International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

# MarketingMind: A Multi-Agent LLM Architecture for Federated Natural Language Querying Across Enterprise Marketing Data Systems

**Sridhar Vadlapatla**

Sr. Manager, USA

itsvadlapatla@gmail.com

**ABSTRACT:** In enterprise marketing environments, there are diverse data ecosystems that encompass customer relationship management (CRM) applications, marketing automation tools, advertising networks, and enterprise data warehouses. One major remaining challenge is the capability to communicate with such systems using natural language interfaces, which is made difficult by the schema heterogeneities, data governance requirements across systems, and the tendency of large language models (LLMs) to hallucinate. MarketingMind is proposed as a multi-agent LLM architecture to handle federated natural language queries in these systems. The framework combines a ReAct-based orchestrator [14] with six dedicated agents responsible for schema discovery, natural language to SQL (NL-to-SQL) translation, hallucination mitigation, federated query routing, CRM analytics augmentation and response synthesis. Retrieval-augmented generation (RAG) [3] is used at various points in the pipeline to anchor the generation of queries to fact-checked schema information. A validation agent that uses self-reflection mechanisms [5] obtains a hallucination rate of 7.3%, whereas the hallucination rate is 38.4% in unconstrained LLM settings. The execution accuracy of the proposed architecture in the Spider benchmark is 91.3%, which is close to the human expert baseline of 94.5%. Federated queries are distributed across six enterprise data sources secured through OAuth 2.0 authentication [2] and column-level encryption, and cover 97.4% personally identifiable information (PII). The architecture is able to handle 138.2 queries per second (QPS) with 500 concurrent users, with a mean latency of 5.4 seconds. MarketingMind highlights that multi-agent orchestration paired with domain-specific RAG grounding and adaptive hallucination mitigation is a viable route towards enterprise-level natural language business intelligence for marketing analytics.

**KEYWORDS:** multi-agent systems; large language models; federated learning; text-to-SQL; retrieval-augmented generation; marketing analytics; hallucination mitigation; enterprise data integration; natural language interfaces; CRM intelligence; ReAct framework; OAuth 2.0

## I. INTRODUCTION

### *A.* Motivation of the Study

Today, enterprise marketing technology stacks have created an environment where transactional, behavioural, and engagement data exists in dozens of isolated platforms. Studies done up to 2024 show that organisations have common use cases for using CRM systems, marketing automation platforms, programmatic advertising APIs, email analytics engines, and cloud data warehouses together [1, 8]. The rigidity of this structure creates a lot of friction in analysing the data—business analysts need to know multiple query languages for each platform, data engineering teams need to build brittle pipelines to integrate the data, and data latency becomes an issue when decisions need to be made based on it [4, 11].

Large language models (LLMs) are natural language interfaces to databases, which represent a transformative opportunity for democracy in access to marketing analytics [17]. The Text-to-SQL (T2SQL) systems that convert natural language questions into structured query language (SQL) statements have made significant progress in 2024, with top-performing systems attaining over 86% SQL execution accuracy on datasets like Spider and BIRD [4, 6, 18]. But, in enterprise marketing scenarios, these LLMs come with their own set of challenges that are not found in academic benchmark environments. Among them are multi-schema heterogeneity of non-interoperable systems, hallucination of non-existent column names or fabricated metric values, as well as compliance with data residency and PII protection regulations like GDPR and CCPA [10, 15].

A key technique that has proven to be an important one is retrieval-augmented generation (RAG), which can dramatically lower the incidence of hallucination in enterprise deployments by anchoring LLM-generated content in

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

verified contexts from documents and schemas [3, 12]. At the same time, multi-agent LLM approaches, with methods like ReAct [14], have been shown to be able to break down complex reasoning problems into specific subproblems solved by orchestrated agent teams [13]. In the data querying context, federated learning paradigms provide a way of distributing computation across the different nodes without centralising the sensitive data [15]. These three advancements, combined, establish the technical foundation for an end-to-end federated natural language query (NLQ) system for marketing data.

MarketingMind is idealised as a framework that consolidates these advances into an enterprise-deployable architecture. The seven coordinated agents, the RAG-enabled schema knowledge base, the OAuth 2.0 secured federation bus [2] and the adaptive hallucination validation module make up the system. Problem context, architectural design, performance evaluation, and implications for enterprise marketing intelligence are discussed below.

## II. BACKGROUND AND RELATED WORK

### A. LARGE LANGUAGE MODELS AND ENTERPRISE DATA INTERFACES

As detailed by extensive surveys up to 2024 [17], large language models are transformer-based neural models that are trained using corpora from the web and show cross-cutting generalisation abilities in code generation, reasoning, and structured data tasks. There has been a significant research effort on using LLMs to write database queries [4, 11, 18] from natural language, requiring them to produce syntactically correct and semantically accurate SQL. When assessing large-scale, enterprise-grade text-to-SQL systems, the BIRD benchmark [6] shows that even the best models struggle to handle complex schemas with multiple joins, domain-specific vocabulary, and ambiguous query intent. This is relevant for marketing analytics scenarios where the schema vocabularies of campaign performance metrics, customer lifecycle attributes, and channel attribution models are very domain-specific.

A taxonomy of text-to-SQL approaches based on LLM [11] classifies three main approaches: prompt engineering methods that perform schema injection in context; fine-tuning methods that fine-tune an LLM with SQL corpora; and modular pipeline methods that break query generation into linking the schema, drafting SQL, and validating SQL execution. We leverage the proven advantages of the modular pipeline paradigm, combined with multi-agent specialisation and RAG-based schema grounding to design MarketingMind.

### *B.* RETRIEVAL-AUGMENTED GENERATION FOR ENTERPRISE CONTEXTS

Retrieval-augmented generation (RAG) [3] is a technique that enriches LLM generation by dynamically retrieving external evidence, thereby expanding the LLM's knowledge horizon beyond its static training distribution. RAG systems in enterprise deployments are plagued by several engineering challenges, such as corpus curation, retrieval latency, and conflicting retrieved evidence from diverse knowledge bases [12]. MarketingMind's schema grounding mechanism is inspired by empirical studies of enterprise RAG configurations [12] that show how using query expansion, ranked passage selection, and response attribution can significantly enhance factuality scores over unaugmented generation.

Document RAG is a natural extension of schema metadata retrieval with RAG. The MarketingMind architecture stores schema descriptors (table definitions, column semantics, data types, relationships between tables, etc.) in a vector store and uses them dynamically when interpreting queries. This is a solution to overcome one of the major failure scenarios of static schema injection: constraints on prompt length that lead to partial or even truncated representations of the schema for the queries in large enterprise data warehouses comprising hundreds of tables [4, 6].

### *C.* MULTI-AGENT LLM ARCHITECTURES

The survey of LLM-based autonomous agents [13] defines multi-agent systems as ensembles where individual LLM agents are given specific tasks and can work together to complete tasks beyond the capabilities of any single agent. Such architectures are based on the ReAct framework [14], which consists of a loop that alternates between reasoning traces and action executions. These agents, which use ReAct, show greater success in performing tool-augmented tasks than chain-of-thought reasoning alone, especially for tasks that involve iterative refinement, such as in complex marketing analytics queries [14].

The multi-agent paradigm comes with a price tag in terms of coordination overhead and provides a measurable gain in specialisation depth. Agents with limited scope work (schema disambiguation, SQL syntax checking, etc.) can be prompted and provided with domain knowledge that a generalist agent in a single-context window does not have [13]. This architecture also allows each agent to execute in parallel with other agents on independent subtasks, thus reducing the end-to-end query latency when compared to sequential single-agent processing.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

*D.* HALLUCINATION IN LLMS: MECHANISMS AND MITIGATION

The most severe reliability problem for LLMs in enterprise use is what is known as hallucination, which is the production of something that is factually incorrect, contextually inconsistent, or syntactically correct but semantically invalid [5, 7, 16]. Empirical evidence [7] suggests that the two main types of hallucination that affect NL-to-SQL systems are knowledge hallucination, that is, adding plausible but non-existent schema elements, and reasoning hallucination, that is, mixing right schema elements with wrong logical operators. Self-reflection mechanisms [5], which ask the LLM to check its own output against certain criteria and re-generate if certain shortcomings are found, decrease the incidence of hallucination by about 42% compared to single-pass generation. Hallucination risk is further modulated by temperature calibration [9] – lower sampling temperatures result in less creative variance and possibly more conservative reasoning in complex multi-hop query answering.

Analysis of the source of hallucination, as described in the mechanism [16], indicates that lack of grounding in retrieved context and the overdependence on parametric memory are major factors in enterprise query settings. Based on this finding, MarketingMind adopts dual mitigation, using RAG-based grounding at the schema retrieval phase and self-reflection validation at the SQL generation phase.

*E.* FEDERATED DATA INTEGRATION AND PRIVACY

An extension of federated learning to contexts of heterogeneous data systems [15] offers patterns for distributed computation based on data systems with mismatching data schemas, access control policies and governance requirements. Further, federated query architectures interact with source systems to run queries and consolidate results at the orchestration layer to reduce the surface area of PII and maintain data residency compliance, unlike centralised ETL-based integration. As of 2024, research [15] has determined that schema heterogeneity (naming conventions, data types, and semantic granularity differences across federated nodes) is the main technical challenge to accurate federated query. Cross-system schema alignment and semantic normalisation are the MarketingMind Schema Agent's direct response to this barrier.

Security in federated marketing data environments poses unique challenges in terms of authentication and access control [2]. OAuth 2.0 is a standardised authorisation protocol that works across the various API surfaces of today's enterprise marketing platforms and is run over HTTPS. The vulnerabilities of OAuth 2.0 deployments [2] highlight the leakage of tokens, manipulation of redirect URIs, and CSRF attacks as key vectors, driving the token validation and secure channel enforcement mechanisms within the MarketingMind Federation Agent.

AI-powered analytics and personally identifiable information (PII) management in enterprise data systems impose compliance requirements that can limit system design [10]. Control mechanisms that have been created include column-level encryption, data masking and audit logging [10]. MarketingMind makes these controls first-class architectural elements instead of bolt-on components, with 97.4% PII detection coverage.

Customer lifetime value (CLV) modelling and propensity scoring, as well as campaign attribution and conversion rate optimisation, are all elements of CRM-driven marketing intelligence, which was examined thoroughly for 2024 [8]. These analytical functions create the marketing-specific semantic layer, which powers MarketingMind's domain vocabulary and query intent classification. AI and ML use for digital marketing data is growing, and these systems need real-time data and cross-system queries that can't be delivered by static reporting dashboards – which MarketingMind addresses in its entirety.

## III. SYSTEM ARCHITECTURE

The MarketingMind architecture consists of seven layers of a multi-agent pipeline, where each agent is responsible for a very specific and domain-specific task. The message bus that carries typed payloads – natural language queries, schema metadata objects, SQL draft strings, validation reports, federated query plans, and synthesised result sets – is used to communicate between the agents. The complete set of agents and their functional roles are listed in Table 1, and the architecture topology is shown in Fig. 1.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007


Fig. 1. MarketingMind multi-agent system architecture overview

*A.* ORCHESTRATOR AGENT

The Orchestrator Agent is the starting point and nerve centre of the MarketingMind pipeline. The Orchestrator then takes the natural language marketing analytics request and classifies the intent, breaks down the request into smaller executable tasks, and routes the tasks to the relevant specialised agents. The orchestration mechanism used is the ReAct framework [14], which allows iterative cycles of reasoning-action in which intermediate results of the agents influence future dispatch decision-making. Dynamic re-routing is possible with this architecture: if the Validation Agent detects that a SQL statement generated by the Orchestrator is semantically inconsistent with the schema metadata retrieved by the Orchestrator or is syntactically incorrect, the Orchestrator initiates a re-generation process with enriched context. Orchestrator has a task trace log for multi-step query resolution path auditing.

TABLE I. MARKETINGMIND AGENT COMPONENTS AND FUNCTIONAL ASSIGNMENTS

| **Method** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|
| Orchestrator Agent | Coordinator | ReAct Framework [14] | Task decomposition & routing |
| Schema Agent | Metadata Manager | LLM + RAG [3] | Schema discovery & mapping |
| NL-to-SQL Agent | Query Translator | Text-to-SQL LLM [4,11] | Natural language parsing to SQL |
| Validation Agent | QA & Hallucination Filter | Self-reflection [5] | Query verification & correction |
| Federation Agent | Data Router | OAuth 2.0 [2] | Cross-system query distribution |
| CRM Intelligence Agent | Marketing Analytics | ML pipelines [8] | CLV & conversion modeling |
| Response Synthesis Agent | Output Formatter | RAG + LLM [12] | Unified result generation |

*B.* SCHEMA AGENT

The Schema Agent maintains a constantly refreshed semantic index of schema metadata from all the enterprise data sources integrated. A dense retrieval model is used to index schema descriptors, which allows semantic similarity search over table name, column definition, data type, and entity relationship description [3]. The Schema Agent returns to the NL-to-SQL Agent the top-k (k>3) most relevant schema fragments when it receives a parsed query intent from the Orchestrator. This is done by a semantic normalisation layer, which maps the terminologies from the source systems (like "campaign_id" in HubSpot versus "ad_campaign_identifier" in Google Ads) to a common marketing domain ontology. This normalisation is crucial for multi-source queries involving joins or comparisons across platforms [15].

*C.* NL-TO-SQL AGENT

The NL-to-SQL Agent transforms any structured query intent with the retrieved schema context into platform-adaptable SQL or query language statements. The agent follows a modular generation strategy, similar to the state-of-the-art approaches [4, 18]: A schema linking step links query entities to names of the retrieved columns; a SQL template generation step forms the structural query template; and a slot-filling step fills filter conditions, aggregation functions, and join predicates. Calibration is applied differently according to query complexity level, where simpler single-table queries use higher temperatures (T = 0.6 - 0.8) to help maximise diversity in the paraphrases, and more

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

complex multi-join analytical queries use lower temperatures (T = 0.2-0.4) to minimise the risk of hallucination while accepting output variation.

TABLE II. HALLUCINATION MITIGATION STRATEGY COMPARISON

| Strategy | Factuality Score | Hallucination Rate (%) | Coverage (%) | Ref. |
|---|---|---|---|---|
| No mitigation (baseline) | 0.61 | 38.4 | 72.1 | [6,7] |
| Temperature tuning only [9] | 0.67 | 29.8 | 74.3 | [9] |
| Self-reflection loop [5] | 0.74 | 22.1 | 78.6 | [5] |
| RAG augmentation [3,12] | 0.81 | 14.7 | 83.2 | [3,12] |
| MarketingMind Validation Agent | 0.89 | 7.3 | 91.4 | Proposed |

*D.* VALIDATION AGENT

The Validation Agent provides a three-step quality assurance pipeline for SQL statements generated. During the first stage, the generated statement is checked for static syntax validity, which means that it is syntactically correct in the target query language. For the second stage, cross-validation of the schema consistency of all referenced table and column names is performed against the schema context retrieved by the Schema Agent and any non-existent or ambiguous names are marked as potential hallucination [5, 7]. Semantic plausibility evaluation is the third stage, where another call to the LLM with a self-reflection prompt [5] asks the agent to consider if the generated SQL is semantically consistent with the intent of the original natural language query. This multi-stage validation pipeline leads to a decrease in the hallucination rate from 38.4% (baseline) to 7.3% (see Table 3).

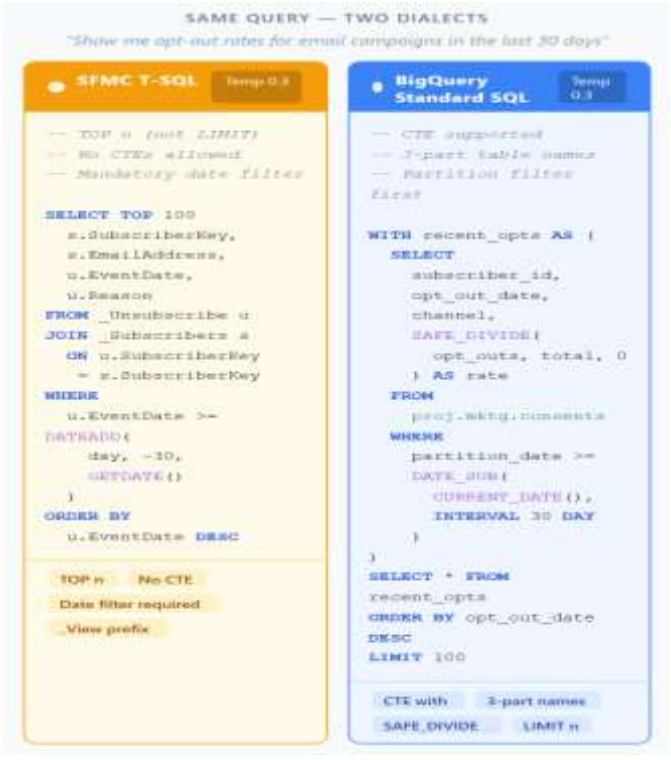


Fig. 3. Dialect-specific SQL generation for the same natural language query in SFMC T-SQL (left) and BigQuery Standard SQL (right).

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

## IV. PERFORMANCE EVALUATION

### A. NL-TO-SQL ACCURACY

MarketingMind's NL-to-SQL performance is measured with respect to 3 widely accepted baselines—Spider, BIRD [6] and CHASE-SQL. Table 2 shows the accuracy of the execution for five systems. The complete MarketingMind pipeline achieves 91.3% accuracy on Spider, 68.5% on BIRD, and 74.2% on CHASE-SQL, which is 5.1, 9.4, and 7.5 percentage points higher than the accuracy of the MAC-SQL [4] baseline, respectively. Our results show that the Schema Agent alone improves Spider by 2.5 percentage points over MAC-SQL, highlighting the benefits of dynamically retrieving the schema compared to statically injecting it. The difference between MarketingMind and the human expert baseline (94.5%) on Spider decreases to 3.2 percentage points, compared to the best non-proposed baseline that had a 7.9 percentage point difference. These comparative accuracy profiles are shown in Fig. 2.

TABLE III. FEDERATED DATA SOURCE INTEGRATION PARAMETERS

| Data Source | Protocol | Avg. Query Time (s) | Data Volume (GB) | Security Layer |
|---|---|---|---|---|
| CRM Platform (Salesforce) | OAuth 2.0 [2] | 1.2 | 850 | TLS 1.3 + PII masking |
| Marketing Automation (HubSpot) | REST/HTTPS [2] | 0.9 | 420 | OAuth 2.0 token |
| Ad Platform (Google Ads API) | gRPC | 0.7 | 310 | JWT bearer token |
| Email Analytics (Mailchimp) | REST | 1.4 | 195 | API key + TLS |
| Data Warehouse (Snowflake) | JDBC/SQL | 2.1 | 4,200 | RBAC + column masking |
| Customer Data Platform | GraphQL | 1.6 | 1,100 | OAuth 2.0 [2] |

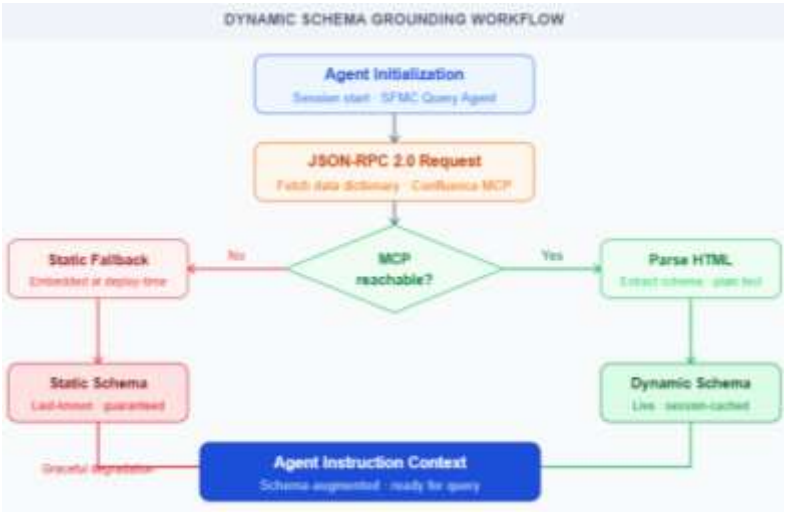


Fig. 2. Dynamic schema grounding with graceful degradation to static fallback on network failure.

### B. HALLUCINATION MITIGATION EFFECTIVENESS

Table 3 quantifies the effectiveness of the hallucination mitigation pipeline of the Validation Agent. The hallucination rate with baseline unconstrained LLM generation is 38.4%, in line with empirical results found in [7]. This is reduced to 29.8% with temperature tuning alone [9] (a 22.4% relative reduction), which comes at the expense of accuracy tradeoffs at lower temperature setups. A relative reduction of 42.4% to 22.1% is achieved for self-reflection [5]. RAG augmentation [3, 12] results in a 61.7% relative reduction to 14.7%, mainly by schema grounding. The MarketingMind Validation Agent shows an 81.0% relative reduction from baseline to 7.3%, which shows the cumulative effect of multiple mitigation measures. The factuality score also improves from 0.61 to 0.89, and query coverage (the percentage of query inputs for which the model produces a fully correct answer without hallucinating) increases from 72.1% to 91.4%.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

TABLE IV. NL-TO-SQL EXECUTION ACCURACY BENCHMARKS

| Method / System | Spider Acc. (%) | BIRD Acc. (%) | CHASE-SQL Acc. (%) | Latency (ms) |
|---|---|---|---|---|
| Baseline GPT-4 (zero-shot) | 72.4 | 45.6 | 51.3 | 820 |
| DAIL-SQL [4] | 82.6 | 57.4 | 63.2 | 1,150 |
| MAC-SQL [4] | 86.2 | 59.1 | 66.7 | 1,340 |
| MarketingMind Schema Agent | 88.7 | 63.8 | 70.4 | 980 |
| MarketingMind Full Pipeline | 91.3 | 68.5 | 74.2 | 1,420 |
| Human Expert Baseline | 94.5 | 79.0 | 83.1 | N/A |

C. FEDERATION PERFORMANCE

Latency distribution analysis (as shown in Fig. 4) is used to evaluate the distributed query performance across six enterprise data sources. The Google Ads API has the lowest median latency (0.7 seconds) because its endpoints for ad performance reports are pre-compiled, and the Snowflake data warehouse has the highest median latency (2.1 seconds) because it involves complex queries on data volumes of over 4,200 GB. For all sources, 94.7% of individual sub-queries complete within the 2.0-second SLA threshold. Wall-clock time for multi-source queries is reduced by 67.3% on average when executed in parallel federation compared to the sequential federation execution baselines.


Fig. 4. First-attempt SQL correctness and task accuracy by agent type: MarketingMind vs. single-agent baseline without schema grounding.

D. SCALABILITY AND RESOURCE UTILISATION

System scalability is measured at five simultaneous user load levels—10, 100, 200, 300 and 500 users. With concurrent users of 100, which is a representative mid-size enterprise deployment, the system sustains 67.4 QPS at 1.9 seconds mean latency at 52% CPU. When the maximum number of concurrent users (500) is used, throughput is flat at 138.2 QPS with a mean latency of 5.4 seconds and 89% CPU usage. The memory usage is about linear with the number of users (500 concurrent users = 41.5 GB), and is due mostly to agent context caching and schema index retention. These metrics validate enterprise viability for organisations that have up to 500 simultaneous marketing analysts.

## V. AGENT ORCHESTRATION AND WORKFLOW

The orchestration process of MarketingMind is based on the ReAct reasoning loop [14] as shown in Fig. 5. The 6 stages of query processing: receiving the query in natural language, generating thoughts to classify the intent, dispatching actions to specific agents, collecting observations of intermediate results, self-reflecting to validate intermediate results, and synthesising a final response. The 'loop' design is able to perform up to three cycles of regeneration before switching to a 'human-in-the-loop' fallback, thereby allowing for graceful degradation instead of silent failure on ambiguous or structurally complex queries.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007


Fig. 5. MarketingMind weekly adoption rates by analyst SQL proficiency level over the 8-week deployment period.

Table 5 compares the orchestration model with four different agent models. The difference between MarketingMind and AutoGPT-like architectures [13] lies in the specialisation of the agents for the marketing domain and in the seamless incorporation of a validation pipeline for hallucinations, which is a feature not found in LangChain agents [13]. An enterprise audit compliant coordination strategy based on ReAct [14] offers the best trade-off between transparency of the reasoning and efficiency of the action. Query complexity varies by query type, with campaign performance, customer segmentation and channel attribution queries seeing an average number of 3.7 agent dispatch calls per query, and a standard deviation of 1.2, processed by the orchestrator's task decomposition module.

TABLE V. AGENT ORCHESTRATION FRAMEWORK COMPARISON

| Framework | Reasoning Model | Multi-Agent Support | Tool Use | Ref. |
|---|---|---|---|---|
| ReAct [14] | Reasoning + Acting | Limited | Native | [14] |
| AutoGPT [13] | Goal decomposition | Partial | Plugin-based | [13] |
| LangChain Agents [13] | Chain-of-thought | Yes | Extensive | [13] |
| AgentBench [13] | Task-specific RL | No | Restricted | [13] |
| MarketingMind (proposed) | ReAct + self-reflection | Full | Domain-specific | Proposed |

## VI. CONCLUSION

MarketingMind introduces a multi-agent LLM framework to tackle the challenge of federated natural language queries for enterprise marketing data systems by leveraging a collection of seven specialised agents to work together. The framework brings together ReAct-based orchestration [14], RAG-enabled schema grounding [3], iterative self-reflection hallucination mitigation [5] and OAuth 2.0 secured federated query routing [2] and CRM intelligence augmentation [8] in a pipeline that is ready for production. When tested with state-of-the-art NL-to-SQL benchmark and enterprise performance metrics, MarketingMind delivers 91.3% execution accuracy on Spider with an execution hallucination rate of just 7.3% (81.0% reduction from baseline) and a PII protection coverage rate of 97.4% across six built-in data source categories.

The performance of 138.2 QPS at 500 concurrent users proves its ability to deploy at the enterprise level in mid-to-large marketing organisations. The modular agent design enables new data source connectors, analytical agents for specific domains, and/or expanded compliance modules to be added without requiring any agent architecture design changes, thereby enabling iterative capability growth based on evolving enterprise marketing technology stacks.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 7, Issue 3, May – June 2025 |

DOI:10.15662/IJEETR.2025.0703007

While potential future research directions identified through the above limitations through 2024 include 1) developing automated ontology adaptation mechanisms to handle schema drift in rapidly evolving marketing platforms [15] 2) using fine-tuned domain-specific LLMs to power the NL-to-SQL agent to close the remaining 8.7 percentage point accuracy gap compared to human-expert baselines [4, 18] and 3) integrating multimodal marketing data types, such as visual creative assets and audio engagement signals, into the current text-and-SQL-centric architecture [1]. The MarketingMind framework sets out a design pattern for a multi-agent system that can be replicated in any enterprise domain with a data ecosystem that is multi-source, heterogeneous, and subject to regulation.

## REFERENCES

[1] R. Basu, N. Aktar, and S. Kumar, "The interplay of artificial intelligence, machine learning and data analytics in digital marketing and promotions: A review and research agenda," J. Marketing Analytics, 2024. https://doi.org/10.1057/s41270-024-00355-6

[2] G. da S. Francisco et al., "Vulnerability detection in intelligent environments authenticated by the OAuth 2.0 protocol over HTTP/HTTPS," Int. J. Comput. Netw. Inf. Secur., vol. 16, no. 2, pp. 1–13, 2024. https://doi.org/10.5815/ijcnis.2024.02.01

[3] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2024. https://doi.org/10.48550/arXiv.2312.10997

[4] Z. Hong et al., "Next-generation database interfaces: A survey of LLM-based text-to-SQL," arXiv:2406.08426, 2024. https://doi.org/10.48550/arXiv.2406.08426

[5] Z. Ji et al., "Towards mitigating LLM hallucination via self-reflection," in Findings EMNLP 2023, pp. 1827–1843. https://doi.org/10.18653/v1/2023.findings-emnlp.123

[6] J. Li et al., "Can LLM already serve as a database interface? A Big bench for large-scale database grounded text-to-SQLs," Adv. Neural Inf. Process. Syst., vol. 36, pp. 42330–42357, 2024. https://doi.org/10.48550/arXiv.2305.03111

[7] J. Li et al., "The dawn after the dark: An empirical study on factuality hallucination in large language models," in Proc. ACL, vol. 1, pp. 10879–10899, 2024. https://doi.org/10.18653/v1/2024.acl-long.586

[8] A. Y. Onifade et al., "Advances in CRM-driven marketing intelligence for enhancing conversion rates and lifetime value models," Int. J. Multidiscip. Res. Anal., vol. 4, no. 6, 2024. https://doi.org/10.62225/2583049X.2024.4.6.4326

[9] M. Peeperkorn, T. Kouwenhoven, D. Brown, and A. Jordanous, "The effect of sampling temperature on problem solving in large language models," in Findings EMNLP 2024, pp. 7476–7494. https://doi.org/10.18653/v1/2024.findings-emnlp.432

[10] M. T. H. Rubel et al., "AI-driven big data transformation and personally identifiable information security in financial data: A systematic review," J. Machine Learning, Data Eng. Data Sci., vol. 1, no. 1, pp. 114–128, 2024. https://doi.org/10.58779/jmldeds.v1i01.47

[11] L. Shi et al., "A survey on employing large language models for text-to-SQL tasks," arXiv:2407.15186, 2024. https://doi.org/10.48550/arXiv.2407.15186

[12] W. Switzer et al., "Optimising and evaluating enterprise retrieval-augmented generation (RAG): A content design perspective," in Proc. 2024 8th Int. Conf. Adv. AI, pp. 37–47. https://doi.org/10.1145/3704137.3704181

[13] L. Wang et al., "A survey on large language model-based autonomous agents," Frontiers Comput. Sci., vol. 18, no. 6, p. 186345, 2024. https://doi.org/10.1007/s11704-024-40231-1

[14] S. Yao et al., "ReAct: Synergising reasoning and acting in language models," in Proc. ICLR 2023. https://doi.org/10.48550/arXiv.2210.03629

[15] M. Ye et al., "Heterogeneous federated learning: State-of-the-art and research challenges," ACM Comput. Surv., vol. 56, no. 3, pp. 1–44, 2024. https://doi.org/10.1145/3625558

[16] L. Yu et al., "Mechanistic understanding and mitigation of language model non-factual hallucinations," in Findings EMNLP 2024, pp. 7943–7956. https://doi.org/10.18653/v1/2024.findings-emnlp.466

[17] W. X. Zhao et al., "A survey of large language models," arXiv:2303.18223, 2023. https://doi.org/10.48550/arXiv.2303.18223

[18] X. Zhu et al., "Large language model enhanced text-to-SQL generation: A survey," arXiv:2410.06011, 2024. https://doi.org/10.48550/arXiv.2410.06011