European Research Studies Journal
Volume XXVIII, Issue 3, 2025

pp. 1201-1217

Prompt Engineering in Finance: An LLM-Based Multi-Agent Architecture for Decision Support

Submitted 15/08/25, 1st revision 29/08/25, 2nd revision 11/09/25, accepted 30/09/25

Artur Kulpa[1], Grzegorz Wojarnik[2]

Abstract:

Purpose: *This article systematizes prompt engineering techniques for finance and proposes a novel, LLM-based multi-agent architecture to enhance the quality, reproducibility, and auditability of automated decision support. The goal is to create a framework that ensures outputs are not only accurate but also transparent and compliant with stringent regulatory requirements.*

Design/Methodology/Approach: *This is a conceptual paper that puts forward a decision-support architecture built on four pillars: (i) enforced multi-path reasoning through Chain-of-Thought (CoT) and self-consistency; (ii) grounding outputs in a curated corpus via Retrieval-Augmented Generation (RAG); (iii) a structured dialogue between agents with specialized roles (Strategist, Critic, Moderator); and (iv) strict verification guardrails and auditable output formats. The methodology integrates proven techniques into a coherent system designed for financial applications.*

Findings: *Since the study is conceptual, the findings are presented as operational hypotheses supported by existing literature. The proposed architecture is expected to yield: (A) greater accuracy and stability of reasoning; (B) a significant reduction in hallucinations and improved provenance due to RAG; (C) more effective detection of errors and analytical blind spots through the multi-agent workflow; and (D) increased auditability and regulatory readiness via standardized, verifiable outputs.*

Practical Implications: *The framework offers financial institutions a clear path toward developing more reliable AI tools. Its implementation can lead to higher quality "first drafts" of decisions, fewer subsequent corrections, and shorter audit cycles. This approach has direct applications in areas such as credit analysis, risk management, and compliance monitoring, promising faster processing with more robust documentation.*

Originality/Value: *The paper's main contribution is the synthesis of several distinct lines of LLM research-prompt engineering, RAG, and multi-agent systems-into a single, coherent architecture tailored to the specific needs of the financial sector. It addresses a critical gap by providing a systematic blueprint for building explainable and auditable AI decision-support systems in a highly regulated environment.*

Keywords: *Large Language Models (LLMs), prompt engineering, multi-agent systems, financial decision support, artificial intelligence in finance.*

[1]*University of Szczecin, Poland, ORCID: 0000-0002-6739-2606,*
e-mail: artur.kulpa@usz.edu.pl;
[2] *University of Szczecin, Poland, ORCID: 0000-0001-6946-547X,*
e-mail: grzegorz.wojarnik@usz.edu.pl;

JEL codes: *G17, G32, C80, C45, O33.*
Paper type: *Conceptual paper.*
Acknowledgement: *This research was co-financed by the Minister of Science under the "Regional Excellence Initiative.*

1. Introduction

Dynamic advances in large language models (LLMs) in recent years are reshaping the landscape of automating analytical and decision-making tasks in finance. The sector is marked by high informational complexity, stringent regulatory oversight, and a need for transparent, auditable rationales for decisions. In this setting, prompt engineering-understood as the design of roles, rules, and formats for interaction with the model-becomes a key determinant of response quality.

In addition, connecting LLMs within multi-agent architectures (with clearly defined roles and dialogue protocols) offers a promising path to improving the accuracy, coherence, and reproducibility of reasoning. Empirical work on chain-of-thought and self-consistency shows substantial gains on reasoning-heavy tasks, while retrieval-augmented generation (RAG) improves factual grounding and enables source tracing; frameworks such as AutoGen and CAMEL illustrate how to program role collaboration and rules of debate in multi-agent systems.

This article adopts a conceptual perspective: we propose and systematize principles for prompt design and a protocol for LLM-agent collaboration to support decision-making in finance. At the core is a Strategist-Critic-Moderator triad, in which system quality derives from: (i) enforced multi-path reasoning (chain-of-thought with self-consistency), (ii) grounding outputs in a curated corpus (RAG), and (iii) auditable output formats and rules of interaction among agents. The structure of the paper, the research questions, and the emphasis on prompt engineering and multi-agent architecture follow from the assumptions and scope set out in the source document.

The paper is review-cum-proposal in character. We reference literature documenting the effectiveness of specific techniques (CoT, self-consistency, RAG, role-playing/multi-agent) and present a coherent scheme for deploying them in financial settings. At the same time, we stress that our conclusions constitute

operational hypotheses requiring empirical verification in production applications (on institutional data and within institutional compliance procedures).
This structure links the latest findings in LLM research with the specific demands of finance, where decision-making must be not only effective but also compliant with supervisory requirements and straightforward to replicate and audit.

Article objectives are as follows:

- *to review and systematize prompt-engineering techniques relevant to finance (roles, rules, formats, CoT, self-consistency, RAG, guardrails);*
- *to formulate a role-based, multi-agent decision-support architecture built around the Strategist-Critic-Moderator triad together with its information flow;*
- *to specify quality requirements for prompts and outputs (factuality, provenance, numerical consistency, uncertainty calibration, auditability).*

Research hypotheses for empirical verification:

H1: Structural prompt design (roles + CoT + self-consistency) increases the accuracy and stability of LLM responses in decision tasks relative to generic prompts.

H2: Incorporating RAG with mandatory provenance (citations, dates) reduces hallucinations and improves the audit acceptability of results.

H3: A Strategist-Critic-Moderator architecture with clearly defined interaction rules enhances the coherence and reproducibility of decisions compared with a single-agent setup.

Building on these premises, the remainder of the paper presents a synthetic literature review, a detailed methodology (including a prompt library and agent-debate protocols), followed by a discussion of the conceptual results and limitations that delineate directions for empirical research and implementation requirements in financial institutions.

2. Literature Review

The literature on large language models (LLMs) increasingly converges on three strands that are especially salient for financial applications: prompt-engineering techniques, multi-agent architectures, and case studies/reviews of deployments in the financial sector-together with catalogs of risks and regulatory requirements. Below, we present a synthetic state of the art and identify research gaps relevant to the design of empirical studies on portfolio decision-making.

2.1 Prompt Engineering-Approaches and Classifications

Prompt engineering is a set of techniques for constructing queries and model working context that systematically improve reasoning quality, mitigate

hallucinations, and enhance compliance with domain requirements (e.g., finance). Contemporary treatments organize this space into role and template design, reasoning-support techniques (e.g., chain-of-thought and self-consistency), integration with retrieval (RAG), and guardrails-safety and compliance rules applied at system input/output (Liu *et al.,* 2023).

In role prompting, assigning a role to the model (e.g., "investment analyst," "risk controller") structures tone, scope, and evaluation criteria, which fosters coherence and task relevance in domain settings. In practice, roles are paired with additional constraints (checklists, table formats, acceptance criteria) as well as with RAG and guardrails, so that the role not only "sets the tone" but also enforces expected procedures (Liu *et al.,* 2023).

Chain-of-thought (CoT) encourages the model to articulate its reasoning step by step (a "scratchpad"), markedly improving performance on arithmetic, logical, and commonsense tasks. Prior work shows that adding a few CoT exemplars to the prompt (few-shot with worked reasoning) can raise results to state-of-the-art levels on GSM8K and other benchmarks, especially for larger-capacity models (Wei *et al.,* 2022).

In practice, CoT is combined with simple patterns: a brief instruction ("solve step by step"), 1-3 worked examples, and a fixed output schema (e.g., Steps - Answer), which facilitates verification and downstream processing. CoT is typically applied selectively (e.g., only to tasks requiring justification) to contain costs and generation length.

Self-consistency is a decoding strategy in which the model generates multiple reasoning paths and the answer most supported by independent derivations is selected. This approach substantially amplifies CoT-e.g., +17.9 percentage points on GSM8K-and yields clear gains on SVAMP, AQuA, StrategyQA, and ARC-Challenge (Wang *et al.,* 2023). In financial applications it is often paired with verification rules (e.g., re-computing figures in a spreadsheet) and heuristics that filter out internally inconsistent outputs.

Retrieval-augmented generation (RAG) links the model to an external "memory" (e.g., corporate document corpora, regulatory databases, data services), enabling generation that references up-to-date sources and returns citable justifications. A classic study (Lewis *et al.,* 2020) showed that this approach improves factuality and specificity on knowledge-intensive tasks. In finance, RAG supports, inter alia, attaching excerpts from periodic reports, market quotations, or supervisory guidance as the basis for answers.

Guardrails are programmable safety rails at system input and output: PII filters, content policies, jailbreak/prompt-injection defenses, enforced data formats and schemas, and even orchestration of validation steps (e.g., "first cite sources, then

provide a recommendation"). Frameworks such as NVIDIA NeMo Guardrails and projects like Guardrails AI provide ready-made rule engines, validators, and specification languages that can be inserted between the application and the model (NVIDIA, 2024; Guardrails AI, 2025). In regulated contexts (finance), guardrails are commonly combined with RAG (to justify claims with sources) and self-consistency (to reduce the risk of single-path reasoning errors).

2.2 Multi-Agent Architectures in LLM Research

Multi-agent approaches assume that several model instances collaborate or compete within structured roles to improve reasoning quality and the credibility of outcomes. Common patterns include: (i) debate prompting (agents advance competing hypotheses and a judge selects the stronger one), (ii) role specialization (e.g., specialist-moderator-critic), and (iii) interaction-orchestrating frameworks in which agents "converse" and invoke tools.

A historical foundation is the AI safety via debate concept, where two agents argue before a human judge to provide scalable quality control wherever direct human evaluation is difficult (Irving, Christiano, and Amodei, 2018). More recent work on multi-agent debate shows that multi-round agent debates can raise accuracy on mathematical and strategic tasks relative to single models, and can also help non-experts (including weaker models acting as judges) better identify correct answers in reading-comprehension tasks (Du *et al.,* 2023; Khan *et al.,* 2024).

Specialist, moderator, and critic roles. In practice, debate is often combined with an explicit division of labor. In CAMEL, the "expert" and "task-specifier" roles are stabilized via inception prompting, enabling more coherent multi-stage problem solving (Li *et al.,* 2023). Tooling frameworks such as AutoGen allow developers to program specialist agents (e.g., data analysis, source retrieval) and to add a judge/evaluator for dispute arbitration or result validation (Wu *et al.,* 2023).

Critic/reviewer functions are further strengthened by self-reflection and self-refine techniques, in which an agent critiques and iteratively improves its own outputs without additional supervised training (Shinn *et al.,* 2023; Madaan *et al.,* 2023). Separately, the LLM-as-a-Judge line of work indicates that strong models can serve as reliable judges/reviewers for other models-albeit with biases and limits that must be mitigated (Zheng *et al.,* 2023).

Surveys of multi-agent debate frameworks point to a rapidly evolving ecosystem (frameworks, benchmarks, taxonomies) but also highlight failure modes: susceptibility to rhetorical persuasion, superficial agreement (sycophancy), and performance degradation under symmetry (when judge and debaters are of similar strength) (Smit, 2023; Chern *et al.,* 2024).

Accordingly, recommended practices include: clear orchestration rules (number of

rounds, stopping criteria), separation of roles (specialist-critic-moderator/judge), factual validation (e.g., via RAG), and auditable protocols for selecting the winning hypothesis. This configuration leverages the benefits of debate (stronger justifications, resilience to single-path hallucinations) while curbing quality degradation risks.

2.3 Applications of LLMs in Finance

The most frequently discussed application areas include investment advisory, credit analysis/risk assessment, and report generation with information extraction from documents. Recent surveys synthesize the capabilities and limitations of financial deployments (Lee et al., 2024; Nie et al., 2024), while the development of domain-specific models (e.g., BloombergGPT) and open initiatives (FinGPT) supports sector-specific tasks (Wu et al., 2023; Yang, Liu, and Wang, 2023).

lsIn investment advisory, LLMs can assist in preparing recommendations (summaries, scenarios, justifications), although quality depends on prompt design, source grounding (RAG), and risk controls. The models are generally viewed as complementary tools to human advisers (Lee *et al.,* 2024; Nie *et al.,* 2024).

In credit analysis and risk assessment, early studies indicate that appropriately tuned LLMs can rival classical methods on selected scoring tasks, provided there is strict validation, bias control, and assurance of explainability (Feng *et al.,* 2023; Nie *et al.,* 2024).

For report generation and document-centric workflows, domain-specific models report improvements in classification, company-centric question answering, and document summarization, particularly when combined with RAG and source attribution (Wu *et al.,* 2023; Lee *et al.,* 2024).

Key risks include hallucinations and factual errors (Ji *et al.,* 2023), as well as arithmetic and quantitative-reasoning mistakes-prompting numerical validation and integration with computational tools (Lee *et al.,* 2024; Nie *et al.,* 2024). In the EU, advisory use cases and the processing of personal data require compliance with MiFID II (ESMA suitability guidance) and GDPR; in addition, the EU AI Act strengthens obligations concerning risk management, transparency, and oversight of AI systems (ESMA, 2023).

2.4 Research Gaps

Despite extensive evidence that chain-of-thought and self-consistency improve performance on logical and arithmetic tasks (Wei *et al.,* 2022; Wang *et al.,* 2022), there is a lack of studies that isolate the effect of prompt structure (e.g., role assignment, CoT, number of exemplars, validation) on portfolio-decision metrics (e.g., Sharpe/Sortino, drawdown, turnover) under realistic market conditions.

Recent finance surveys acknowledge the promise of LLMs but point specifically to a shortage of standardized domain evaluations and experimental protocols in finance (Lee *et al.,* 2024; Nie *et al.,* 2024).

There is no publicly established, peer-reviewed "auditable prompt library" for financial tasks-i.e., a set of templates with metadata (purpose, constraints, sources), regression tests, compliance mapping (MiFID II/ESMA, GDPR), and built-in quality controls. Existing guardrails frameworks (e.g., NVIDIA NeMo Guardrails) are general rather than sector-specific; at the same time, EU regulations (ESMA suitability guidance, the EU AI Act) raise requirements for transparency, audit trail, and risk management, strengthening the case for such a library (ESMA, 2023; EU AI Act, 2024; NVIDIA, 2024).

3. Methodology

The aim of the proposed methodology is to design prompt engineering and a multi-agent collaboration protocol (Strategist-Critic-Moderator) that enhances the quality, reproducibility, and auditability of LLM inferences across a broad range of financial tasks (credit assessment, risk/ALM, compliance, investment screening). It rests on four pillars: (A) enforced multi-path reasoning (chain-of-thought with self-consistency), (B) grounding outputs in a curated knowledge corpus (retrieval-augmented generation, RAG), (C) role dialogue within multi-agent architectures (role-playing, inception prompting), and (D) verification of claims and the safe use of computational tools.

Evidence indicates that self-consistency markedly amplifies the effectiveness of CoT on reasoning-intensive tasks; RAG reduces hallucinations and improves factuality; and frameworks such as AutoGen and CAMEL enable programmable collaboration among roles.

The proposed methodology assumes that the quality of a decision system derives chiefly from the design of prompts and the rules of interaction among agents, rather than from a single "monolithic" model instance. Accordingly, we require: (i) explicit reasoning (concise CoT steps) with multi-path sampling and consensus selection (self-consistency); (ii) full knowledge provenance (each claim indicates its source and date from the RAG corpus); (iii) quantification of uncertainty together with a catalog of risks; (iv) output formats that enable sanity checks (numerical consistency, quantile monotonicity) and automated auditing; and (v) guardrails-refusal to respond when data are insufficient or when policies would be violated.

3.1 Agent Architecture

The system proposes three roles:
Strategist - generates 2-3 alternative decision options (e.g., grant/deny credit,

maintain/raise the liquidity limit, include/exclude an instrument), each packaged as: evidence → rule/policy → figures → uncertainty → risks. The Strategist draws exclusively on the RAG context and-where permitted-on tools (calculator, queries).

Critic - conducts fault-finding: tests consistency with the context and policies, verifies numerical outputs, assesses uncertainty calibration, and augments the risk register. The Critic follows a chain-of-verification (CoVe) plan, i.e., targeted control questions and independently derived answers.

Moderator - consolidates a solution on a "consensus-by-evidence" basis, discards elements with unresolved "high-severity" critiques, performs sanity checks, and records the decision log.

This role dialogue accords with observations from AutoGen (configurable agent interactions) and CAMEL (role-playing, inception prompting), which supports stability and improves output quality.

Interaction rounds. T0: the Strategist produces alternatives; T1: the Critic classifies issues (A evidence/provenance, B policies/limits, C consistency with context, D uncertainty/calibration, E risks) by severity (high/medium/low) and proposes a minimal fix; T2: the Strategist corrects high/medium issues; T3: the Moderator merges and audits (sanity checks, decision log).

3.2 Prompt Library

What follows is a set of concrete, ready-to-use prompts. They are designed as modular "building blocks" for diverse financial decisions and require injection of CONTEXT (RAG). For models that support sampling, we recommend self-consistency (3-5 runs with selection of the most concordant variant).

In practice, if a model can return different answers to the same question (i.e., generation involves some randomness-commonly controlled by a parameter such as temperature), self-consistency means running the same prompt multiple times (e.g., 3-5), collecting the outputs, and selecting the conclusion that appears most frequently or commands the broadest agreement across runs. The intuition is that a correct answer is more likely to recur consistently, whereas errors are more idiosyncratic-hence the aggregate result is, on average, more reliable.

3.3 Prompt for the Strategist - Generator of Decision Alternatives

[ROLE] You are a Strategist for financial decision-making.
[GOAL] Propose 2-3 alternative decisions for the task: <problem description>.
[CONTEXT/DATA] Use ONLY the materials in the CONTEXT section (RAG).
Cite every claim (ID, date).
[RULES]

- Conduct explicit reasoning (concise CoT steps). If you use a tool (calculator/BI), show input/output.
- For each alternative return: (i) evidence and the applicable rule/policy (with citation),
  (ii) key figures/calculation results relevant to the decision,
  (iii) uncertainty (P5/P50/P95 or an equivalent interval),
  (iv) a list of risks and side effects.

[FORMAT] JSON:

```
{
 "alternatives": [
   {
    "name": "A",
    "decision": "...",
    "evidence": [
      {"quote": "...", "source": "ID", "date": "YYYY-MM-DD"}
    ],
    "figures": {"metric_1": ..., "metric_2": ...},
    "uncertainty": {"P5": ..., "P50": ..., "P95": ...},
    "risks": ["...", "..."]
   },
   { ... }
 ]
}
```

[CONTROL] Do not go beyond the CONTEXT; if data are missing - provide a JUSTIFIED REFUSAL with a list of gaps.
[CONTEXT] <<inject summaries and/or document excerpts from RAG here>>

3.4 Critic Prompt - Fault-Finding + Verification

[ROLE] You are a Critic of compliance, risk, and correctness.
[GOAL] Evaluate the Strategist's alternatives and identify gaps/defects together with minimal fixes.
[RULES]

- Classify issues by category: (A) evidence/provenance, (B) policies/limits, (C) consistency with CONTEXT, (D) uncertainty/calibration, (E) risks/side effects.
- Assign each issue a **severity**: high/medium/low; propose a minimal fix.
- Apply a chain-of-verification (CoVe) plan: list control questions and answer them independently, citing sources from the CONTEXT.

[FORMAT] JSON:

```
{
 "critique": [
   {
    "alternative": "A",
    "issues": [
```

```
      {
       "type": "B",
       "severity": "high",
       "evidence": "ID:...",
       "fix": "..."
      }
    ]
   }
 ],
 "verification_questions": [
   {
    "question": "...",
    "answer": "...",
    "source": "ID"
   }
 ]
}
```

[CONTROL] Do not introduce new data; rely exclusively on the CONTEXT.

3.5 Moderator Prompt - Evidence-Based Consensus + Audit

[ROLE] You are the Moderator of the decision-making process.
[GOAL] Select or consolidate the final solution based on evidence and prepare audit-ready materials.
[RULES]

- Reject elements with **high-severity** critiques that the Strategist has not addressed.
- Prefer alternatives with better uncertainty calibration and more complete risk coverage.
- Use the scheme: **evidence → rule/policy → conclusion**; always provide citations (ID, dates).
- Perform sanity checks on numbers and consistency (e.g., P5 ≤ P50 ≤ P95; ranges, totals-where applicable).

[FORMAT] Markdown + JSON block:
Final recommendation
<justification with citations> ## Uncertainty and risks P5 = ..., P50 = ..., P95 = ...; Risks: [...]
Decision log ```json [{"ts": "ISO8601", "agent": "...", "action": "...", "details": "..."}] ```

Evaluation is conducted ex ante (without market verification):

- Factuality/provenance - share of claims with a correct citation (ID, date) from RAG.
- Decision completeness - presence of all five sections (evidence,

rule/policy, figures, uncertainty, risks).

- Uncertainty calibration - satisfaction of P5 ≤ P50 ≤ P95 and a brief justification of the spread.
- Policy compliance - number of violations and justified refusals.
- Arithmetic consistency - passing sanity checks (ranges, totals-where applicable).
- Agent stability - variance of recommendations across repetitions (with self-consistency vs. without).
- Evaluation effectiveness - agreement between the Moderator and an independent "LLM-as-a-judge" (optional).

The choice of metrics (1)-(2) follows from the logic of RAG and enforced provenance; (6) reflects the effect of self-consistency; and (7) draws on findings regarding an independent LLM judge.

Implementation requires: (i) a whitelist of sources for RAG (policy repositories, supervisory reports, macro data) with versioning and timestamps; (ii) multi-agent orchestration (conversation graph, rounds, stopping criteria, CoVe question queue) in line with the AutoGen documentation; (iii) secure invocation of tools (calculator/BI) with an input/output ledger; and (iv) a complete decision log and automated format tests. These practices are recommended in publications on RAG and multi-agent conversational frameworks.

4. Research Results and Discussion

Because this study is conceptual in nature, the results below should be read as hypothetical effects of implementing the methodology described in Section 3 within real decision-making processes at financial institutions. These conclusions draw on consolidated findings in the literature concerning (i) prompt-induced reasoning techniques (chain-of-thought, CoT), (ii) the self-consistency decoding strategy, (iii) source-grounded answering (RAG), and (iv) multi-agent architectures (role dialogue, interaction orchestration). Each of these pillars has empirical support in general LLM research and-subject to their limitations-grounds the expected practical implications for the financial sector.

4.1 Hypothetical Quality Effects

(A) Greater accuracy and stability of reasoning via CoT + self-consistency. Using explicit reasoning steps (CoT) together with repeated sampling and selection of the most concordant conclusion (self-consistency) should reduce response variance and the incidence of logical-arithmetic errors in finance tasks that require numerical justification (e.g., risk thresholds, policy rules, inferences from due-diligence analyses). The literature reports substantial gains in reasoning performance when CoT and self-consistency are employed (e.g., on GSM8K, SVAMP, AQuA), which justifies expecting improvements in domain decisions, if prompts enforce a

structured return of content (sections: evidence → rule → figures → uncertainty → risks).

(B) Reduced hallucinations and improved provenance via RAG. Using retrieval-augmented generation-i.e., tightly constraining the model to a curated corpus (policies, supervisory guidance, reports, data summaries) and requiring citations of sources and dates-should lower hallucinations, increase answer specificity, and facilitate auditability (traceability). RAG studies demonstrate higher factuality and quality on knowledge-intensive tasks; in financial settings this translates into document-based justifications and faster compliance reviews.

(C) Better detection of errors and "blind spots" through a multi-agent architecture. Separating roles into a Strategist (generator of alternatives), a Critic (fault-finding, CoVe), and a Moderator (consensus, sanity checks) should raise the detection rate of evidentiary gaps, policy violations, and numerical inconsistencies, while limiting the single-path bias typical of a solitary agent. Frameworks such as AutoGen document the benefits of programmable agent interactions, suggesting that in finance the Critic can serve as an internal reviewer and the Moderator as an auditor of outcomes.

(D) Increased auditability and regulatory readiness. Enforced output formats (JSON/Markdown with citations, dates, decision, uncertainty, and a decision log) streamline internal review, regression testing, and compliance inspection. In practice this means shorter time to assemble audit materials and greater transparency of the reasoning trail (who/what/on what basis). Mechanical verification (sanity checks, quantile monotonicity) provides an additional safeguard against numerical errors. (This effect is indirectly supported by findings on CoT/self-consistency and RAG, as well as reported orchestration and control capabilities in AutoGen.)

4.2 Practical Implications

Below we explain what the previously described methodology can offer in real financial decision-making processes. These are assumptions to be confirmed in actual deployments.

(A) Stronger "first drafts" of decisions. When the model "thinks aloud" (chain-of-thought) and is run multiple times, after which we select the conclusion with the greatest agreement (self-consistency), the answers tend to be more stable and, on average, more sensible. This is particularly helpful for credit applications, liquidity-limit settings, or concise due-diligence summaries. Studies indicate that such a procedure improves reasoning quality across many tasks.

(B) Fewer fabricated statements by working on in-house documents. If the model is required to use only the provided materials (policies, reports, data)-that is, RAG-

and, in addition, must cite sources and dates, the risk of hallucinations (plausible-sounding but false claims) declines. In practice, it also becomes easier to trace where a given conclusion came from.

(C) Fewer oversights thanks to the "three-role" workflow. Dividing work among a Strategist (proposes options), a Critic (hunts for errors and gaps), and a Moderator (consolidates and checks consistency) function much like an effective team process: one party produces, another reviews, a third organizes. As a result, inconsistencies, missing documents, or broken rules surface more quickly. Multi-agent frameworks (e.g., AutoGen) show that such agent collaboration is practically feasible.

(D) Easier auditing and lower regulatory risk. Standardized response formats (brief narrative plus structured data: citations, dates, figures, risk list, and a decision log) make it faster to verify what a decision rests on and whether it complies with internal rules. This matters for both internal and external reviews. Additionally, a "trust-but-verify" procedure (Chain-of-Verification) can be enabled: the model first drafts an answer, then formulates control questions and answers them independently reducing factual errors.

Where this can help-three typical scenarios:

- Credit analysis: The Strategist produces 2-3 variants (e.g., "approve," "conditional," "deny") with references to policies and data; the Critic checks compliance and gaps; the Moderator selects the final version and records the decision trail. Expected effect: faster processing with better documentation.
- Risk/ALM and treasury: Requiring uncertainty intervals (e.g., P5/P50/P95) and automatic numerical consistency checks helps "calibrate" the prudence of recommendations (e.g., when adjusting liquidity limits).
- Compliance and reporting: Working on an internal corpus (RAG) and requiring citations shortens preparation for audits and reduces the risk of erroneous guidance.

These benefits are conditional-reasonable expectations that require testing within a given institution (on its data, procedures, and quality metrics). It is also worth noting that the quality of such solutions is increasingly assessed with separate models acting as "judges," which accelerates evaluation but has its own limitations and calls for caution.

4.3 Discussion of Limitations

Despite promising foundations, interpretive caution is warranted. First, evidence for CoT and self-consistency comes primarily from general benchmarks (mathematical, logical), and transferring these results to the financial domain requires dedicated tests and task-appropriate metrics (e.g., stability of

recommendations under context shifts, compliance with policies).

Second, RAG does not eliminate risk: the quality of justifications depends directly on the quality and timeliness of the corpus and on retriever choice; moreover, in environments with frequent regulatory change, maintaining the repository can be costly.

Third, multi-agent architectures incur computational overhead (more turns, longer outputs) and demand precise orchestration; excessive debate can paradoxically prolong the process without meaningful quality gains if the rules are not tightly specified. (In practice, 2-3 turns with minimal, pre-defined stopping criteria are advisable.)

5. Intermediate Conclusions and Hypotheses for Verification

Considering the analysis, we formulate three key operational hypotheses for empirical testing in future work:

H1 (reasoning): prompts with CoT + self-consistency improve the accuracy and stability of responses on decision tasks relative to instruction prompts without CoT. (Rationale: systematic gains on reasoning benchmarks.)

H2 (provenance): incorporating RAG with mandatory citation (ID, date) reduces hallucinations and increases the audit acceptability of responses. (Rationale: RAG results on knowledge-intensive tasks.)

H3 (multi-agent): the Strategist-Critic-Moderator triad, with sanity-check gates and a CoVe plan on the Critic's side, increases error detection and the coherence of final recommendations compared with a single agent. (Rationale: benefits reported by multi-agent frameworks).

5.1 Implications for Practice

If these hypotheses are borne out, financial organizations can expect the following benefits when operationalizing decision processes:

- Higher quality "first drafts" of decisions: more accurate and complete alternatives at the outset (effect of the Strategist + CoT/self-consistency).
- Fewer ex post corrections thanks to early detection of formal gaps and inconsistencies (effect of the Critic + CoVe + sanity checks).
- Shorter audit cycles and stronger compliance: decisions accompanied by citations, dates, and a decision log (effect of RAG + Moderator).

We emphasize that these conclusions are conditional-they require empirical validation in realistic settings (institutional data, compliance procedures, version-

controlled RAG corpora). Only such tests will allow assessment of the magnitude of measurable benefits (time, cost, quality) and tuning of orchestration parameters (number of turns, stopping thresholds, refusal policies).

6. Conclusions, Proposals, Recommendations

In this study we present a conceptual, multi-agent architecture for supporting financial decision-making, in which output quality is explicitly a function of prompt design and of the interaction rules among the Strategist, Critic, and Moderator roles.

The proposal integrates four strands of contemporary LLM research: prompt-induced explicit reasoning (chain-of-thought), its stabilization via the self-consistency strategy, grounding in a curated corpus (retrieval-augmented generation, RAG), and cooperation among multiple agents in a structured role dialogue.

Prior literature shows that CoT and self-consistency systematically improve performance on reasoning-intensive tasks, which motivates their use as mechanisms for reducing logical and numerical errors in financial processes. At the same time, RAG provides provenance and recency frameworks that limit hallucinations and enable auditing of the reasoning trail, while multi-agent frameworks allow roles and rules of debate to be programmed, increasing the detection of a single model's "blind spots."

The proposed methodology is theoretical: it describes anticipated mechanisms for quality improvement and their implications for institutional practice, but it does not constitute empirical proof of effectiveness in real-world settings. In particular, the following require verification: the magnitude of quality and cost gains from applying self-consistency in operational tasks; the durability of RAG benefits under changes to the corpus and data-ingestion procedures; and the optimal number of rounds and stopping rules in multi-agent debate given response-time and compute constraints.

An additional area for testing concerns robustness to errors external to the model (e.g., incomplete or outdated RAG sources) and the impact of base-model choice on result stability and uncertainty calibration.

It is worth underscoring the alignment of the proposed architecture with rising requirements for oversight and transparency: enforced output formats, strict source provenance, a decision log, and sanity-check gates provide a natural anchor for internal control and audit processes and may facilitate implementation of principles stemming from European regulatory frameworks for AI systems. While legal compliance itself demands separate analysis, the mechanisms advocated here-especially the explicit presentation of evidentiary bases and the reproducibility of

the reasoning path-are conducive to solutions that accord with the spirit of regulation.

In sum, the paper offers a coherent scheme for combining prompt-engineering techniques with multi-agent orchestration as a route to more reliable, auditable, and operationally useful decision support in finance.

It is, however, a point of departure: the next step must be an empirical research program-from A/B tests of prompt and role configurations, through stability and calibration evaluations on real tasks, to cost-effectiveness studies in production environments.

Only such validation will make it possible to specify precisely when and in what form the proposed architecture outperforms alternatives, and how it should be tailored to the specifics of decision processes within financial institutions.

References:

Chern, S., Chern, E., Neubig, G., Liu, P. 2024. Can large language models be trusted for evaluation? Scalable meta-evaluation of LLMs as evaluators via agent debate. Available at: https://arxiv.org/abs/2401.16788.

Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I. 2023. Improving factuality and reasoning in language models through multi-agent debate. Available at: https://arxiv.org/abs/2305.14325.

ESMA. 2023. Guidelines on certain aspects of the MiFID II suitability requirements (ESMA35-43-3172). Available at: https://www.esma.europa.eu/sites/default/files/2023-04/ESMA35-43-3172_Guidelines_on_certain_aspects_of_the_MiFID_II_suitability_requirements.pdf.

European Parliament and Council. 2024. Regulation (EU) 2024/1689 (Artificial Intelligence Act). Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ%3AL_202401689.

Feng, D., Dai, Y., Huang, J., Zhang, Y., Xie, Q., Han, W., Chen, Z., Lopez-Lira, A., Wang, H. 2023. Empowering many, biasing a few: Generalist credit scoring through large language models. Available at: https://arxiv.org/abs/2310.00566.

Guardrails, A.I. 2025. Guardrails (framework and validators for reliable LLM apps). Available at: https://guardrailsai.com/docs/.

Irving, G., Christiano, P., Amodei, D. 2018. AI safety via debate. Available at: https://arxiv.org/abs/1805.00899.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Chan, H.S., Madotto, A., Fung, P. 2023. Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), Article 248.

Khan, A., Hughes, J., Valentine, D., Ruis, L., Sachan, K., Radhakrishnan, A., Grefenstette, E., Bowman, S.R., Rocktäschel, T., Perez, E. 2024. Debating with more persuasive LLMs leads to more truthful answers. Available at: https://arxiv.org/abs/2402.06782.

Lee, J., Stevens, N., Han, S.C., Song, M. 2024. A survey of large language models in finance (FinLLMs). Available at: https://arxiv.org/abs/2402.02315,

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M.,

Yih, W.T., Rocktäschel, T., Riedel, S., Kiela, D. 2020. Retrieval-Augmented Generation for knowledge-intensive NLP tasks. NeurIPS 2020.

Li, G., Hammoud, H.A.A.K., Itani, H., Khizbullin, D., Ghanem, B. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. Available at: https://arxiv.org/abs/2303.17760.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in NLP. ACM Computing Surveys.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y. et al. 2023. Self-Refine: Iterative refinement with self-feedback. Available at: https://arxiv.org/abs/2303.17651.

Nie, Y., Kong, Y., Dong, X., Mulvey, J.M., Poor, H.V., Wen, Q., Zohren, S. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. Available at: https://arxiv.org/abs/2406.11903.

NVIDIA. 2024. NeMo Guardrails - Documentation. Available at: https://docs.nvidia.com/nemo-guardrails/index.html.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. Available at: https://arxiv.org/abs/2303.11366.

Smit, A. 2023. Should we be going MAD? A look at multi-agent debate for problem solving. Available at: https://arxiv.org/abs/2311.17371.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D. 2023. Self-consistency improves chain-of-thought reasoning in language models. ICLR 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D. 2022. Chain-of-Thought prompting elicits reasoning in large language models. NeurIPS 2022.

Wu, Q., et al. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. Available at: https://arxiv.org/abs/2308.08155.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., Mann, G. 2023. BloombergGPT: A large language model for finance. Available at: https://arxiv.org/abs/2303.17564.

Yang, H., Liu, X.Y., Wang, C.D. 2023. FinGPT: Open-source financial large language models. Available at: https://arxiv.org/abs/2306.06031.

Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I., Zhang, H. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. Available at: https://arxiv.org/abs/2306.05685.