Models: The team extensively evaluated ElevenLabs' text-to-speech offerings, which are well-known in the industry for their realism. Tests were done across the full suite of ElevenLabs models, focusing on the latest versions. The standout choice was the eleven_turbo_v2_5 model, which offered an excellent balance of low latency and high audio quality. Their new offering - Flash v2/v2.5 models delivered ultra-low latency — around ~75 milliseconds response time for generating speech, which is nearly instantaneous. This low latency is crucial for real-time interaction in a conversational financial advisor. The voice output from ElevenLabs was consistently human-like: testers noted that the intonation, rhythm, and emotion in the speech felt very natural, often indistinguishable from a human financial advisor speaking. Another advantage was voice uniqueness: ElevenLabs supports custom voice creation (through its VoiceLab feature), allowing the team to design a voice that would be unique to their product's persona.

One challenge encountered was that ElevenLabs' models had difficulty reading raw GPT-4 output when it contained technical notations. For example, numeric figures, formulas, or any text with LaTeX/Markdown formatting from the AI could trip up the speech (e.g., reading out punctuation or mispronouncing complex numbers). To address this, the team introduced a preprocessing step before feeding text into ElevenLabs:

Text Cleanup: All LaTeX or Markdown syntax (such as math formulas, bullet markers, or formatting characters) was stripped out or converted to plain language. This ensured the input to TTS was clean and speech-ready.

Numerical Formatting: Numbers, dates, and financial symbols were converted into a spoken format. For instance, a raw output "$5,000" would be transformed into "five thousand dollars," and a percentage like "12%" into "twelve percent." This prevented the TTS from attempting to read symbols or large numbers digit-by-digit.

Prompt Tuning: The team adjusted the system prompts given to GPT-4 so that the textual output was already more narration-friendly. The AI was instructed to produce responses in a conversational tone with full sentences (avoiding things like lists of numbers or overly technical formatting unless necessary). This reduced the burden on post-processing.

Optimized Chunking: For very long responses, the text was automatically split into logical chunks before sending to the voice API. This ensured that the TTS could handle the input without speed or small context issues.

With these adjustments, ElevenLabs models produced the most clear and natural-sounding voiceover for the financial advisor, while reading out complex financial information correctly. The voice could be tailored in timbre and style, meaning the team could create a distinctive “AI advisor” voice that users would not confuse with other common AI voices. The only downsides were that this approach required an additional processing layer (as noted) and reliance on a third-party cloud service for TTS. Overall, ElevenLabs Flash v2.5 emerged as a top contender by delivering premium voice quality and uniqueness, meeting both the technical and branding requirements.

Open-Source Self-Hosted Models: The team also explored leading open-source TTS solutions, including Coqui TTS, Bark (by Suno), and an experimental model known as Sesame’s CSM-1B. The motivation was to see if a self-hosted solution could meet the quality bar (which would allow more control and potentially lower long-term costs or data privacy benefits). These open models were chosen for their reputation in research or community: for instance, Coqui TTS is a toolkit with many pre-trained voices and the ability to clone voices with fine-tuning, and Bark is a transformer-based generative audio model known for expressive speech.

In practice, however, none of the open-source options matched the quality or stability of the cloud-based providers (OpenAI and ElevenLabs). Test listeners often detected robotic or less natural intonation, and some models had noticeable artifacts (glitches like odd pronunciations or audio distortions) especially on longer sentences. For example, Bark produced relatively natural sounding speech in short clips, but it sometimes added unintended pauses or sounds, and it was computationally heavy, resulting in higher latency. Coqui TTS allowed more customization (even voice cloning by training on a custom dataset), but achieving the desired quality required significant manual tweaking and training, which was not feasible under our project timelines.

Sesame's CSM-1B, while a promising research model with ~1 billion parameters, was still in an early stage; it required substantial engineering effort to get it running reliably and scaling it to handle many user requests would have demanded a robust (and costly) infrastructure.

Additionally, maintaining these models would mean ongoing engineering overhead – we would have to host GPU servers, optimize model inference speed, handle updates, and possibly retrain models to fine-tune the voice. This was in contrast to the nearly plug-and-play nature of the hosted APIs from OpenAI or ElevenLabs. Given the gap in voice naturalness and the development effort required to close that gap, the open-source route was deemed not production-ready for our needs. It remains a long-term possibility (as open-source TTS technology is rapidly improving), but at the time of evaluation it was not the optimal choice for immediate product deployment.

Outcomes and Next Steps

After careful evaluation, the ElevenLabs Flash v2.5 model was selected as the primary voice provider for the AI financial advisor. This choice was driven by its strong combination of unique voice capability, superior naturalness, and low latency. In our tests, ElevenLabs voices not only sounded convincingly human, but also allowed us to create a signature voice for the advisor – something that would set our product apart from others using more common voice assistants. The near real-time response speed (~75ms) also ensures a smooth interactive experience, which is crucial for user satisfaction in conversational applications.

To integrate ElevenLabs into the product, the team developed a custom text-to-speech integration framework. This mini-framework takes the raw text output from text-to-text LLM model and prepares it for speech synthesis by ElevenLabs.

Next Steps: With ElevenLabs v2.5 in place, the team will move on to fine-tuning the voice experience further. This includes monitoring feedback on the voice quality and persona – for example, ensuring the tone conveys the right level of empathy and expertise expected from a financial advisor. There are plans to experiment with additional voice profiles in ElevenLabs to possibly offer users a choice (for instance, a different voice for different languages or user preferences) while still maintaining that unique feel. The team will also keep an eye on emerging TTS technologies; if open-source models or new providers reach parity with ElevenLabs in quality, they could be incorporated in the future.

Selecting LLM Orchestration Framework

Can the AI maintain clarity even with a complex financial situation?
Does the AI stay aware of the softer client details?
Can the AI use input from various sources to keep track of the client's finances?
Can we ensure full conversation observability is built into architecture?

Building an AI-based financial advice assistant poses unique technical challenges beyond just language understanding. One key decision was selecting a suitable LLM orchestration framework – the software infrastructure to manage prompt flows, tool usage, and multi-step reasoning. The chosen framework needed to handle complex financial queries, integrate external data (market info, user portfolios, etc.), and ensure compliance checks, all while providing a reliable developer experience. In early development, the team evaluated several emerging frameworks (LangChain, Semantic Kernel, Haystack) for this purpose. LangChain stood out during the initial phase due to its maturity and rich ecosystem: by early 2023 it offered integrations with major cloud platforms, APIs, and model providers out-of-the-box. This extensive library of tools and connectors gave it a practical edge for rapid prototyping. Semantic Kernel (an open-source SDK from Microsoft) was also considered – it promised a flexible, enterprise-ready approach to building AI agents across languages, with built-in support for observability and even modalities like voice input. Meanwhile, Haystack (by deepset) provided a robust question-answering framework oriented around retrieval augmented generation, well-suited for document querying. Each had strengths, but given the project's need for a proven, end-to-end solution, LangChain's broader adoption and plugin ecosystem made it the preferred choice initially. The team proceeded with LangChain as the foundation for the assistant, leveraging its community-vetted modules and support.

Fast-forward to 2025: the AI orchestration landscape evolved significantly, and the team re-evaluated their choice in light of new frameworks. The question became whether newer frameworks could better support advanced use cases (like coordinating multiple specialized agents or offering real-time conversation) and improve development efficiency. At this stage, two leading options emerged – LangGraph and

OpenAI's Agents SDK – each representing a different philosophy in LLM orchestration. The following sections outline the approaches tested with these frameworks and how they compare, guiding the decision on the optimal framework for different parts of the financial advisor system.

Approaches Tested

Initial Framework Evaluation (2023)

LangChain – A popular framework for LLM applications, known for its "chain of thought" orchestration. LangChain had rapidly grown an ecosystem of tools, connectors, and community contributions, making it a mature choice. Its design allowed developers to chain prompts and actions (e.g., call an API, then feed result into the LLM) with minimal boilerplate. This maturity and integration breadth gave confidence for production use. However, some complexity in LangChain's abstractions was noted – the flexibility meant a learning curve to optimize chains.

Semantic Kernel – Microsoft's open-source orchestration SDK geared towards enterprise AI solutions. Semantic Kernel offered a modular, extensible approach with support for multiple programming languages (C#, Python, Java). It emphasized reliability (no-breaking changes in v1.0), observability, and security (telemetry, policy filters) from the start. Semantic Kernel's plugin model and planners were attractive for long-term maintainability. Still, in 2023 Semantic Kernel was relatively new and its ecosystem smaller; the team found fewer out-of-the-box tools for financial data, tilting the balance toward LangChain at that time.

Haystack – An established open-source framework for building QA systems. Haystack excelled at retrieval-augmented Q&A: combining language models with document search. It provided pipeline components (retrievers, readers, generators) that could be assembled to handle knowledge base queries. The team considered Haystack for its strong retrieval capabilities, which are important for sourcing up-to-date financial information. Yet, as a general orchestration engine for multi-step dialogues, Haystack was less focused on agent-style tool use beyond search.

Re-evaluation in 2025: New Orchestration Frameworks

By 2025, more specialized orchestration frameworks had emerged to address the limitations of early approaches. The team revisited the framework choice with two promising candidates:

LangGraph (evolution of LangChain): LangGraph builds on LangChain's concepts but introduces a graph-based orchestration paradigm. Instead of linear sequences of prompts and tools, LangGraph represents dialogues and decision flows as a directed graph of nodes (agents or functions) and edges (transitions). This design enables advanced multi-agent workflows and finer control over complex dialogues. In testing, the team found that LangGraph improved manageability for complex tasks – each step or branch in a conversation can be explicitly defined and monitored. This stepwise flow control made it easier to debug and optimize the assistant's behavior (e.g., if a sub-agent handles retrieving financial data, and another summarizes, their interactions can be orchestrated clearly). Another benefit was ecosystem continuity: LangGraph was compatible with many LangChain components, allowing reuse of existing tools and connectors, so the extensive LangChain integrations remained available. External analyses noted that LangGraph is a robust, production-oriented framework with many customizable features. However, the trade-offs became apparent, too. The team encountered some architectural inconsistencies – parts of LangGraph's API still resembled LangChain while others were revamped, leading to a steeper learning curve. Documentation lagged behind the latest features, echoing a common issue from rapidly evolving open-source projects. Certain features felt over-engineered for the needs of a straightforward advisor chatbot. These drawbacks align with industry feedback that LangGraph, while powerful, can be "more complex than necessary" for simpler use cases, introducing additional overhead. In short, LangGraph excelled in control and flexibility, but at the cost of added complexity and some maturity gaps in docs/tooling. Also, it lacks new features from AI providers like Realtime conversation from OpenAI.

OpenAI Agents SDK: The team also evaluated OpenAI's new Agents SDK – a lightweight framework for designing LLM "agents" with native OpenAI model support. This SDK takes a more minimalist approach, aiming to simplify agent development by providing just the essential abstractions. The design is clean and extensible: developers define the agent's abilities (tools or functions it can call) and let the LLM drive the interaction via OpenAI's function-calling interface. The SDK has built-in support for streaming outputs and even voice input/output, leveraging OpenAI's recent advances in speech capabilities. In practice, enabling voice in the Agents SDK was straightforward – the framework can integrate speech-to-text for user queries and text-to-speech for responses without heavy custom code. Another positive was the strong documentation and examples provided by OpenAI, which made onboarding faster. The team was able to stand up a working prototype of the financial advisor agent with less code and complexity compared to LangChain/LangGraph. This agility confirmed the SDK's value for rapid development and iteration. The minimalist philosophy does mean some features are not as "batteries-included." For instance, observability (monitoring the agent's reasoning steps and outcomes) is basic out-of-the-box and their Logs/Traces UI is barely usable. Features like long-term infinite persistent memory or complex multi-step decision policies rely on the developer to implement or plug in external services. The missing advanced features (logging, monitoring, etc.) could be mitigated by integrating third-party tools. They successfully hooked the agent's logs to LangSmith (LangChain's monitoring suite) for tracing and experimented with a logging service (e.g., LogFire) to capture analytics. With these augmentations, the OpenAI Agents SDK proved to be a lightweight yet capable foundation. It especially shines for simple conversational flows – e.g., a user asks for portfolio advice, the agent calls an API via a function call, and streams back a voiced explanation – all accomplished with minimal orchestration overhead.

Comparison of Framework Capabilities

To inform the decision, the team compared LangGraph and OpenAI's Agents SDK across several key performance indicators (with the original LangChain as a baseline reference):

KPI	LangChain (baseline)	LangGraph	OpenAI Agents SDK
Developer Experience	Moderate – established patterns but some heavy abstractions and verbose configuration. Active community support helps.	Mixed – more structured control improves clarity, but new concepts add complexity. Inconsistent APIs can hinder onboarding.	Excellent – very simple to set up an agent with minimal code. Quick iteration with straightforward APIs.
Extensibility	High – plugin integrations for many tools; can customize chains and agents, though some parts are tightly coupled.	High – modular graph nodes allow inserting custom logic; compatible with LangChain tooling for expansion.	High – clean interfaces to add new tools/functions. Lacks some pre-built plugins, but flexible to integrate custom APIs easily.
Performance	Good – thin overhead atop model calls; some latency from chain management but acceptable.	Good – graph orchestration adds slight overhead, but largely efficient.	Excellent – minimal abstraction means almost no overhead beyond direct API calls. Lightweight runtime footprint.
Ecosystem & Integrations	Very rich – dozens of built-in integrations (databases, APIs, knowledge bases) en.wikipedia.org. Mature ecosystem with community contributions.	Leverages LangChain's ecosystem – existing integrations work out-of-the-box, ensuring a wide range of tools available.	Narrower – focused on OpenAI and basic tools. Integrations must be added manually or via function calling spec. Smaller community ecosystem at present.

Observability	High – supports logging and tracing via callback handlers; LangSmith platform provides robust monitoring if used.	High - Relies on Langchain observability support.	Moderate – has built-in basic functionality with OpenAI Logs/Tracing. OTPL support via third party providers. Some third-party frameworks introduced integrations with Agents SDK.
Documentation Quality	Good – extensive docs and examples, though rapid changes caused some sections to become outdated. Community tutorials fill gaps.	Fair – documentation lagged behind releases, making some features hard to learn. Improving but not as polished.	Strong – well-written official docs with clear guides. Simpler scope means fewer concepts to document. Developers report easy adoption.
Suitability for Multi-Agent	Limited – supports single-agent tool use well, but coordinating multiple agents (cooperating LLMs) requires custom logic.	Excellent – designed for multi-agent orchestration with graphs. Natively supports supervising agents and parallel branches for sub-agents.	Moderate – primarily geared toward single-agent scenarios. Has built-in handover and basic multi-agent capabilities.
Voice/Streaming Capabilities	Good – streaming token output is supported in API; voice integration possible via integrations.	Good – inherits LangChain's capabilities; no unique voice features but can integrate speech tools as nodes.	Full – native support for streaming responses and direct integration with OpenAI's speech APIs. Enables voice and real-time conversations out-of-the-box.

Analysis: **The comparison highlights a clear pattern. LangGraph offers the most power** and structure for complex AI orchestration (especially where multiple agents or decision branches are needed), at the cost of higher complexity. OpenAI's Agents SDK provides simplicity and speed, ideal for straightforward interactive flows, but requires augmentation for advanced functionality.

Outcomes and Next Steps

Considering these findings, the team decided on a hybrid strategy: use each framework where it fits best. For the financial assistant's standard conversational workflows – e.g., answering client questions, doing on-the-fly calculations, explaining financial concepts – the OpenAI Agents SDK suites best. Its lightweight nature will reduce development overhead and allow faster iteration on user-facing features. Simpler architecture means fewer points of failure when the assistant is giving real-time advice, and the built-in streaming/voice support aligns well with plans to offer voice-based guidance to users. On the other hand, LangGraph can be utilized for the more complex, offline analytical tasks that the assistant performs behind the scenes. For example, generating a comprehensive financial report by having multiple specialized agents (one fetching market data, one analyzing risk, one composing the summary) can be orchestrated elegantly with LangGraph's graph paradigm. In these scenarios, the extra structure and control are worth the complexity overhead, and real-time speed is less critical than getting a correct, auditable result.

Prompt Tuning for an LLM-based Financial Advisor

Can the AI give consistently good and accurate advice?
Can our AI discuss complex topics?

The team's mission was to enhance the quality, consistency, and reliability of AI-generated financial advice from a large language model (LLM) assistant. Early deployments of the LLM as a financial advisor revealed several issues: some responses were off-mark or contained factual/calculation errors; the tone and persona of the assistant sometimes drifted between conversations; and occasionally the AI would deviate from the prescribed multi-step advisory process. These issues are especially problematic in finance, where trust and accuracy are paramount.

The LLM community is sharing the common opinion that better prompt engineering – i.e., carefully crafting the instructions and persona given to the LLM – could significantly mitigate these problems. By refining the prompt design, they aimed to guide the model's behavior more tightly, improving answer consistency while reducing mistakes. The problem statement was clear: find ways to systematically tune the AI's prompts to improve the quality of its financial advice, enforce a consistent advisor persona, and lower the error rate in responses.

Manual Prompt Tuning and Conversation Testing: The experimentation began with hands-on prompt adjustments and live conversation trials. Engineers manually tweaked the phrasing of system and user prompts, then engaged the LLM in full-length simulated advisory sessions to observe the effects. For instance, they added explicit reminders in the system prompt (e.g., "remember to double-check calculations" or "always respond in a calm, professional tone") and noted how the AI's answers changed. Through these trial-and-error sessions, the team gathered evidence of what wording made the advisor more consistent or accurate. They found, for example, that instructing the model to outline its advice in numbered steps led to more structured outputs, and that emphasizing the advisor's professional persona reduced the incidence of overly casual replies. This manual approach provided invaluable intuition about the LLM's behavior – confirming that even small prompt changes could noticeably alter the output. However, it was also clear that a more systematic, scalable approach would be needed to iterate efficiently beyond just a few tests.

Retool Backoffice UI for Prompt Engineering: To scale up prompt experiments and involve the broader product team, the group built a custom back-office interface using Retool. This internal tool made prompt engineering accessible to both engineers and product managers by exposing key prompt components and settings through a user-friendly dashboard. In the UI, the prompt was separated into two sections: a “Persona” and an “Agent.” The Persona section defined the advisor’s character and tone (for example, “You are FinanceBot, a seasoned financial advisor with a friendly and patient demeanor…”), while the Agent prompt sections contained task-specific instructions guiding the conversation flow (for example, steps to follow when addressing a user’s query). This separation allowed the team to adjust the advisor’s personality independently from its problem-solving approach. The UI also provided controls for various agent parameters – the team could toggle a “supervisor” mode (which involved a second-layer AI agent overseeing the conversation), switch between different underlying LLM models (e.g. choosing GPT-4 for higher quality vs. GPT-4o-mini for cost-efficient testing), select text-to-speech (TTS) voices for the assistant’s spoken output, and enable or disable external tools (such as a calculator for financial math or a web search for market data). All these options could be configured without writing code, and changes took effect immediately for testing.

This empowerment of non-engineers proved invaluable. Product managers and domain experts could directly experiment with prompt wording and agent settings, injecting their expertise into the tuning process. In effect, prompt tuning became a collaborative cross-functional effort – an approach increasingly recognized as best practice. The Retool interface also dramatically sped up iteration: the team would adjust prompts or parameters in the UI, then run a test conversation on the spot to see how the LLM responded, all in one place. This tight feedback loop increased the volume of experiments and brought diverse perspectives into the loop, yielding a richer set of prompt refinements than engineering alone might have produced.

LLM-to-LLM Automatic Evaluations with a User Simulator: As the prompt variants multiplied, the team introduced automated evaluations to objectively measure which prompts performed better across a range of scenarios. They developed a user-simulator system – essentially leveraging one LLM to play the role of a synthetic user interacting with the advisor LLM. Each test scenario was defined via a scenario API: it specified a simulated user persona (including the user's background, goals, and a particular query or task) and sometimes even a scripted temperament or conversation style. For example, one scenario might simulate a cautious investor in their 50's asking about retirement planning, while another might be a young entrepreneur seeking budgeting advice. The advisor LLM, configured with a given prompt version, would then carry out a full conversation with this simulated user. Alongside the simulation, the team crafted evaluation prompts to assess the quality of the advisor's performance after each dialogue. In practice, this meant using an LLM (such as GPT-4 in "judge" mode) to review the conversation transcript and rate it on several key dimensions. This LLM-as-judge approach provided a scalable proxy for human evaluation, which is notoriously hard to do at scale for open-ended dialogue. Because judging a free-form conversation is nuanced (there are many ways to be "right" without exactly matching a reference answer, and style or tone are subjective), the team found it more effective to encode evaluation criteria into an AI grader rather than rely on rigid automated metrics.

In these automated evaluations, the team defined five main criteria to quantify success:

Goal Achievement – Did the advisor ultimately help the user achieve their stated goal or answer their questions effectively? (For instance, if the user wanted a retirement savings plan, did the conversation result in a clear, actionable plan?)

Factual Accuracy – Was the advice correct and grounded in truth? This included checking that any financial facts, figures, or calculations the model provided were accurate and free of hallucinations or mistakes.

Formatting – Did the response follow the desired format and structure? The advisor is expected to present information cleanly (e.g., a step-by-step plan or a bullet list of recommendations) as per guidelines. This metric flagged responses that were jumbled, overly verbose, or missing an expected structure.

Persona Consistency – Did the AI maintain its persona and tone throughout the interaction? The assistant should consistently sound like the seasoned, friendly financial expert defined in the Persona prompt. If the tone shifted unnaturally or the assistant broke character (e.g., by revealing its AI nature or changing style abruptly), it would lose points here.

Step Adherence – Did the agent follow the procedural steps it was instructed to take? The prompts often explicitly enumerated a sequence for the AI to follow (e.g., first gather user info, then analyze options, then present recommendations). This metric checked if those steps were executed in order without skipping or blending them.

After each simulated dialogue, the evaluating LLM produced a report with scores or judgments on these metrics, often accompanied by a brief critique. This allowed the team to quantitatively compare prompt versions. For example, one prompt variant might consistently score higher on Formatting (meaning the answers were well-structured) but slightly lower on Persona Consistency than another variant, indicating a trade-off in tone adherence. By running a battery of diverse scenarios, the team identified which prompt tweaks led to improvements across the board. They also caught regressions early – if a change to improve persona consistency inadvertently caused the model to use more complicated language (hurting the formatting or clarity score), the metrics would reflect that immediately. Over many iterations, this automated LLM-to-LLM testing became a form of ongoing regression testing for the AI advisor's prompt: every new prompt version could be vetted against the same suite of simulated conversations. Notably, employing an LLM as an evaluator in this way emerged as a practical alternative to costly human reviews, allowing fast and repeatable testing at scale. It gave the team confidence that improvements in prompt design were real and measurable.

OpenAI Playground for Prompt Brainstorming: In parallel to formal evaluations, the team made creative use of OpenAI's Playground environment to further refine their prompts. The Playground offers an interactive sandbox for trying out prompts with various models and also includes prompt-generation aids (a feature that can suggest or autocomplete prompts based on user instructions). The team leveraged these tools to brainstorm new prompt phrasing and ideas – essentially, asking GPT itself for guidance on how to best instruct GPT. For example, a prompt engineer might input a high-level request in Playground like: "Generate a system prompt that ensures the AI advisor always explains its reasoning step-by-step before giving recommendations."

The Playground’s generative feature would then propose candidate prompt text or variations to consider. By reviewing these AI-suggested prompts, the team gained insight into how the model internally interprets certain instructions, guiding them to phrasing that aligned more naturally with the LLM’s logic.

Using Playground in this way yielded several concrete improvements. Simply rewording the Agent prompt to be more direct led the LLM to follow the multi-step process more strictly. The team also found that including a specific precaution in the prompt greatly reduced errors: for instance, adding a line like “If any calculation is involved, double-check the math before finalizing your answer.” This prompt tweak prompted the AI to catch its own arithmetic mistakes during testing. In effect, they turned the model into a partner for its own improvement – by using one instance of GPT-4o to suggest optimal prompt styles, they could update the advisor’s prompt to better match the way GPT-4o “thinks.” This iterative Playground-driven brainstorming not only sped up the discovery of effective phrasing but also helped the team optimize prompts to suit GPT’s known strengths and avoid its weaknesses. The result was a noticeable improvement in the model’s step-following behavior (the AI became more reliable at executing instructions in order without omissions) and a decrease in certain categories of errors. In short, by creatively tailoring the prompt to the LLM’s internal patterns, the team was able to coax more aligned and accurate responses from the model.

LangSmith Observability for Prompt Iteration: To tie everything together and monitor performance in real-world conditions, the team integrated LangSmith – an LLM observability and evaluation platform – into their workflow. With LangSmith, every conversation the advisor had (whether in simulation or with real beta users) was logged as a trace, capturing all inputs, outputs, and intermediate reasoning steps. This provided a transparent view into the model’s behavior with each prompt version, giving the team full visibility into how the assistant was functioning. Crucially, they tagged each new prompt iteration in the trace metadata, making it easy to filter and compare conversations by prompt version.

Beyond the metrics, the team could drill down into specific conversation traces to debug nuanced issues. The trace view exposed the full sequence of messages and the model’s internal reasoning, which was invaluable for diagnosing why the model behaved a certain way. The combination of precise logging and LLM-based evaluators meant every prompt change could be measured and validated (or flagged) using both real and simulated data, giving the team a high degree of confidence in the robustness of each new prompt version. In summary, turning on this observability was essential for understanding and debugging the AI advisor’s behavior in depth – it transformed a once opaque system into a glass box, illuminating exactly how prompt changes translated into outcome differences.

Outcomes and Next Steps

The prompt tuning experiments resulted in notable improvements in the LLM advisor’s performance. Conversations with the AI became more natural and consistently on-topic; the flow of dialogue was smoother and more coherent, as the model was less likely to get confused or go off on tangents after the prompt refinements. Error rates dropped significantly. Internal evaluation runs showed far fewer factual mistakes or policy violations than before – by the end of these experiments, the advisor handled even tricky financial queries (the kind that previously tripped it up) with a much higher degree of accuracy. For example, in the retirement planning scenario tests, the proportion of cases where the AI met the user’s goal went up markedly, and the advisor’s answers almost always followed the expected step-by-step format (whereas earlier versions might skip a step or merge steps together). The tone and persona of the assistant also stabilized; users received advice in a reliably consistent voice that matched the intended friendly, professional character. It became clear that, while prompt engineering is not a panacea for all AI shortcomings - the model’s fundamental knowledge and reasoning limits still apply – it nonetheless proved to be a powerful lever for steering the model’s behavior. By iteratively refining the prompts, the team was able to align the AI’s output more closely with the needs of this financial advice domain, all without retraining the underlying model. In effect, prompt tuning bridged the gap between a generic LLM and a specialized financial advisor, bringing performance up to a level that inspired more trust.

Buoyed by these positive outcomes, the team has laid out several next steps to further improve the system. One major area of exploration is dynamic step-based prompting. The idea is to break the consultation dialogue into discrete phases and expose the model only to the relevant instructions and context at each phase, rather than a single monolithic prompt or the entire chat history at once. For example, the conversation could be structured into stages: an “intake stage” where the model only sees the Persona prompt and asks the user questions to gather financial information; an “analysis stage” where, after the user’s info is collected, the model is given a focused prompt containing just that info plus instructions to analyze and formulate a plan (hiding the earlier persona instructions which are already internalized); and a “advice stage” where the model is prompted to present recommendations based on the analysis (perhaps only exposing key points from the analysis rather than the raw chat history). By progressively feeding the model context in this way, the team expects to reduce context dilution and maintain consistency over long dialogues. The model will be less prone to forgetting earlier directives or mixing unrelated information if at each step it only handles a self-contained task. This approach essentially treats the prompt as an evolving script, dynamically constructed as the conversation progresses. Implementing dynamic prompting will require additional orchestration logic to manage the state and transitions between stages, but it promises to further boost the advisor’s consistency and relevance by always keeping the model’s attention on what matters right now in the conversation.

Another frontier the team is moving into is the automated evaluation of real user conversations. Up to now, most evaluations have been with simulated users. The next step is to integrate the evaluation pipeline with actual user chat logs (with appropriate privacy safeguards). The plan is to have a system that continuously or periodically samples real interactions and runs the same metric-based evaluations on them. This will produce ongoing scores for live performance – a kind of running “QA dashboard” for the AI in production. If certain metrics (like factual accuracy or goal achievement) start to dip on user data, the team will catch it early and investigate. Alongside this, there is an idea of self-correction mechanisms for the AI during live use. One idea is to insert a brief reflection phase before the AI finalizes a response: essentially asking the model to review its own answer internally to check for compliance with instructions or obvious errors at critical conversation steps (e.g., calculations). If the model detects a potential mistake, it can correct itself before outputting. This kind of self-monitoring loop, powered by the model's ability to critique or verify its answer, could further reduce error rates without human intervention. Finally, the team is developing alerting tools for undesirable behavior. If the AI advisor ever produces an output that is off-policy or potentially harmful – for example, giving prohibited financial advice or exhibiting a severe persona break – automated alarms will trigger. These might be simple rule-based triggers (keywords or patterns to flag) or more complex evaluations (an LLM judging that "this response sounds unhelpful or unsafe"). When triggered, an alert would notify the developers or a moderation team to review the conversation and act as needed. By implementing these measures, the team aims to create a virtuous cycle of continuous improvement: the model's performance is constantly measured on real interactions, the model is even equipped to catch and correct some of its own mistakes, and the development team is immediately informed of any serious issues that do arise.

In conclusion, this series of experiments shows how prompt tuning can dramatically improve an LLM-based financial advisor and how crucial it is to support the AI with the right tools and processes. Through careful prompt engineering, extensive automated testing, and diligent observability, the team transformed a baseline model into a much more polished conversational agent.

Other factors we have prepared for

Client Attunement

Can the AI talk to clients in a way that meets their personality and emotional needs?

Human advisors build relationships with their clients and talk to them in a way that meets their personality and emotional needs.

To build this into FinleyAI we sought to understand the users' preferences and inject these into the prompt.

There are two approaches to understanding a client's preferences: asking direct questions or gauging them based on their behavior.

In onboarding, we can build on direct questions we ask – a user's age (for e.g., *a 55-year-old may be feeling more stressed about retirement than a 38-year-old*) and their financial experience (*a novice will likely want concepts explained more than someone with experience*). But we also explored the idea of inferring the type of advisor a client might like, and the client's NLP characteristics, from a transcript after an onboarding conversation.

To do this we had a separate agent analyze the client's transcript and look judge for:

- RM preference type (*Length & Complexity of sentences, Colloquial or Formal*) and
- NLP Characteristics (*Optimistic or Pessimistic, Trusting or Guarded, Preference for visual/auditory/kinesthetic metaphors*)

The assessment agent successfully assessed all the transcripts we gave it. The challenge in this is getting the user to speak naturally with the AI early on enough in the relationship for the assessment to be representative. We would seek to research further in this aspect with a wider user base. We anti normalized in the general population.

Regulation & Compliance

Legal Perimeter & Certification Positioning

FinleyAI is not an autonomous financial advisor. It is an AI-enabled software platform that supports financial advice delivery under the direct supervision of a licensed human advisor.

All regulated actions are subject to human-in-the-loop review, with full audit trails, observability, and user transparency.

Domain	FinleyAI Role	Human Advisor Role
Suitability assessment	Structured agent recommendation	Final review and approval
Investment proposals	Draft generation only	Formal sign-off
Account setup	Pre-fills client data	Advisor validates and submits
Cash flow / scenario modelling	Full automation	Optional review
Risk profiling	Structured questions & agent scoring	Review edge cases
News & alerts	Summarized impact	Advisor escalation optional

Regulation

Finley operates under the advisory firm's regulatory umbrella and is positioned as a digital co-pilot—not a discretionary decision-maker. All regulated activities (e.g., investment advice, pension recommendations) are signed off by a certified advisor in the firm. This maintains compliance with regulatory needs from FCA, SEC, and FINRA frameworks around scope of advice, fiduciary responsibility, and liability assignment.

We are aligned with emerging guidance from:

- FCA: Advice vs Guidance clarity, AI-specific reviews in financial services
- CFP Board: Generative AI may support but not replace fiduciary delivery

- SEC/FINRA: Best-interest suitability, data security, and traceability requirements

All Finley deployments are expected to be scoped to remain within these evolving boundaries.

Finley’s 4-step compliance architecture

Our compliance framework is anchored in a 4-Level AI Architecture that ensures every conversation is suitable, auditable, and client-first.

	What it does:	Why it matters:
Level 1: Client Context Layer	Builds a persistent, transparent profile of facts across financial, behavioral, and life context. Includes both hard facts (income, dependents, assets) and soft traits (tone, preferences, NLP-derived behaviors). Users retain visibility and control via the Information Vault	Aligns with Know Your Customer (KYC) principles. Enables suitability and best-interest assessments downstream.
Level 2: Structured Agent Layer	Uses modular, topic-specific financial agents (e.g., retirement, education, cashflow) to guide advice journeys, stringently enforcing defined flows and protocols. Each agent follows regulated decision trees (e.g., risk profiling) with escalation points, as well as specific tooling (e.g., controlled repeatable Python simulations with vetted input assumptions).	Ensures consistent, repeatable conversations. Enables systematic delivery of appropriate advice with human override.
Level 3: Generative Layer	Powers Finley’s conversational interface (voice/text), generating explanations, empathy, and behavioral coaching. Is always constrained by system goals: clarity, client-first tone, and non-discretionary advice. Uses LLM-to-LLM testing of prompts to dramatically reduce human testing time for each agent.	Ensures nothing slips into discretionary or unqualified territory. Reinforces trust, transparency, and emotional attunement in prompts without hours of human input. Cleverly balances efficiency of LLM-to-LLM training while keeping humans for final exploratory tests.
Level 4: Oversight & Controls Layer	Every conversation is logged with full observability (LangSmith + AWS tooling). Built-in compliance agents assess misalignment, flag edge cases, and enforce rewrites. Optional human-in-the-loop checkpoints before any financial action is finalized.	Aligns with audit, supervision, and recordkeeping obligations. Supports hybrid models with IFAs, allowing scalable advice while retaining liability coverage.

Summary Benefits:

Regulatory alignment: Designed to work within UK and US fiduciary advice frameworks.

Human-led model: Finley acts as a co-pilot, not a discretionary advisor.

Scalable safety: From risk profiling to action approval, every stage is explainable and reviewable.

Can the AI pass the CFP test?

FinleyAI is not eligible for CFP, CISI, or equivalent certifications as these are human credentials. However, Finley's logic has been designed to mirror the standards, structure, and scope of certified human advisors. Our goal is to augment, not replace human advisors—enabling a 1:200 ratio versus the traditional 1:20. Future versions may aim to pass CFP-style assessments as internal quality benchmarks, but regulatory certification will always remain with the human.

Conclusion

Our technical research set out to determine whether large language models could deliver a high-quality, compliant, and human-feeling financial advice conversation. This required innovation across several dimensions of AI system design, from orchestration and memory to voice and multi-modal interaction.

Across four key domains, we discovered that:

➔ Hybrid orchestration delivers the best performance.

A mix of natural language and light structural formatting achieved a 97% success rate in multi-modal conversations, avoiding the brusqueness of rigid output while preserving accuracy and flow.

➔ Tool integration must be carefully scoped.

Native LLM tooling works well at small scale but degrades when overloaded. Our approach limited tools and used orchestration to control invocation intelligently, minimizing errors.

➔ Voice design is critical in trust-centric domains.

ElevenLabs Flash v2.5 provided ultra-low-latency, emotionally credible voice output. Combined with preprocessing layers and prompt tuning, it enabled a compelling vocal advisor persona.

➔ Prompt tuning benefits from cross-functional iteration.

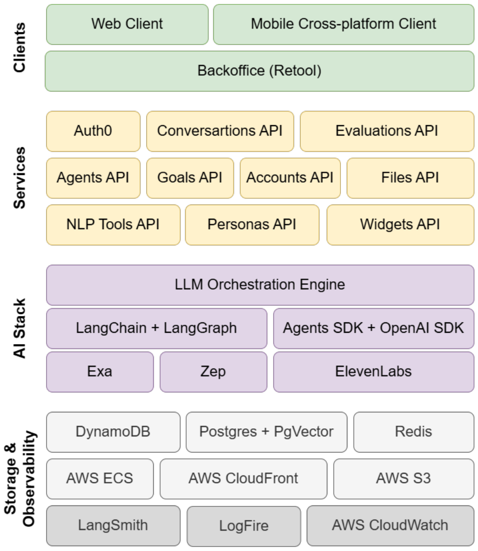
Giving product and design teams direct control through a UI (via Retool) led to stronger advisor personas, more consistent tone, and reduced hallucinations. Our LLM-to-LLM evaluation system accelerated testing by scoring conversations across key metrics: accuracy, tone, structure, and goal completion.

The resulting system represents a functional blueprint for AI-delivered financial advice: modular, auditable, multi-modal, and human-aligned. It would never be implemented with full autonomy, but in a hybrid model—with human review and regulatory safeguards—it could augment real financial advisors today, allowing them to serve more people, and start closing the financial advice gap.

Appendix 1

Technical diagrams

Finley Architecture



Clients

Web Client | Mobile Cross-platform Client

Backoffice (Retool)

Services

Auth0 | Conversartions API | Evaluations API

Agents API | Goals API | Accounts API | Files API

NLP Tools API | Personas API | Widgets API

AI Stack

LLM Orchestration Engine

LangChain + LangGraph | Agents SDK + OpenAI SDK

Exa | Zep | ElevenLabs

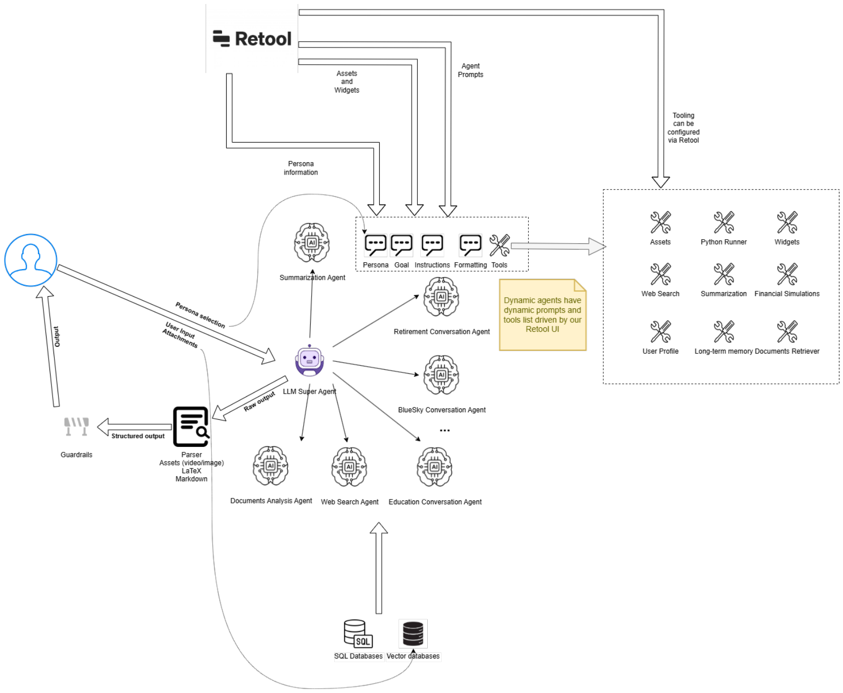
Storage & Observability

DynamoDB | Postgres + PgVector | Redis

AWS ECS | AWS CloudFront | AWS S3

LangSmith | LogFire | AWS CloudWatch

AI Agents Architecture



Dynamic agents have dynamic prompts and tools list driven by our Retool UI

Appendix 2

Timeline

How FinleyAI Was Engineered to Match Human Advisor Standards

To move beyond transactional chatbots, we began by mapping the traits of highly rated financial advisors. Then we engineered Finley's AI systems to match each of these traits through design, orchestration, and testing. Below is the trace of that journey, alongside industry events as we navigated a landscape that shifted in real time.

❖ Good Communication

Clarity							
	Q4 2023	Q1 2024	Q2 2024	Q3 2024	Q4 2024	Q1 2025	STATUS
Can our AI have a human sounding conversation? →Voice Model Evaluation	Researching early voice experiences in the market Industry: Basic Voice Mode on ChatGPT available ElevenLabs expands support for multilingual and emotional tone modulation Initial release of Bark as open source by Suno AI ElevenLabs improved voice cloning capabilities and emotion-aware synthesis		Evaluation of TTS Providers Tested ElevenLabs TTS Models	Industry: OpenAI's Advanced Voice Mode released Tested the OpenAI TTS Models Fine-tuning Voice Experience (text cleanup, numerical formatting, prompt tuning for voice, and optimized chunking) Eleven 2.5 released, featuring real-time emotion modeling and improved synthetic prosody for voice AI.		Open source TTS model Sesame CSM-1B released Tested Open-Source Self-Hosted Models Eleven 3.0, with real-time adaptive synthesis and AI character voices	Achieved **Best in class voice experience supported through ElevenLabs Flash v2.5** Selected for its unique voice capability, superior naturalness, and low latency Next: Continue to test emerging models and voice tech
Can our AI share charts and interactive tools during the conversation? →Multimodal conversations	Testing calculations within conversations – no multimodal		Tested Native LLM Tooling (Tool misuse rises ~20% beyond 5 tools) Designing interactive widgets & graphs Introduction of GPT-4o (Omni), OpenAI's first natively multimodal model	Tested Fully Structured Output (Very high success rate, but robotic)	Developed and adopted a Hybrid Approach (balances achieving the conversation goals– response speed, conversational naturalness, and error rate)	Added 'Tool-Use' to Future Model Assessment Framework	Achieved **The hybrid model we developed serves graphs/widgets with ~97% success rate** while maintaining natural and pacey conversations. Next: Experimenting with a more advanced step-by-step agent architecture
Can our AI	Hands-on Prompt Experiments			Development of	LLM-to-LLM	Continued	Achieved

discuss complex topics? →Prompt engineering			prompt management system in retool Optimization of prompts for voice Dynamic system prompt generation per each conversation iteration	Automatic Evaluations with a User Simulator Used OpenAI Playground for Prompt Brainstorming	experiments	**The prompts tuning and orchestration allow for financial advice conversations across a range of products and situations** Next steps: dynamic step-based prompting
Attunement						
Can the AI talk to clients in a way that meets their personality and emotional needs? →NLP prompting	-	Tested assessments of Behavioral Loss Tolerance, RM Preference and NLP traits from conversational text Tested injection of these traits into dynamic system prompt generation				Achievable **User preferences can be injected into prompts. Preferences can be gathered during onboarding or inferred by agents analyzing client transcripts.** Next steps: More comprehensive testing
Does the AI stay aware of the softer client details? *→Memory management*	Experiments with conversation storage and retrieve engines	Created automated facts extraction from conversations to fill-in user profile	Conversation summaries implemented		Optimized relevant facts extraction	Achieved **The User Profile builds the view of the user – hard facts and soft facts.** Our system is optimized to extract the most relevant facts.
Proactivity						
Can the AI use inputs from various sources to keep track of the client's finances? *→Orchestration*		Build of our tech architecture Design of personalized news alerts, which helps the user understand the world through the lens of their money.		Research into Openbanking API <> LLM interactions		Achievable **The Finley architecture is able to act proactively based on the inputs it gets** from Investment APIs, Openbanking and even News APIs as well as scheduled check-ins, and work with the Agent architecture to deliver this as personalized actions.

❖ Competence

Certification							
	Q4 2023	Q1 2024	Q2 2024	Q3 2024	Q4 2024	Q1 2025	STATUS
Can this system pass the CFP? And keep up to date with regulatory changes? *→CFP Test*	Decision that Finley would always be positioned as a digital co-pilot, not a discretionary decision-maker. All regulated activities (e.g., investment advice, pension recommendations) are signed off by a certified advisor in the firm. Finley is not eligible for CFP, CISI, or equivalent certifications as these are human credentials.						Achievable **Regulatory certification will always remain with the human**.
Good Judgement							
Can the AI give consistently good and accurate advice: *→Prompt engineering*	Experiments with LLM-driven calculations using Python			Pre-coded financial forecasting calculations as LLM tools. Experiments with newest models and LLM-driven calculations with/without Python. Variables extraction enabled consistent calculations without hallucinations or data identification.			Achieved **Tests showed consistently good calculations** and decision making without hallucinations or issues with data identification
Systems thinking							
Can the AI maintain clarity even with a complex financial situation? *→LLM Orchestration*	Tested: LangChain, Semantic Kernel, Haystack Experiments with conversation storage and retrieve engines Built with LangChain Industry: Orchestration framework Langgraph released for streamlining agent workflows			Langgraph v0.2 released - increased customization with new checkpointers. Migrated to LangGraph		OpenAI Agents SDK and PydanticAI experiments Langgraph 0.3 released. By Q1 2025, LangGraph Server (APIs), LangGraph SDKs (clients for the APIs), LangGraph CLI (command-line tool), and LangGraph Studio (UI/debugger) available.	Achieved **Our agent orchestration and memory management maintain a constant context of the user and their finances** and is able to prompt and navigate conversations with that context.
Can the AI maintain a long conversation without losing context or important facts *→Memory management*	Tested: LangChain, Semantic Kernel, Haystack Experiments with conversation storage and retrieve engines	Extensive experiments with vector databases	Create an ultra-fast conversation history retrieval mechanism based on key-value storage		Industry: ChatGPT memory rolled out broadly		

❖ Integrity

Fiduciary Duty							
	Q4 2023	Q1 2024	Q2 2024	Q3 2024	Q4 2024	Q1 2025	STATUS
Can we validate that the AI consistently acts in the client's best interest? *→AI alignment?*		Prompt engineering: Finley is guided to always work in the interest of a client.		Continued research on the subject of AI alignment			Achievable Finley is guided to always work in the interest of a client, but most importantly **our 4-step compliance architecture provides regulatory grade guard rails**
Data Security and Privacy							
Can we ensure data is kept secure within an LLM based architecture? →LLM data security?	Experiments with LLM to SQL security measures	Experiments with LLM Guardrails and LLM tooling security Implement initial prompt sanitization to strip personally identifiable information (PII)	Agents architecture implemented with client data isolation. Architectured AI guardrails to detect and prevent sensitive data leakage from AI outputs.		Implement isolated vector search mechanisms to ensure no cross-client data leakage occurs.		Achievable **Finley does not have knowledge of other clients when talking to a particular one. Strict security and audit controls**. For productionizing we would need to formally implement OWASP LLM Security Verification Standard implementation and achieve SOC 2 Type II certification.
Can we provide users with understanding and control of their data? *→Design*	Benchmarking and user research			Design of information vault developed			Achieved Information Vault gives user complete visibility and control of their data
Accountability							
Can we ensure full conversation observability is built into architecture? *→observability*		With LangChain orchestration we get logging and tracing via callback handlers	Experiments with LangFuse	LangSmith deployment means we support a high level of traceability	Implemented metadata system for agents and conversations	LLM Metrics dashboard implemented with alert system	Achieved **High level of conversational traceability** provided with LangSmith Conversations stored in DynamoDB. Plus, with our 4-step compliance architecture accountability is with the Advisory house.

Appendix 3

ChatGPT Testing

Testing completed in Q3 2024	Chat GPT	Custom GPT	Third Party GPT
Good Communication			
Clarity – Simplifies complex financial jargon into easy-to-understand terms during consultations	🟨 Good at explaining complex terms if specifically asked but can forget to follow through on checking in on user understanding in complex conversations with lots of details. Can overwhelm user with info and multimodal is limited.	🟨Good at explaining complex terms. Communication style is reliant on the prompt the user has created.	❌ NOT PERMITTED by OpenAI
Attunement – consistently aware, responsive, and mindful of the client's needs, preferences, and experiences, not just in isolated moments, but across the whole relationship.	🟨 Can adjust their communication style if specifically asked. Standard communication is very good, but not necessarily personalized.	🟨Can prompt CustomGPT to ask demographic information and adjust communication accordingly, but a) user has to have this knowledge off the bat and b) difficult to orchestrate other personalization entirely in system prompt.	
Proactivity – Offering guidance without being asked, Constant monitors the client's financial situation and acts accordingly	❌ Cannot monitor financial situation. Individuals are discouraged from sharing personal banking details and information such as SSN on the platform. ChatGPT has limited ability to proactively contact the user.	❌Individuals are discouraged from sharing personal banking details and information such as SSN on the platform.	
Competence			
Certification – Certified by regulator, up to date	❌	❌	❌ NOT PERMITTED by OpenAI
Good judgement – Makes consistently good decisions	❌ Occasionally asked, but not consistently	🟨If included in the prompt, it can follow, but not 100%	
Systems thinking – Must be complexity literate and able to manage multiple aspects of a client's financial life	🟨 Does well with remembering multiple goals and being able to summarize them at the end of a conversation. Memory is not always accurate across chats	🟨Does well with remembering multiple goals and being able to summarize them at the end of a conversation. Memory is not always accurate across chats.	
Integrity			

Fiduciary duty – Provides unbiased recommendations aligned with the client’s goals	❌ A ‘black box’ so user can never be sure of LLM’s alignment	❌A ‘black box’ so user can never be sure of LLM’s alignment	❌ NOT PERMITTED by OpenAI
Data Security and Privacy – Maintain confidentiality, ensuring the client's sensitive financial information is always protected.	🟨 While ChatGPT provides a level of data security, given it is not regulated as a financial advisory we cannot know what the business may do with the data (now or in future)	🟨While ChatGPT provides a level of data security, given it is not regulated as a financial advisory we cannot know what the business may do with the data (now or in future)	
Accountability – Responsible for actions, decisions, and their outcomes. Able to explain to the regulator if necessary.	❌ the user must complete account opening elsewhere	❌the user must complete account opening elsewhere	

Sources

Executive Summary

[1] Kitces, Report on *How Financial Planners Actually Do Financial Planning, Page 106* https://www.kitces.com/kitces-report-how-financial-planners-actually-do-financial-planning/

[2] National Financial Educators Council, 2023 & 2024 *Financial Literacy Surveys* https://www.financialeducatorscouncil.org/financial-illiteracy-cost/

[3] Federal Reserve, Report on the *Economic Well-Being of U.S. Households in 2023* – https://www.federalreserve.gov/publications/2024-economic-well-being-of-us-households-in-2023-retirement.htm

Section 1

[1] Lin JT, Bumcrot C, Mottola G, et al. *Financial Capability in the United States: Highlights from the FINRA Foundation National Financial Capability Study (5th Edition)*. FINRA Investor Education Foundation; 2022. www.FINRAFoundation.org/NFCSReport2021

[2] OECD (2023), "OECD/INFE 2023 International Survey of Adult Financial Literacy", *OECD Business and Finance Policy Papers*, No. 39, OECD Publishing, Paris, https://doi.org/10.1787/56003a32-en.

[3] Kitces, Report on *How Financial Planners Actually Do Financial Planning, Page 106* https://www.kitces.com/kitces-report-how-financial-planners-actually-do-financial-planning/

[4] Sall D. 100+ savings statistics 2025: averages, rates, and by age. MoneyZine. Published February 14, 2024. Accessed April 24, 2025. https://moneyzine.com/personal-finance/savings-statistics/

[5] Adam J. How much does the average American have in savings? Forbes Advisor. Published March 2024. Accessed April 24, 2025. https://www.forbes.com/advisor/banking/savings/average-american-savings/

[6] Deaton H. Americans Without Advisors are More Stressed. So Why Aren't They Engaging with Financial Professionals? Institutional Investor. Published December 2022. Accessed May 12, 2025.

https://www.institutionalinvestor.com/article/2azn9e8r9k3izwmsmcfls/ria-intel/americans-without-advisors-are-more-stressed-so-why-arent-they-engaging-with-financial-professionals

Section 2

[5] Vanguard Group. (2019). *Assessing the Value of Advice*. Link
Vanguard Group. *Advisor's Alpha*. Link

[6] CFP Board. (2018). *Code of Ethics and Standards of Conduct*.

[7] Blanchett, D. (2013). *The Value of Financial Advice*. Morningstar.

[8] Financial Planning Association. (2012). *Communication Behavior and Relationship Building in Financial Planning*. Journal of Financial Planning.

[9] J.D. Power. (2023). *U.S. Financial Advisor Satisfaction Study*.

[10] FINRA Foundation. (2018). *Uncertain Futures: 7 Myths About Millennials and Investing*