Journal on
Artificial Intelligence

Tech Science Press

https://doi.org/10.32604/jai.2026.078487

ARTICLE

LLM-Enabled Multi-Agent Systems: Empirical Evaluation and Insights into Emerging Design Patterns & Paradigms

Harri Renney[1,*], **Maxim Nethercott**[1], **Nathan Renney**[2] **and Peter Hayes**[1]

[1]Kaze Technologies, Kaze Consulting, Bath, UK
[2]Computer Science Research Centre, University of the West of England, Bristol, UK
*Corresponding Author: Harri Renney. Email: harri@kaze-consulting.com

Received: 31 December 2025; Accepted: 04 March 2026; Published: 17 April 2026

ABSTRACT: This paper provides systemisation on the emerging design patterns and paradigms for Large Language Model (LLM)-enabled multi-agent systems (MAS), evaluating their practical utility across various domains, bridging academic research and industry practice. We define key architectural components, including agent orchestration, communication mechanisms, and control-flow strategies, and demonstrate how these enable rapid development of modular, domain-adaptive solutions. Three real-world case studies are tested in controlled, containerised pilots in telecommunications security, national heritage asset management, and utilities customer service automation. Initial empirical results show that, for these case studies, prototypes were delivered within two weeks and pilot-ready solutions within one month, suggesting reduced development overhead compared to conventional approaches and improved user accessibility. However, findings also reinforce limitations documented in the literature, including variability in LLM behaviour that leads to challenges in transitioning from prototype to production maturity. We conclude by outlining critical research directions for improving reliability, scalability, and governance in MAS architectures and the further work needed to mature MAS design patterns to mitigate the inherent challenges.

KEYWORDS: Multi-agent systems (MAS); agent coordination; human-agent interaction; human-in-the-loop; large language models (LLMs); automation; Single Information Environment (SIE)

1 Introduction

The concept of distributing problem solving across multiple AI programs is a field in contemporary artificial intelligence (AI) that emerged in the late 1970s with works from Lesser and Corkill [1] and Hewitt [2]. Research explored distributed problem solving, in which multiple AI type programs could cooperate to solve complex tasks that were too large for a single monolithic system to solve at the time. Originating from these concepts, the modern terminology around multi-agent systems (MAS) has been formalised [3,4]. With the recent emergence of a combination of key enabling technologies, including the transformer [5], large language models (LLMs) [6], and access to considerable compute power [7], previously conceptual multi-agent designs and MAS are now possible to investigate in practice using LLMs [8].

Since Google published "Attention is All you Need" [5], modern language models have been converging on the use of *transformer* deep learning architecture with significant generative results [9]. When model parameters are scaled up, the word by word predictive mechanism leads to impressive emergent abilities. These include solving certain arithmetic calculations [10], creative writing [11] and software code generation [12]. Scaled up versions of language models like GPT [13] and PaLM [14] are referred to as LLMs, and

particular LLMs have been shown to pass professional exams such as the US medical licensing exam [15] and the US law practise BAR examination [16], along with convincingly going undetected in writing [17,18] and code generation [19].

The prominence of LLMs in contemporary AI research and industry is unmistakable, with leading multinational corporations investing heavily in their development. Individual LLM builds are estimated to cost between $1 million and $100 million [20], and since 2023, at least 86 major releases have been recorded (Fig. 1). This trend reflects an unprecedented acceleration in model innovation and capital expenditure, reinforcing the imperative for scalable and systematic design paradigms to support sustainable deployment.

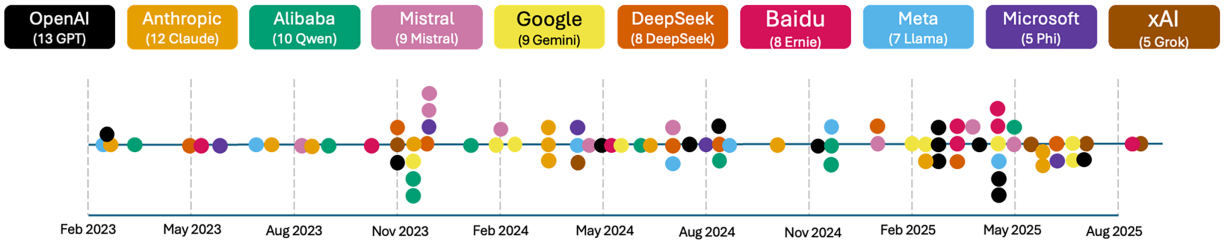

Figure 1: Timeline illustrating the cumulative release of major LLM models by ten leading AI companies since 2023 [21].

Using modern LLMs as the core reasoning engine, specialist AI agents can be formed [22]. With a well-designed prompt, access to up-to-date domain data, and the necessary tools, an agent can be tailored to a specific task or field. The agent then becomes highly specialised for a target task, improving its effectiveness whilst reducing execution time and compute costs [23–25].

By arranging a number of specialist agents in a network that can hand off the next part of the process to another specialised agent, a MAS solution can be formed. Their efficiency stems from the division of labour inherent in MAS, whereby a complex task is divided into multiple smaller tasks, each of which is assigned to a distinct agent [26]. It is this flexibility that as we propose in this paper, makes MAS suited to solve problems in a variety of sectors including utilities and telecoms.

1.1 Why MAS Design Patterns Matter Now

Across sectors, organisations report persistent difficulties turning large, heterogeneous data-assets into operational value [27–31]. Three recurring challenges motivate the need for standardised, reusable MAS *design patterns* include: (1) Disparate data and silos [32–34]; (2) Unstructured information at scale [35–37]; and (3) Domain-specific constraints [38–41]. LLM-enabled MAS directly target these issues by coordinating specialist agents over structured and unstructured sources [8,42,43]. However, in practice, a lack of detailed design pattern choices and ad-hoc orchestration choices (e.g., control flow, history-sharing, tool arbitration) lead to long development cycles and brittle cross-domain adaptation.

We explicitly call out the current disconnect between industry converging on pattern-like agent orchestration practices and academic works that remain split between narrow domain-specific MAS and visionary large-scale ecosystems. This motivates the present work, to systematise emergent MAS design patterns and make their applicability conditions and trade-offs explicit, using evidence from real deployments. Building on these findings, we identify the critical areas of advancement required to further mature and stabilise this emerging paradigm on the path toward production-ready technology.

1.2 Paper Structure

We investigate this space and present a more systematic characterisation of MAS, along with initial results from real-world implementations. These results provide insight into the utility and limitations of LLM-based MAS approaches. The remainder of this paper is organised as follows:

- Section 2 reviews the latest academic, industry and open-source literature on LLM-enabled MAS, concluding with how this paper aims to provide further systemisation of design patterns and test this approach on real-world case-studies.
- Section 3 identifies the motivations for using LLM-based MAS solutions, that address key data challenges being faced by organisations.
- Section 4 presents our additional systemisation of the MAS design patterns, outlining how configured specialist agents and network arrangements according to repeatable design patterns can be applied to solve problem classes such as data retrieval and semi-automated, human-in-the-loop pipelines.
- Section 5 documents the application of this approach across three real-world case studies, capturing the experiences and observations of developers, end-users, and senior business executives.
- Section 6 provides a discussion of the findings, highlighting both strengths and limitations of the proposed design patterns and broader implications for MAS development.
- Finally, Section 7 concludes the paper with a summary of key insights and future research directions in this evolving field.

2 Literature Review

The rapid advances in LLM architectures and the unpredictable emergent behaviours indicate that the technology is still in a formative stage [44]. Exasperated by the lack of understanding of how LLMs achieve their emergent abilities [45] and ethical concerns leading to potential government regulation [46], effective application of LLMs for intelligent agent networks is still ongoing and timely.

One foundational technique that enables LLMs to exhibit improved reasoning capabilities is Chain-of-Thought (CoT) prompting [47]. This method guides the model through a series of intermediate reasoning steps before producing a final answer, improving accuracy on tasks requiring logical inference. CoT has proven particularly effective for complex problem-solving and offers an interpretable view of the model's reasoning process, revealing how conclusions are reached and where errors may arise. While CoT operates sequentially, an extension known as Tree-of-Thought (ToT) introduces branching reasoning paths, yielding measurable improvements on tasks with high complexity [48]. Notably, ToT reasoning invokes multiple agents to explore alternative reasoning trajectories, representing a rudimentary form of dynamic multi-agent coordination, a concept that this paper expands upon in greater detail.

Beyond prompt engineering and multi-agent invocation, tool integration has emerged as a critical technique for specialising general-purpose LLMs into task-specific intelligent agents. A prominent example is Retrieval-Augmented Generation (RAG) [49], introduced by Lewis et al. in 2020, which enables LLMs to access and reason over external datasets not included in their training corpus. By indexing dynamic, domain-specific, or proprietary data sources, RAG allows models to generate informed and contextually grounded outputs, making it an industry-standard option for knowledge-intensive applications [50]. As one of RAG's pioneering researchers notes [51]:

"Language models are about 20% of the entire system/solution, and is surrounded by further engineering parts."

emphasising the need to design systems rather than relying on building isolated models. Similarly, other specialised agents require interfaces to interact with diverse tools, such as SQL databases, continuing the evolving drive towards MAS, where multiple LLM-powered specialised agents collaborate to solve complex, heterogeneous tasks.

2.1 Academic Literature

As specialised agents proliferate, research increasingly explores how they can be connected and coordinated within MAS to address complex, targetted, domain specific problems. Recent literature presents several MAS designs targeting focused applications. For instance, Onobhayedo et al. [52] investigate MAS for requirements capture in software development, demonstrating how LLM-powered agents can assist in eliciting and validating specifications. Similarly, Lin et al. [53] propose a MAS framework for code generation, where agents act as domain experts across different stages of the development lifecycle, interacting with engineers in a human–machine teaming environment. Their findings indicate promising efficacy in adopting engineering roles and facilitating collaborative workflows. Beyond software engineering, Kearney's *Bots of the SoCs* study [54] evaluates MAS integration within Security Operations Centres (SoCs), benchmarking different agent arrangements using the Boss of the SOC dataset [55]. Results suggest that the most effective MAS configuration tested achieves performance comparable to a junior cyber analyst, highlighting MAS potential to augment human expertise in high-stakes domains.

Despite the promising results in the literature, MAS implementations face notable challenges. Pan et al. [56] highlight that as agent networks scale, coordination complexity can introduce significant overhead, reducing efficiency and responsiveness. This can lead to misalignment of the MAS solutions caused by agents local optimisation conflicting with global goals, the stochastic nature of using LLM's and error propagation where an agent's incorrect reasoning can cascade through a system. Hence, to be effective, the MAS approach needs structured design patterns and paradigms to mitigate intrinsic challenges from MAS including coordination failures and improve reliability. Further, matters of sustainability and ethics will need to be considered as MAS architectures build on LLMs, addressing the emerging challenges inherent to LLM-based systems [57–59]. Where, in some cases, a prototype may imply the opportunity to defer the importance of issues such as verified accountability, biases, and new vectors of exploitation, it is important that these be considered and addressed [60]. It is therefore important to remain attentive to developments in this area as a better understanding emerges, particularly in areas such as censorship, transparency, and intellectual property and plagiarism, which are currently less explored ethical considerations.

Beyond research focused on domain-specific MAS designs, more ambitious horizon initiatives are emerging. These efforts explore how AI agents might cooperate at scale, potentially evolving into ubiquitous tools that replace existing technologies, including aspects of the internet. A notable example is MIT's NANDA project [61,62], which proposes a decentralised *Internet of AI Agents* framework designed to support collaboration among billions of specialised agents across a distributed architecture. Positioned between narrowly focused MAS implementations and visionary concepts of global agent autonomy, this paper focuses on the further systemisation of emerging design patterns that enable practical MAS solutions for diverse domain-specific problems.

2.2 Open-Source & Industry Initiatives

Drawing from the activity outside of the academic literature, the open-source community is pushing ahead with frameworks and architectures for supporting the development of LLM-enabled MAS. The LangChain framework has supplied considerable support for multi-agent LLM programming [63]. This includes the LangChain Expression language [64], a domain-specific language that is parsed by the LangChain framework to arrange "chains" of agents to feed inputs and outputs between each other, mostly with support for sequential arrangements. The LangChain expression language is extended by the LangGraph project [65] to manage and arrange branching networks of multiple agents for MAS development.

Major industry vendors are increasingly aligning with community-driven frameworks, actively integrating multi-agent approaches into their products. Notable examples include Microsoft's AutoGen [66] and Copilot [67], as well as Amazon's Bedrock Agents [68]. As organisation and open-source projects expand, the challenge of interoperability becomes more pronounced. In response, Anthropic introduced[1] the Model Context Protocol (MCP), an open standard designed to enable LLMs to connect and interact seamlessly with external tools, data sources, and services [69].

Most leading cloud compute suppliers and Agentic AI vendors are rushing to develop similar tooling and support within their own proprietary environments, either based off or heavily influenced by the open-standards and initiatives. At the time of writing, Microsoft has it's Agentic AI development tooling in beta phase in Azure AI Foundry [70], and OpenAI offers it's custom '*GPT*'s for users to specialise agents that they can then integrate with wider protocols and environments [71].

2.3 Root Causes of Academic-Industry Disconnect

Recognising the academic-industry disconnect, we propose three interacting causes behind this observation:

1. **Granularity mismatch.** Academic studies often evaluate narrow, domain-specific MAS in controlled settings, optimising for benchmark clarity; industry prioritises tool-and platform-centric delivery, optimising for integration, governance, and cost.
2. **Evaluation incentives.** Research emphasises correctness and model comparisons; practitioners emphasise development velocity, cross-domain adaptability, and operability (logging, audit, safety rails).
3. **Reproducibility vs. variability.** Industry settings face distributional shifts, schema drift [72], and changing tool availability, favouring dynamic orchestration; academic evaluations often fix distributions for comparability, underexposing wider generalisability.

This paper aims to bridge the gap between industry and academia through sysmetaisation of design approachs between the granularity mismatch, showcase demonstrable case-studies and point towards the academic reigour needed to evaluate real-world implementations and wrap it all up in a reproducible format for future researchers.

2.4 Design-Patterns and Scenario Applicability

Existing MAS design patterns operate across key MAS design axes such as coordination topology [66], control flow, interaction style [73] and history sharing. Seperately, scenario-oriented applicability is addressed in literature discussing disparate siloed datasets with enterprise integration [27,28], unstructured document QA at scale [8,43], cross-domain analytics and exploratory investigations with OSINT fusion [74–76].

[1] And since handed over to the Agentic AI Foundation, directly funded by the Linux Foundation

Based on our review, the literature currently lacks well-established work that combines MAS design axes with scenario applicability and demonstrates insights into adaptation conditions, benefits, and bottlenecks without over-claiming performance. Hence, our work will systemise the MAS design axes in Section 4, demonstrate scenario applicability with a case-series in Section 5 and draw insights across both of these in our discussions in Section 6.

2.5 Quantitative Gaps and Under-Reported Evaluation Dimensions

Although prior work demonstrates the feasibility of LLM-enabled MAS, it largely lacks quantitative reporting at the design-pattern level. In particular, existing research provides little data on the reusability rate of MAS design patterns across domains, the frequency and nature of failures during cross-domain adaptation, or the latency and cost attributable to orchestration logic. These omissions make architectural choices difficult to compare and hinder cumulative progress. While our case studies are exploratory and do not yield statistically robust estimates of these indicators, they report structured proxy measures (e.g., development effort, throughput, unit cost, and observed failure modes) that help expose design trade-offs and motivate the need for systematic, pattern-aware quantitative evaluation in future MAS research.

2.6 Research Gap

Our review reveals a disconnect between two dominant strands of academic literature: (i) highly specialised LLM-enabled MAS designs evaluated in narrow contexts, and (ii) visionary frameworks proposing global, decentralised agent ecosystems. In contrast, industry efforts are rapidly converging on practical MAS development, driven by advanced tooling and open standards that enable organisations to address domain-specific challenges at scale whilst mitigating the inherent challenges that arise from MAS.

This divergence highlights the need for clearer academic systemisation of emerging design patterns that underpin these industry-driven solutions. While such patterns are increasingly adopted in practice, they remain under-explored in academic discourse. In this paper, we address this gap by documenting a common MAS design paradigm and evaluating its utility through empirical tests across three case studies. Our findings provide insights into the strengths and limitations of this approach and identify complexities, such as alignment and scalability, that traditional architectures often avoid.

Considering this review, our contribution is pattern-level systematisation with stated applicability and limits (Section 4), followed by a deployment-oriented case series (Section 5) that further evidences value in the cross-domain approach, and where limitations begin to emerge. These insights inform the future need for quantitative indicators as proposed in Section 2.5.

3 Motivating the Use of LLM Enabled MAS Design Patterns in Organisations

Organisations face a multitude of challenges in managing their data-assets effectively [27,28]. In particular, many have accumulated vast quantities of data [29], yet struggle to organise, govern and interpret it in a way that generates meaningful insights for decision-making and supports automation [30,31], with as little as 16% being classed as data-driven according to Lewkowicz [77]. In this section we examine three core data challenges confronting organisations that MAS is positioned to address: Disparate Datasets, Unstructured Datasets, and Domain Specific Data Context.

3.1 Disparate Data

As organisations grow in scale and complexity, they often encounter significant challenges associated with data fragmentation and siloed information systems [32]. For large organisations in sectors such as Finance [33] and Government [34], data is often generated and stored across multiple departments, locations, and platforms, each of which may employ different data standards, formats, and schemas. Over time, this unmanaged decentralisation leads to inconsistencies and incompatibilities that make data integration, governance, and cross-functional analysis increasingly difficult. Traditional solutions, such as static data pipelines or schema-mapping tools, often struggle to cope with the dynamic and heterogeneous nature of these distributed data environments, especially when applied retrospectively [78].

LLM-enabled MAS offers a promising alternative to overcoming this challenge. Through the use of specialist AI agents capable of interpreting, aligning, and translating disparate data schemas in real time, MAS can dynamically integrate information from multiple, previously isolated sources. These agents can autonomously identify relationships between datasets, resolve schema mismatches, and harmonise data structures to enable unified access and analysis. This adaptability allows agent-based systems to act as intelligent intermediaries, facilitating improved navigation across diverse, distributed, and evolving data landscapes that would usually consume significant workforce time.

3.2 Unstructured Data

When data is structured and well-organised at its source, it can be readily managed, queried, and analysed to generate reliable insights. For example, sensor outputs typically conform to predefined schemas [79], producing consistently formatted records, while transactional data stored in relational databases adheres to established data models and validation rules [80]. Such structured datasets are well-suited to conventional analytical and automation techniques [81].

However, organisations are increasingly faced with large volumes of unstructured or semi-structured data, estimated to comprise as much as 90% of enterprise information [35]. These datasets lack consistent schemas or formats [36] and include examples such as interview transcripts, emails, reports, and free-form text, all characterised by ambiguity, context dependency, and linguistic variability [37]. These properties make integration and interpretation significantly more challenging, often requiring advanced natural language processing (NLP) and semantic reasoning techniques to extract meaningful structure [82,83].

LLM-based agents offer a powerful capability to address these challenges by performing sophisticated NLP and semantic reasoning on unstructured datasets [42]. When orchestrated within a MAS, these agents can operate seamlessly across both structured and unstructured data sources, creating an integrated Single Information Environment (SIE) through which information can be navigated, queried, and understood in real time [8,43].

3.3 Domain Specific Problems

Across industries, organisations often face similar categories of challenges, yet the specific context, constraints, and requirements within each domain introduce nuances that demand tailored solutions [38]. For instance, systems built for healthcare [39], defence [40], or finance [41] may share a need for secure data retrieval or analysis, but differ substantially in their data semantics, regulatory obligations, and interpretive logic. Developing and maintaining such bespoke systems traditionally requires specialised software engineering expertise and personnel capable of aligning technical solutions with domain-specific needs, creating dependencies that can limit scalability and agility [84].

LLM-enabled MAS provides adaptive AI that when linked to contextual resources and tooling, is quickly configured to meet domain specific needs. By configuring specialist agents with domain-relevant knowledge, ontologies, and reasoning capabilities, a MAS can be rapidly adapted to new contexts without extensive retraining or redevelopment. These agents can interpret domain-specific data [74], apply contextual understanding [75], and generate informed outputs, effectively bridging the gap between generic AI capability and domain-specialised application. As a result, MAS can reduce reliance on human intervention for bespoke problem-solving [76], allowing professionals to focus on higher-level analysis while maintaining a flexible, reusable framework for tackling diverse, domain-specific challenges.

4 An Emerging Paradigm

Building on the key challenges faced by organisations, this section introduces the foundational elements of the emerging MAS paradigm being documented in this paper. We begin with the latest conceptualisation of specialist autonomous agents, commonly referred to as ReAct Agents and progress towards their integration into collaborative MAS networks. These networks form the basis of broader systems designed to address complex, domain-specific problems.

4.1 ReAct Agents

In 2022, the concept of an individual agent operating within a CoT reasoning loop and performing decision-making tasks has been documented in the definition of the ReAct agent [73]. The ReAct agent involves the following core components:

- **Tools:** Functions that perform specific tasks, such as querying the Google Search API, accessing an SQL database, executing code via a Python Interpreter [85], or performing calculations.
- **Reasoning Engine:** The large pre-trained model that powers the system. State-of-the-art LLMs are preferred due to their advanced reasoning capabilities.
- **Agent Orchestration:** The overarching system that manages interactions between the LLM and its tools.
- **Memory:** Mechanisms for tracking past reasoning and inputs to inform current decisions. Short-term memory is typically maintained within the prompt context, while long-term memory is stored externally (e.g., in a vector database) and retrieved using semantic similarity to provide relevant context for ongoing tasks.

Following the workflow loop illustrated in Fig. 2, such an agent typically operates through the following steps:

1. **Query Processing:** The system processes an input query from another entity (e.g., a user or another agent).
2. **Tool Utilization:** The agent selects an appropriate tool (if necessary), prepares its input, executes the tool, and obtains the output.
3. **Information Processing:** The agent interprets the tool's input and output, generates an observation, and determines the next action.
4. **Synthesis & Response:** Using the accumulated information, the agent decides whether it has sufficient context to return a response or whether to repeat the process (returning to Step 1) until a coherent answer is formed, as per CoT reasoning.

For MAS, ReAct agents contribute: (i) local memory for capturing short-term prompt state, with optional long-term vector store; (ii) tool arbitration where decision on internal reasoning or tool invokation is made; and (iii) handoff readiness where the agent decides if tasking responsibility can be given to another

agent with all necessary information. The MAS layer then manages the orchestration and interactions between multiple ReAct agents under a chosen *topology*.

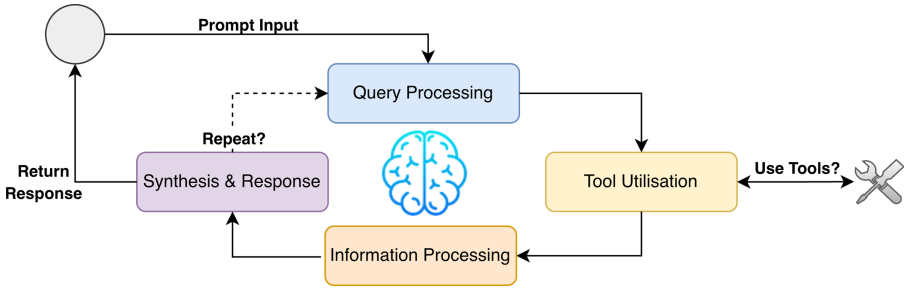

Figure 2: Workflow loop of a ReAct agent, depicting iterative reasoning and action execution of key stages, enabling recursive reasoning through chain-of-thought reasoning.

4.2 Multi-Agent Systems (MAS)

While a single agent can be highly effective when tailored to perform a specialised task [86], increasing its scope by adding too many available tools often introduces complexity and degrades performance. Empirical evidence shows that agent performance often degrades when the toolkit expands beyond 8–12 tools, due to context-window overload and cognitive interference [87,88]. This underpins the rationale for delegating capabilities to specialist agents in a MAS, rather than a single, overburdened general model.

With MAS, capabilities can be distributed across specialist agents that communicate and share reasoning with one another. This approach preserves the effectiveness of individual agents while enabling the system to tackle more complex, multi-faceted problems. The resulting MAS offers several key benefits: **(i) Modularity:** Independent agents simplify development, testing, and maintenance; **(ii) Specialisation:** Expert agents can focus on specific domains, improving accuracy and efficiency; **(iii) Control:** Each agent can be explicitly configured and managed, offering finer grained control to improve alignment with system objectives; **(iv) Economic:** Tasks are allocated only to relevant agents, meaning agents that are not required remain inactive. This modular activation reduces computational overhead and operational costs. Furthermore, agents can be distilled into smaller, more efficient versions optimised for their specialised tasks [89].

Defining communication mechanisms is a critical step in assembling a MAS. Building on the concept of specialist ReAct Agents introduced in Section 4.1, Table 1 covers key components of MAS architectures that will be referred back to in this paper, including control flow strategies, interaction styles, history-sharing policies, and network configurations to coordinate agent behaviour effectively. Fig. 3 illustrates these components within an abstract Single Information Environment (SIE) configuration. In this example, a strategic decision-maker interacts with the system as a human-on-the-loop, requesting reports through a coordinator agent. The coordinator employs a dynamic control flow to determine, at runtime, which specialist data-retrieval agents should be engaged via agent handoff with history sharing. One specialist agent uses an explicit tool flow[2] to fetch data, while another operates under an explicit human-in-the-loop control flow for data acquisition. This diagram syntax will be used throughout the remainder of the paper to convey MAS network configurations concisely.

[2]Under explicit flow, tools are called with a fixed priority. Whilst under dynamic flow, agents can decide on what tools it calls either purely based on its internal reasoning or using a kind of scoring system.

Table 1: Stack of common definitions for key components used to define MAS architectures, augmented with applicability conditions and boundaries/differentiation.

Aspect	Type	Description	Applicability	Limitations
Control Flow [66]	**Static**	Predefined sequence of agent interactions via graph edges.	Governed and stable pipelines with strict requirements (such as auditing).	Susceptible to domain drift as changes require graph edits.
	Dynamic	Routing/branching chosen at run time by an agent's reasoning.	Open-ended queries, heterogeneous tools, ability to skip irrelevant steps to save cost/latency.	Higher decision variance, less predictable and consistency.
Interaction Styles [90]	**Handoff**	Pass a task off for another agent to take responsibility of.	Transition of responsibility, ownership handed to most suitable agent.	Excessive handoffs may accumulate context and cascade errors [91].
	Tool-Flow	Current agent remains focus of control and calls leaf tools.	Low-latency leaf operations with stable APIs and clear I/O contracts.	Risk of single-agent overload as toolset grows or APIs drift.
History Sharing [92]	**Full reasoning trace**	Share labelled logs of actions, communications, and reasoning steps during handoff.	Regulated review, context for downstream agents, detailed debugging.	Increases token/latency cost and may expose sensitive intermediates.
	Final results only	Do not exchange intermediate steps; pass only the artefact/answer.	High-throughput, low-risk tasks with strict latency or privacy constraints.	Reduced interpretability; slower error localisation.
Network Configurations [93]	**Supervisor**	Central coordinator manages leaf agents.	Clear accountability, focused and low sophistication.	Bottlenecks at the hub and potential routing bias.
	Swarm/Distributed	Peers hand off dynamically without any hierarchy.	Exploratory, open-ended tasks and with dynamic tool availability.	Route unpredictability, handoff loops, reproducibility/cost spikes.
	Hierarchical	Multi-tier supervisors for scalability/structure.	Large decompositions, heterogeneous subdomains, clear escalation paths.	Cross-tier coordination overhead, possible error propagation between levels.
	Single Information Environment (SIE) [94]	Agents specialise by dataset/service; coordinator routes to the minimal subset for grounded answers.	High data heterogeneity; per-source governance (e.g., multi-repository retrieval).	Limited purely for data retrieval applications.
Human Interaction [95]	**Human-in-the-loop**	Human intervention for: input/outputs, approve/reject, review tool calls, or interrupt flows.	Risk-sensitive decisions, policy enforcement, model-drift detection.	Reduces throughput; provide clear escalation/rollback paths.

(Continued)

Table 1 (continued)

Aspect	Type	Description	Applicability	Limitations
	Human-on-the-loop	Oversight only (monitor, audit).	Monitoring stable pipelines; lower operational cost.	Requires robust telemetry and alerts; issues may be detected post hoc.

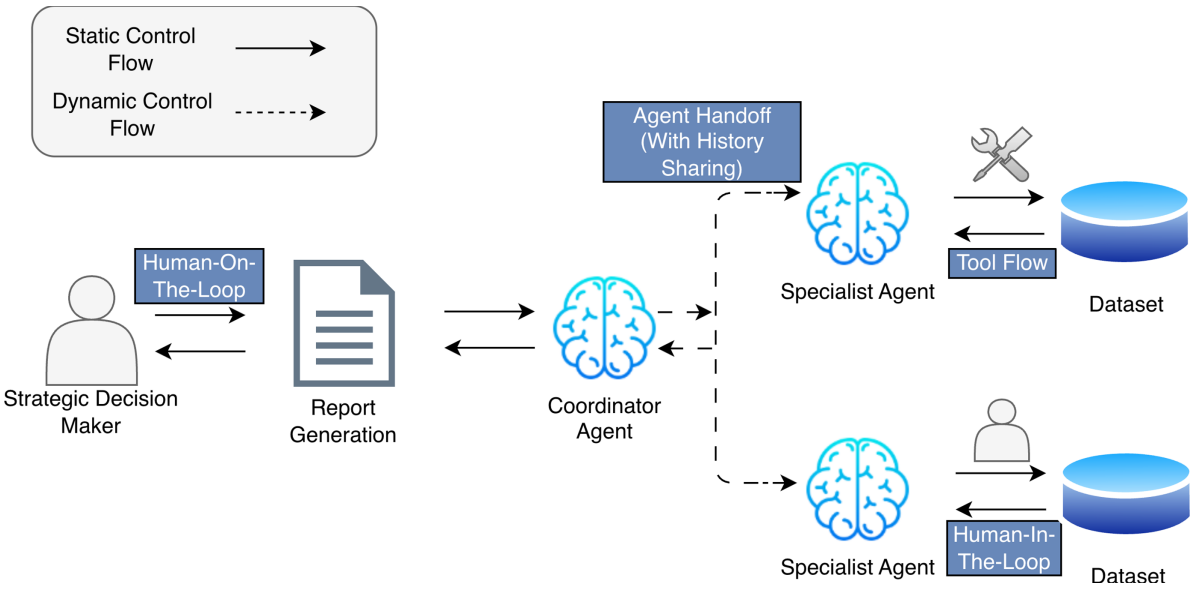

Figure 3: Abstract representation of a multi-agent system configured as a single information environment (SIE). The diagram illustrates dynamic control flow, agent handoff strategies, and human-on-the-loop oversight for coordinated data retrieval and decision-making.

4.3 Specalist Configuration: The Single Information Environment (SIE)

Originating in use in Defence [94], we capture the Single Information Environment (SIE) as part of a a data-centric MAS topology in which agents are specialised by dataset or data service (e.g., a knowledge base), and a coordinator routes each request to the minimal subset needed to return a grounded answer. The coordinator enforces the chosen axis settings (explicit vs. dynamic control flow, handoff vs. tool-flow at the leaves, and selective history sharing for governance), while source-specific logic remains encapsulated within data-specialist agents. This pattern is most appropriate when data heterogeneity and source-level governance dominate; it is less suitable when a single authoritative repository suffices or tight cross-source transactional joins are required. Fig. 3 illustrates an SIE instance with dataset-specialised agents, coordinator-mediated routing, and selective trace sharing.

The SIE pattern is complementary to existing data platforms rather than an alternative. Dataset-specialist agents can be connected to organisational data lakes and data warehouses via adapters and work alongside other specialist agents in a MAS network managing separate structured and unstructured data sources. So where storage, lineage, and batch modelling remain the remit of lakes/warehouses; SIE adds an agentic interface layer.

5 Case-Series

This section applies the emerging paradigm that leverages MAS patterns and LLMs as core reasoning engines for each agent in a series of three proofs-of-concept (PoCs) and pilot studies outlined in Table 2 addressing real-world challenges across multiple industries. These include telecommunications, government defence, national heritage, and utilities, demonstrating the versatility and scalability of MAS in diverse, high-stakes domains. Full details on these can be found in the Appendix Table A1.

Table 2: Summary table of the case studies covered within this study.

ID	Domain	MAS Architecture	Evaluation Metrics
CS1	Telecommunications Security	SIE	Stakeholder Feedback
CS2	National Heritage Sector	SIE	Stakeholder Feedback & Open-Ended review per Query
CS3	Utilities Sector	Hierarchical	UAT, Likert ratings, categorisation labelling

Beyond these examples, the approach is also being extended to intelligent and secure data migration within the Government Defence sector, which will be evaluated as part of future work.

5.1 Evaluation Framework and Indicators

We standardise the case-series evaluation around a small set of design-pattern–aware indicators: (1) development velocity (time from project start to first working prototype/UAT); (2) throughput (items processed per minute) and unit cost (estimated cost per item); (3) quality dimensions rated on a 5-point Likert scale (correctness, usefulness, clarity, groundedness, safety); and (4) operational posture (human-in/on-the-loop).

We define groundedness based on Microsoft's RAG evaluator [96] as how well the generated response aligns with the given context, the grounding source, and doesn't fabricate content outside of it. A concise rubric is: 1 = *unsupported/contradicted*, 3 = *partially supported with gaps*, 5 = *fully supported with precise citations or excerpts and no contradictions*.

Given the exploratory scope and deployment constraints, CS1–CS2 report on only qualitative stakeholder feedback and proxy indicators; CS3 reports throughput and unit cost, plus Likert ratings from a pilot UAT. We explicitly avoid over-claiming statistical significance, instead our results encourage reproducing with robust studies and describe threats to validity.

5.2 CS1—Telecoms Security—Security Operations Centre Agents

In this case study, the security division of a major UK telecommunications provider sought to enhance intelligent tooling and data support within its SOC. The organisation's existing approach to cyber threat intelligence was heavily human-centric, requiring dozens of analysts to manage vast volumes of data [97]. This approach is increasingly unsustainable, unable to keep pace with the exponential growth and complexity of the global threat landscape [98]. For additional context, publicly disclosed vulnerabilities, representing only a fraction of SOC data feeds, have surged from a few thousand in 2016 to over 46,000 annually in 2025[3].

[3] https://www.cvedetails.com/browse-by-date.php

Additionally, from the estimated 400 million terabytes of data generated online each day [99], open-source intelligence (OSINT) must be gathered, filtered and analysed. Managing this manually is far beyond the capacity of traditional human-centric analysis [100].

Compounding this challenge, the cybersecurity domain suffers from inconsistent threat taxonomies, fragmented data sources, and a pronounced Western-centric bias [101]. This convergence of data overload, analytical fragmentation, and intelligence gaps creates a critical bottleneck in cyber defence. Forward-thinking organisations, are exploring how LLMs can power specialist agents for automated data retrieval and insight generation within SOC environments.

This study explored how a MAS architecture could support SOC analysts by automating critical tasks such as data analysis and cross-correlation. The prototype, shown in Fig. 4, implemented a SIE to enable seamless integration of structured SOC datasets with unstructured OSINT feeds. To enhance usability, the solution incorporated a conversational chatbot interface, allowing analysts to query and navigate complex data sources intuitively and in real time.

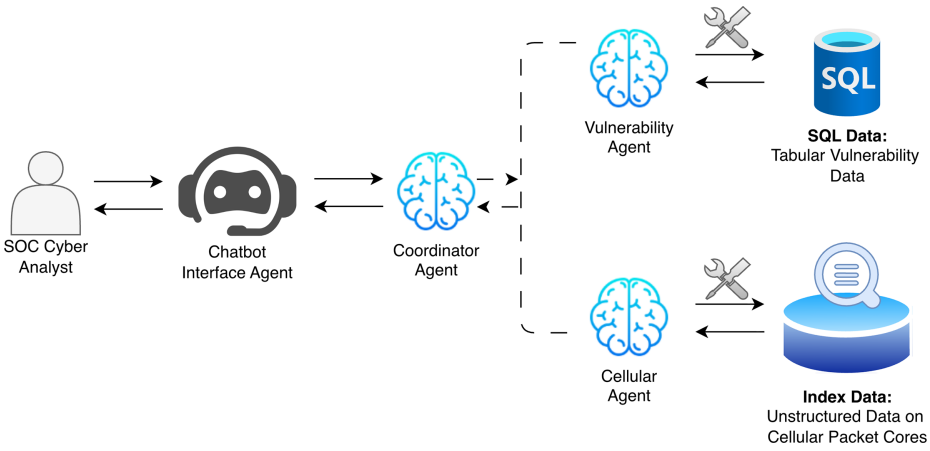

Figure 4: MAS architecture deployed in a Telecom SOC case study. The design integrates specialised agents for threat intelligence retrieval and correlation, coordinated through a central supervisor agent, with a conversational interface for analyst interaction.

Fig. 5 depicts an example of the reasoning stages logged within the Telecoms SOC case study. In this example, a user initiates a query via the Chatbot Interface Agent, requesting details on a specific CVE and its associated CVSS score. The Coordinator Agent delegates tasks to two specialised agents: the Cellular Agent, which performs vector similarity searches to extract contextual threat intelligence, and the Vulnerability Agent, which executes SQL queries using CoT reasoning to retrieve CVSS scores. This multi-agent workflow demonstrates the MAS paradigm's ability to integrate heterogeneous data sources, combining structured vulnerability databases with unstructured threat reports, while preserving modularity and adaptability for SOC operations. Furthermore, the Coordinator Agent optimises resource utilisation by recognising when a query requires only a single source and delegating tasks accordingly, thereby reducing unnecessary computational overhead.

To ensure compliance with telecommunications security policies, the MAS solution was deployed as a portable Docker container, enabling secure on-premises implementation. The PoC was tested with key stakeholders and received highly positive feedback on usability and potential impact. Analysts noted that the system could significantly reduce workload by automating data navigation and correlation across heterogeneous sources.

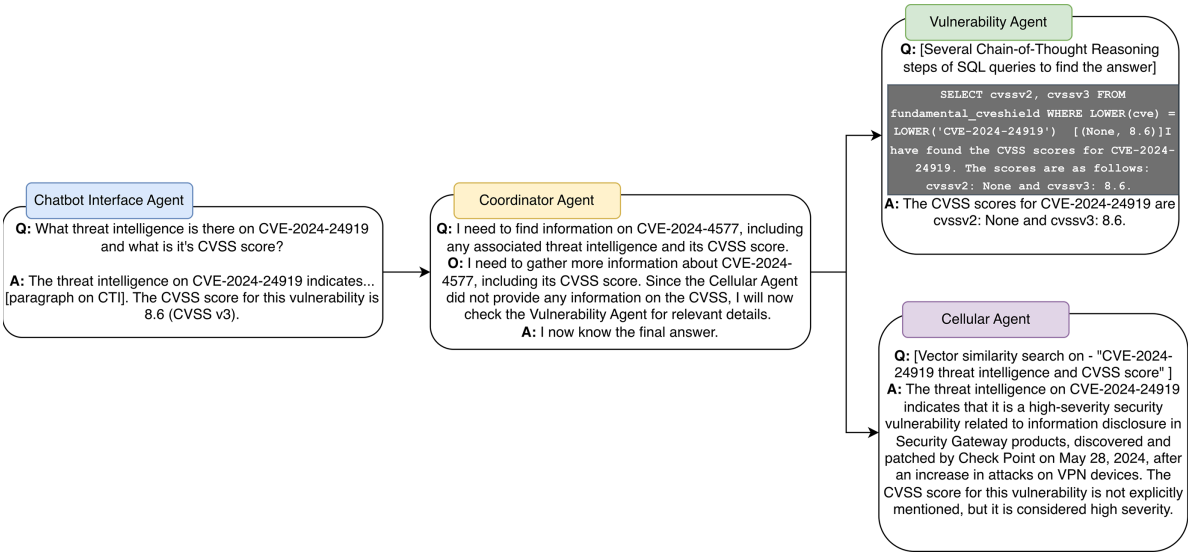

Figure 5: Example of MAS reasoning stages in the Telecoms SOC case study. This demonstrates seamless integration of structured and unstructured data sources, supporting efficient and context-rich responses to analyst queries.

Following the demonstration and subsequent discussion with the Chief Information Security Officer (CISO), they planned further investigation into this approach to augment existing SOC analyst tooling. This feedback indicates the viability of MAS as a method for scaling SOC operations amid increasing cyber complexity. It underscores both the operational benefits for analysts and the strategic potential for senior leadership pursuing digital transformation toward a data-centric SOC model.

5.3 CS2—National Asset Register—Data Sanitation & Retrieval

This case study focuses on a large UK national organisation managing extensive assets and inventories across numerous sites. Over time, siloed datasets emerged at every location, each adopting unique data capture methods and structural conventions. This fragmentation resulted in poor data standardisation, making asset inventory management and retrieval highly inconsistent and inaccessible for staff. Consequently, each site's dataset required a different interpretation approach, complicating integration efforts.

While a long-term initiative to standardise data storage and schema design was planned, an interim solution was needed to enable seamless access and querying of asset information across all UK sites. To address this, a MAS PoC was developed using the design paradigm outlined in this paper. The solution implemented an SIE MAS architecture focused on bridging inconsistencies between structured and unstructured datasets.

Through stakeholder discovery sessions and business requirement workshops, the SIE MAS architecture shown in Fig. 6a was defined. It features a coordinated framework of specialised agents, beginning with a coordinator agent that dynamically hands off prompts to the appropriate data-retrieval specialists that are trained and equipped with data retrieval tooling. To maximise accessibility for non-technical staff, the interface was designed as a natural language chatbot, requiring minimal training and enabling intuitive interaction.

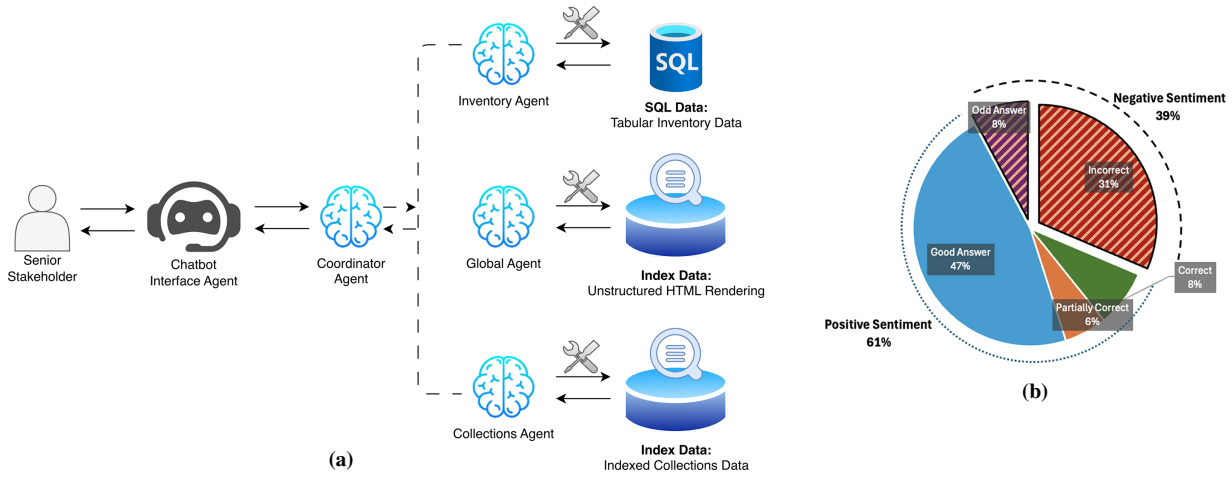

Figure 6: (**a**) MAS architecture for the National Asset Register case study, featuring dynamic agent orchestration for cross-site data retrieval and sanitation. (**b**) Sentiment analysis of stakeholder feedback, categorised into positive and negative responses, indicating overall favourable reception but with clear area for improvement.

The platform-agnostic design was deployed in the organisation's Microsoft Azure cloud environment, leveraging native agent services to ensure scalability and compliance. For validation, the PoC was tested by ten senior staff members over one month, with open-ended qualitative feedback collected with a record of comments per use of the solution. Sentiment analysis using Lexicon-Based Matching [102] categorised responses into positive and negative segments, as shown in Fig. 6b. Overall, feedback at 61% was weighted slightly more positively with detail in the feedback indicating how adjustments could be made to quickly improve this. This assumption for the PoC's potential was acknowledged as the stakeholders requested follow-up work to expand the solution. Additionally, planning discussions were held on evolving the internal tool into a customer-facing chatbot to enhance user experience by providing holistic insights into their national assets.

This PoC demonstrated that SIE MAS architectures have potential to unify fragmented datasets without requiring immediate schema standardisation. The conversational interface lowered adoption barriers for non-technical users, while cloud deployment ensured scalability and compliance. However, considering the speed of development taking place within just one month, results indicated that improvements on consistency and reliability are needed. Despite these limitations, strong stakeholder interest in extending the solution to customer-facing applications underscores its potential for broader organisational impact with further testing.

5.4 CS3—Customer Service—Automated Queries & Complaint Responses

This case study examines a UK utilities company seeking to improve the efficiency and consistency of its customer service operations, with a particular focus on automating first-time response (FTR) emails. The organisation had previously commissioned two external contractors to address this challenge; however, neither produced a solution that was either viable or scalable. The present study therefore explored a MAS approach to assess whether targeted automation could deliver measurable performance gains while retaining necessary human oversight.

Following discovery sessions, analysis of the incumbent architecture, and stakeholder consultations, key opportunities for automation were identified in email triage, knowledge retrieval, and response drafting. The original design was adapted as iterative spiral development was conducted [103]. The final resulting MAS

architecture (Fig. 7) integrates domain-specific contextual knowledge from Knowledge Base Articles (KBA) with up-to-date customer records from the Customer Relationship Management (CRM) system. Specialised agents perform information extraction from inbound emails, retrieval and synthesis of relevant KBA and CRM context, and generation of draft responses aligned with predefined templates. Human-in-the-loop checkpoints remain embedded for risk-sensitive decisions and quality assurance, particularly at the user-refinment stage to transition a FTR refined silver email into a gold email ready to send. This configuration constitutes a domain-specific automation strategy that exceeds the flexibility of conventional scripted or purely rule-based pipelines.

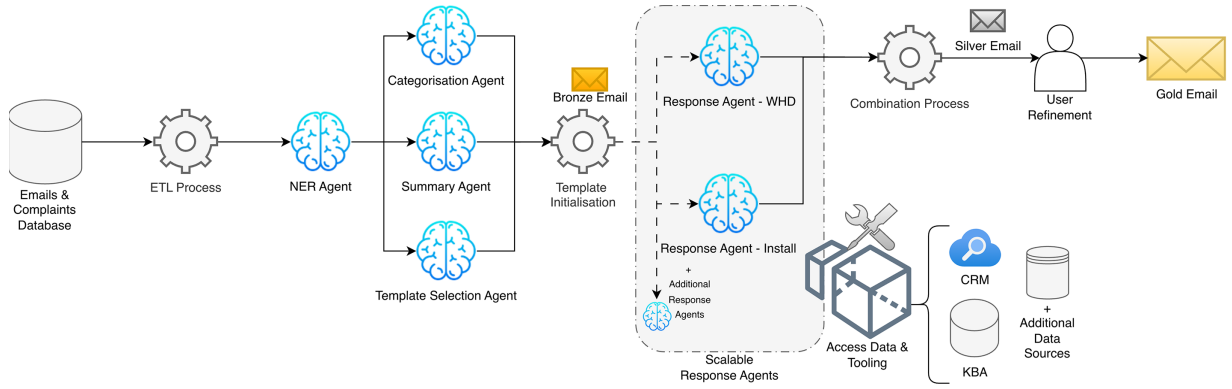

Figure 7: End-to-end multi-agent system (MAS) for customer service automation, where NER-driven email triage (categorisation, summarisation, and template selection) feeds contextual knowledge-retrieval agents (e.g., Warm Home Discount and Install) to generate first-time response drafts, with human-in-the-loop validation for compliance and quality control.

The solution was implemented using a platform-agnostic design and deployed to Microsoft Azure, utilising Azure's agent services and hosted within the Microsoft Fabric ecosystem. Integration with Power BI provided operational monitoring and business analytics. In this environment, the MAS produced approximately five FTR emails per minute at an estimated operational cost of ~£0.05 per email, with horizontal scalability achievable through distributed, parallel execution. By comparison, the fully manual baseline generated about three FTR emails per minute at a substantially higher ~£0.33 per-email according to the business' estimates, indicating potential improvements in both throughput and cost efficiency.

From an engineering perspective, the MAS paradigm accelerated delivery of initial capabilities. As a point of comparison, the lead developer reported on the project a conventional regular-expression (regex) categorisation baseline required ~4.5 h to achieve first testable outputs, whereas an equivalent MAS specialist agent produced comparable first results in less than an hour, reflecting gains in development velocity and extensibility.

A pilot User Acceptance Test (UAT) involved three senior customer response agents who assessed 24 real customer emails. Participants tracked in Fig. 8a rated system output of FTR using five Likert-scale dimensions: correctness, usefulness, clarity, groundedness, and safety (1 = poor; 5 = excellent)[4]. The evaluation considered two tasks: (i) categorisation of incoming emails and (ii) FTR generation aligned with organisational templates.

[4]For the interested reader, full details and data are hosted at: https://osf.io/4dzmv/overview

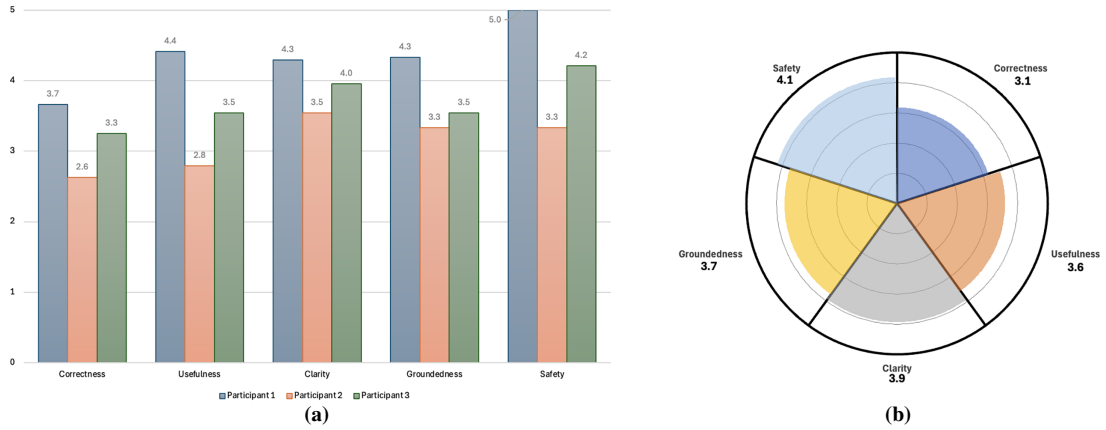

Figure 8: (**a**) Evaluation results from the UAT per participant, showing Likert-scale ratings across five dimensions: correctness, usefulness, clarity, groundedness, and safety. (**b**) Same results aggregated as the mean across participants.

Email categorisation achieved 100% accuracy when aggregating judgements across all participants. For FTR generation email quality, mean scores (Fig. 8b) on each dimension exceeded 3/5, with clarity and safety averaging ~4/5. Although inter-participant variability was observed (e.g., safety ratings ranged from 3.3 to 5.0), overall performance surpassed the manual baseline on time-to-draft and exhibited promising quality indicators.

This pilot study indicates that an MAS architecture can materially enhance operational efficiency and response consistency in customer service workflows, delivering measurable gains in throughput and unit cost while accelerating development cycles. Notwithstanding the limited scope of evaluation and the identified variability in response quality, stakeholder interest in scaling the system and extending it towards first-contact resolution underscores its potential as a scalable, adaptable approach to enterprise customer engagement. Future work that prioritises expanded trials is needed, with rigorous A/B testing against manual baselines, and targeted tuning to reduce variance across safety and clarity metrics.

5.5 SIE vs. Hierarchical

Across cases, *SIE* (CS1–CS2) suited heterogeneous, multi-source retrieval with evolving schemas, where minimal-source routing and per-source governance mattered; primary benefits were rapid navigation across silos and lower coupling between data and reasoning roles. *Hierarchical* (CS3) suited a stable, policy-constrained pipeline, where explicit gates, templated drafting, and human oversight dominated; benefits were predictable latency and less-intervention needed, with a trade-off in more explorative generative outputs. Differences reflect underlying data characteristics (heterogeneity vs. standardisation) and business requirements (governance gates vs exploration), which should guide pattern selection.

6 Discussions

The case series presented in this paper provides an opportunity to reflect on the practical implications of adopting MAS architectures powered by LLMs as the agent reasoning engine. Drawing on insights from three distinct domains: telecommunications security, national asset management, and customer service automation, the discussion considers the strengths, limitations, and future directions of this emerging paradigm. The following subsections examine three key themes that emerged across all case studies.

6.1 Rapid Prototype Development

The case series demonstrates that MAS architectures could accelerate prototype development, enabling functional solutions to be developed in weeks rather than months. For instance, in Case Study 3 (Customer Service Automation), two previous contractors failed to deliver any viable solution after several months of development. In contrast, using the MAS approach, the prototype and UAT were completed within one month.

Across CS1–CS3, rapid prototyping arose from three design mechanisms rather than model choice alone.

1. **Abstraction from process engineering:** LLM-enabled agents infer many of the intermediate steps that would traditionally require explicit process engineering. ReAct-style reasoning–action loops allow agents to determine tool use, sequencing, and problem-decomposition at run time rather than relying on manually predefined workflows.
2. **Modularity:** patternised handoff/tool-flow let us isolate retrieval, synthesis, and governance, reducing cognitive load and enabling parallel development of leaf capabilities.
3. **Reuse at the pattern layer:** once a coordinator–specialist scaffold exists, new tasks require only swapping or extending agents/tool adapters and adjusting handoff criteria, not re-architecting pipelines; this explains the shorter time-to-first-output observed in CS3 compared to a regex baseline.

Together these mechanisms explain why prototypes were delivered in weeks while preserving a path to add governance gates later.

This acceleration is further supported by the platform-agnostic nature of MAS, which allows deployment across on-premises containers and cloud ecosystems without major infrastructure changes, effectively bridging interoperability gaps. However, this speed advantage introduces a critical trade-off: while early-stage development is expedited, transitioning to production requires extensive tuning to address alignment issues and additional compute costs inherent to LLMs, such as variability in outputs and compliance constraints. These challenges may offset initial time savings, underscoring the need for future research to evaluate full lifecycle costs, reliability, and performance benchmarks for MAS at scale.

6.2 Domain-Specific Adaptability and Efficiency

A key strength of MAS architectures lies in their adaptability across diverse domains, particularly when combined with LLM reasoning and contextual enrichment through RAG and tool integration. Unlike rigid, rule-based workflows, MAS enables dynamic orchestration of specialised agents, allowing systems to align with domain-specific semantics, compliance requirements, and operational constraints.

This flexibility was evident in the case studies: for example, the SOC prototype automated cross-correlation of heterogeneous threat intelligence sources, a task traditionally requiring extensive manual analysis, while the customer service solution integrated CRM and knowledge base data to generate contextually grounded responses that a specialist would typically have to learn.

Adaptability derived from two interacting properties. First, LLM + RAG decouples reasoning from static training corpora: retrieval agents pull domain-specific context at query time, so only tool connectors and retrieval prompts require domain changes. Second, the SIE topology (CS1–CS2) specialises agents by dataset/service and routes to a minimal subset per request; this localises schema drift and policy differences to adapters rather than to global prompts. In practice, moving between departments or sites required replacing/adjusting connectors and indices, not refactoring the orchestration. The trade-off is additional adapter and index maintenance, which must be balanced against the gains in reuse and governance.

Beyond adaptability, MAS delivers efficiency gains by reducing reliance on large development teams and accelerating innovation cycles. These benefits extend beyond speed; they reflect the ability of MAS to scale across problem spaces without wholesale redesign, positioning this paradigm as a reusable foundation for bespoke, high-stakes applications.

The immediate consequence is disruptive organisational impact, as highlighted in emerging enterprise AI research [104]. Rapid domain-specific adaptability could reshape workforce structures, potentially reducing the need for certain specialised roles and prompting a re-evaluation of organisational strategies [105].

6.3 Human Oversight as a Persistent Requirement

Despite the automation gains observed, all case studies reaffirm the necessity of human involvement, either in-the-loop or on-the-loop, to ensure accountability, compliance, and contextual judgement. As emphasised by Brown [106], "The future of leadership lies in complementarity [AI], not replacement." MAS architectures can automate routine tasks and provide actionable insights, but they cannot replicate the nuanced decision-making and ethical considerations inherent to human oversight. In practice, this manifested as supervisory checkpoints in the customer service PoC and strategic decision-maker roles in the SOC and asset management solutions. These findings reinforce the argument that MAS should be positioned as augmentative rather than substitutive technologies within organisational workflows.

6.4 Key Limitations

The case series highlights several strengths of the MAS paradigm, such as the initial rapid development lifecycle. However, key limitations are presented through observations by the developers, users and stakeholder feedback. The dependency on LLM reasoning, even when paired with domain specific context via RAG and other tools, inherently has risks of hallucination, grounding errors and and interpretability that can propagate throughout the MAS. This was apparent in the results from Case study 2 and 3, where measurements of reliability and correctness indicate room for improvement. In CS2, some responses exhibited incomplete citation coverage across siloed sources, reflecting retrieval noise and index freshness, an instance of partial grounding rather than egregious hallucination. In CS3, inter-participant variance on safety/clarity ratings indicates sensitivity to prompt and template phrasing. Dynamic routing in CS1–CS2 improved adaptability but introduced route variance and additional cost at audit checkpoints when full traces were enabled. Extended effort is needed to create robust design patterns and guardrails that scale solutions beyond initial rapid development to transition the prototypes into production/release. These issues become more pronounced in high-stakes domains such as cybersecurity or highly regulated industries, reinforcing the need for robust validation pipelines and human-in-the-loop oversight.

Finally, while MAS architectures offer platform-agnostic deployment flexibility, they remain constrained by compute requirements and dependency management. Organisations with limited infrastructure or strict data residency policies may face integration hurdles, particularly when scaling across heterogeneous environments.

6.5 How the Results Address the Literature Gaps

Section 2 identified two gaps: a pattern-level disconnect between academic, narrow-domain evaluations and industry, tool-centric practice, and a need for more robust quantifiable evidence around real-world implementations. Our formalised axes (control flow, interaction style, history sharing, topology) and the SIE definition provides further systemisation of the academic-industry design-pattern bridge. The cases operationalise this bridge: SIE for heterogeneous, multi-source retrieval (CS1–CS2) and a Hierarchical type for regulated, templated communication (CS3). Whilst robust quantifiable metrics were not captured, the

initial progress towards defining these and understanding what would be needed to capture these in future work have been emphasised.

6.6 Actionable Future Work

1) Controlled follow-up studies with quantifiable metrics.

Controlled follow-up studies using pattern-aware evaluation indicators outlined at the end of the literature review Section 2.5 (e.g., development velocity, throughput, unit cost, groundedness rubric, citation coverage/source agreement). Planning predefine task sets, collecting the same small set of metrics across trials, and reporting simple descriptive results with clear inclusion/exclusion rules for measurable quality ratings.

2) Scaling MAS to test agent-count and coordination limits.

Scaling the number of agents and the diversity of roles to probe coordination limits across topologies (Supervisor, Hierarchical, Swarm, SIE). By tracking basic signals of stress—route variance (how often paths change on identical inputs), handoff depth (maximum hops), and budget hits (latency/token caps) to identify where performance or cost begins to degrade.

3) Controlled longitudinal observational study of MAS adoption

Conduct a controlled, long-term observational study that follows end-users as they transition from the current baseline (manual process or incumbent tooling) to the MAS-based solution. Use two matched cohorts (transition group vs. control group that stays on the baseline for the same period) and track a minimal, consistent metric set over 8–12 weeks: productivity (items completed per hour; average minutes per item), quality evaluator metrics using the latest industry and/or academic standards, and agreed key performance indicators (KPIs) with an invested client.

7 Conclusion

This study provides systematisation of emerging design patterns for LLM-enabled MAS and evaluated their practical utility across diverse domains through three real-world case studies. The results demonstrate that MAS architectures can significantly accelerate prototyping, reduce development overhead, and deliver adaptable solutions for complex, domain-specific challenges. By leveraging modularity, specialisation, and dynamic orchestration, MAS offers a reusable paradigm that bridges gaps between rigid rule-based workflows and bespoke engineering approaches.

The findings also reinforce persistent limitations. Variability in LLM behaviour, risks of hallucination, and interpretability challenges remain critical barriers to production maturity, particularly in high-stakes or regulated environments. Furthermore, while MAS architectures exhibit strong potential for scalability and interoperability, they introduce dependencies on compute resources and governance frameworks that organisations must address.

Future research will focus on iterative user studies to rigorously evaluate and refine this approach, informing the development of enhanced design features that meet stringent requirements. These efforts will prioritise strengthening validation pipelines and implementing robust guardrails [107] to mitigate reliability and safety risks.

Acknowledgement: We are especially grateful to Tim Williams for his insightful guidance during the refinement of this manuscript. This work was supported by Kaze Digital & Data Ltd.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm the following contributions: Conceptualization, Harri Renney and Maxim Nethercott; methodology, Harri Renney and Nathan Renney; software, Maxim Nethercott; validation, Peter Hayes and Nathan Renney; writing, Harri Renney and Nathan Renney. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study are openly accessible via the Open Science Framework (OSF) at: https://osf.io/4dzmv/overview and are released under a Creative Commons Attribution-NonCommercial 4.0 International (CC-BY-NC 4.0) licence. The repository includes anonymised evaluation artefacts, user-testing outputs, and supplementary materials referenced in the case studies. Proprietary organisational datasets used within the Telecoms, National Asset Register, and Utilities case studies are not publicly available due to commercial sensitivity and data-protection obligations; however, detailed descriptions of these datasets and the retrieval methods employed are included in the manuscript to support transparency and reproducibility.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Case Study Implementation Details

Table A1: Implementation details for the three case studies (CS1–CS3). This table consolidates configuration parameters to support reproducibility and comparison.

Field	CS1—Telecoms SOC	CS2—National Asset Register	CS3—Utilities Customer Service
Topology	SIE (dataset-specialist agents + coordinator)	SIE (dataset-specialist agents + coordinator)	Hierarchical (coordination tier + specialist leaves)
Approx. agents/roles	4: interface/chat, coordinator, retriever(s) for CVE/OSINT and cellular datasets	5: interface/chat, coordinator, retriever(s) for inventory, global and collections datasets	6: Named-Entity Recognition (NER) extractor, three specialist email analysis agents, and two (with room to scale) specialist email type generator agents
Control flow	Dynamic source selection; explicit tool use	Dynamic source selection; explicit tool use	Explicit end-to-end pipeline
Interaction style	Handoff between chatbot to coordinator to specialist agents, tool-flow at leaves	Handoff between chatbot to coordinator to specialist agents, tool-flow at leaves	Handoff across stages; tool-flow for data connections
History sharing policy	Full-Trace for Coordinator agent with specialist agents. Final result returned to chatbot	Full-Trace for Coordinator agent with specialist agents. Final result returned to chatbot	Final-result between tiers; trace snapshots at Human-In-The-Loop gate
Tool/Data interfaces	SQL connector (Vulnerability DB), vector index for unstructured cellular intel	SQL connector (asset inventory DB), internet API for unstructured HTML, vector index of site collections data	CRM and KBA adapters; template renderer; Fabric/Power BI for monitoring
Deployment environment	On-prem Docker container (policy compliance)	Azure-hosted using native agent services	Azure environment; integrated with Microsoft Fabric
Governance/human oversight	Selective audit traces; coordinator-enforced policy checks	Audit logging at exception flows; oversight for high-risk queries	Human-in-the-loop approval on drafts; templated responses; stakeholder monitoring via PowerBI

(Continued)

Table A1 (continued)

Field	CS1—Telecoms SOC	CS2—National Asset Register	CS3—Utilities Customer Service
Evaluation artefacts	Stakeholder feedback; exemplar reasoning traces	Stakeholder feedback; proxies: citation coverage & semantic analysis	Pilot UAT (n = 3, 24 emails): throughput, unit cost, five-dimensional Likert ratings; published data artefacts
Participant expertise	1 SOC Analysts, 1 Security Management, 1 Chief of Security	1 Senior Operations Manager, 2 Asset Managers	3 Senior Customer Response Agents

References

1. Lesser VR, Corkill DD. The application of artificial intelligence techniques to cooperative distributed processing. In: Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI'79). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1979. p. 537–40.
2. Hewitt C. Viewing control structures as patterns of passing messages. Artif Intell. 1977;8(3):323–64. doi:10.1016/0004-3702(77)90033-9.
3. Abbas HA, Shaheen SI, Amin MH. Organization of multi-agent systems: an overview. arXiv:1506.09032. 2015.
4. Dorri A, Kanhere SS, Jurdak R. Multi-agent systems: a survey. IEEE Access. 2018;6:28573–93. doi:10.1109/access.2018.2831228.
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems. Red Hook, NY, USA: Curran Associates, Inc.; 2017. doi:10.65215/ctdc8e75.
6. Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, et al. A comprehensive overview of large language models. ACM Trans Intell Syst Technol. 2025;16(5):1–72. doi:10.1145/3744746.
7. Sastry G, Heim L, Belfield H, Anderljung M, Brundage M, Hazell J, et al. Computing power and the governance of artificial intelligence. arXiv:2402.08797. 2024.
8. Renney H, Nethercott M, Williams O, Evetts J, Lang J. Reimagining the data landscape: a multi-agent paradigm for data interfacing. In: 2025 8th International Conference on Data Science and Machine Learning Applications (CDMA). Piscataway, NJ, USA: IEEE; 2025. p. 114–9.
9. Charles Munyao JN. Natural language processing with transformer-based models: a meta-analysis. J Artif Intell. 2025;7(1):329–46. doi:10.32604/jai.2025.069226.
10. Zhang W, Wan C, Zhang Y, Ym C, Tian X, Shen X, et al. Interpreting and improving large language models in arithmetic calculation. In: Proceedings of the 41st International Conference on Machine Learning; 2024 Jul 21–27; Vienna, Austria. p. 59932–50.
11. Gómez-Rodríguez C, Williams P. A confederacy of models: a comprehensive evaluation of LLMs on creative writing. arXiv:2310.08433. 2023.
12. Fakhoury S, Naik A, Sakkas G, Chakraborty S, Lahiri SK. LLM-based test-driven interactive code generation: user study and empirical evaluation. arXiv:2404.10100. 2024.
13. Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, et al. Gpt-4 technical report. arXiv:2303.08774. 2023.
14. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. Palm: scaling language modeling with pathways. J Mach Learn Res. 2023;24(240):1–113.
15. Brin D, Sorin V, Konen E, Nadkarni G, Glicksberg BS, Klang E. How large language models perform on the United States medical licensing examination: a systematic review. MedRxiv. 2023. doi:10.1101/2023.09.03.23294842.
16. Katz DM, Bommarito MJ, Gao S, Arredondo P. Gpt-4 passes the bar exam. Philos Trans R Soc A. 2024;382(2270):20230254. doi:10.1098/rsta.2023.0254.
17. Wu J, Yang S, Zhan R, Yuan Y, Chao LS, Wong DF. A survey on LLM-generated text detection: necessity, methods, and future directions. Comput Linguist. 2025;51(1):275–338. doi:10.1162/coli_a_00549.

18. Popescu-Apreutesei LE, Iosupescu MS, Cristiana Necula S, Păvăloaia VD. Upholding academic integrity amidst advanced language models: evaluating BiLSTM networks with GloVe embeddings for detecting AI-generated scientific abstracts. Comput Mat Cont. 2025;84(2):2605–44. doi:10.32604/cmc.2025.064747.
19. Xu Z, Sheng VS. Detecting AI-generated code assignments using perplexity of large language models. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'24/IAAI'24/EAAI'24. Palo Alto, CA, USA: AAAI Press; 2024. p. 23155–62. doi:10.1609/aaai.v38i21.30361.
20. Chanen A. What I learned from Bloomberg's experience of building their own LLM; 2023 [cited 2026 Feb 20]. Available from: https://www.linkedin.com/pulse/what-i-learned-from-bloombergs-experience-building-own-chanen-phd/.
21. Luyen L, Abel MH. How well do LLMs predict prerequisite skills? Zero-shot comparison to expert-defined concepts. arXiv:2507.18479. 2025.
22. Alto V. Building LLM powered applications: create intelligent apps and agents with large language models. Birmingham, UK: Packt Publishing Ltd.; 2024.
23. Wang H, Zhao S, Qiang Z, Xi N, Qin B, Liu T. Beyond direct diagnosis: lLM-based multi-specialist agent consultation for automatic diagnosis. arXiv:2401.16107. 2024.
24. Kalyuzhnaya A, Mityagin S, Lutsenko E, Getmanov A, Aksenkin Y, Fatkhiev K, et al. LLM agents for smart city management: enhancing decision support through multi-agent AI systems. Smart Cities. 2025;8(1):19.
25. Zhang Y, Saber AM, Youssef A, Kundur D. Grid-agent: an LLM-powered multi-agent system for power grid control. arXiv:2508.05702. 2025.
26. Rezaee H, Abdollahi F. Average consensus over high-order multiagent systems. IEEE Trans Autom Cont. 2015;60(11):3047–52. doi:10.1109/tac.2015.2408576.
27. Sarker S, Arefin MS, Kowsher M, Bhuiyan T, Dhar PK, Kwon OJ. A comprehensive review on big data for industries: challenges and opportunities. IEEE Access. 2022;11(3):744–69. doi:10.1109/access.2022.3232526.
28. Escobar CA, McGovern ME, Morales-Menendez R. Quality 4.0: a review of big data challenges in manufacturing. J Intell Manufact. 2021;32(8):2319–34. doi:10.1007/s10845-021-01765-4.
29. Pallardy C. How much data is too much for organizations to derive value? 2024 [cited 2025 Nov 13]. Available from: https://www.informationweek.com/data-management/how-much-data-is-too-much-for-organizations-to-derive-value-.
30. Ugarte R. The data mirage: why companies fail to actually use their data. New York, NY, USA: Business Expert Press; 2021.
31. Sellbery. Why most companies are still struggling to use their data effectively. 2025 [cited 2025 Nov 13]. Available from: https://sellbery.com/blog/why-most-companies-are-still-struggling-to-use-their-data-effectively/.
32. Carruthers A. Breaking data silos. In: Building the snowflake data cloud: monetizing and democratizing your data. Heidelberg, Germany: Springer; 2022. p. 29–50.
33. Johnny R. Addressing data silos with governance frameworks in financial organizations. 2023 [cited 2025 Nov 13]. Available from: https://www.researchgate.net/publication/387174207_Addressing_Data_Silos_with_Governance_Frameworks_in_Financial_Organizations.
34. Kelly P. Legacy systems in government: how to break data silos and innovate. 2025 [cited 2025 Nov 13]. Available from: https://blog.govnet.co.uk/technology/legacy-systems-in-government-how-to-break-data-silos-and-innovate.
35. Muscolino H, Machado A, Rydning J, Vesset D. Untapped value: what every executive needs to know about unstructured data. Needham, MA, USA: IDC Research Ltd.; 2023.
36. Tredinnick L, Laybats C. Managing unstructured information. London, UK: SAGE Publications; 2024.
37. Inmon WH, Nesavich A. Tapping into unstructured data: integrating unstructured data and textual analytics into business intelligence. London,UK: Pearson Education; 2007.
38. Mohagheghi P, Haugen Ø. Evaluating domain-specific modelling solutions. In: International Conference on Conceptual Modeling. Heidelberg, Germany: Springer; 2010. p. 212–21.

39. Ahire PR, Hanchate R, Kalaiselvi K. Optimized data retrieval and data storage for healthcare applications. In: Predictive data modelling for biomedical data and imaging. Gistrup, Denmark: River Publishers; 2024. p. 107–26. doi:10.1201/9781003516859-6.
40. Khairnar SL, Patil GV, Bankar PD, Bharade PP, Sonawane HD. Attribute based secure data retrieval system for decentralized disruption tolerant military networks. Int J Recent Innov Trends Comput Commun. 2016;2(12):4105–8. doi:10.9756/bijsesc.8283.
41. Kazmi ST, Qayyum RF. Data regulations in the UK's financial sector. 2025 [cited 2025 Nov 13]. Available from: https://securiti.ai/data-regulations-in-the-uk-financial-sector/.
42. Deng Q, Li J, Chai C, Liu J, She J, Jin K, et al. Unstructured data analysis using LLMs: a comprehensive benchmark. arXiv:2510.27119. 2025.
43. Li Y, Tan Z, Xiao W. LLM for uniform information extraction using multi-task learning optimization. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. Cham, Switzerland: Springer; 2024. p. 17–29.
44. Berti L, Giorgi F, Kasneci G. Emergent abilities in large language models: a survey. arXiv:2503.05788. 2025.
45. Singh C, Inala JP, Galley M, Caruana R, Gao J. Rethinking interpretability in the era of large language models. arXiv:2402.01761. 2024.
46. Meskó B, Topol EJ. The imperative for regulatory oversight of large language models (or generative AI) in healthcare. npj Digital Med. 2023;6(1):120. doi:10.1038/s41746-023-00873-0.
47. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, et al. Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inform Process Syst. 2022;35:24824–37. doi:10.52202/068431-1800.
48. Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, et al. Tree of thoughts: deliberate problem solving with large language models. In: Advances in neural information processing systems. Red Hook, NY, USA: Curran Associates, Inc.; 2024. doi:10.52202/075280-0517.
49. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inform Process Syst. 2020;33:9459–74.
50. Oche AJ, Folashade AG, Ghosal T, Biswas A. A systematic review of key retrieval-augmented generation (RAG) systems: progress, gaps, and future directions. arXiv:2507.18910. 2025.
51. Kiela D. RAG agents in prod: 10 lessons we learned—douwe kiela, creator of RAG. 2025 [cited 2025 Nov 13]. Available from: https://www.youtube.com/watch?v=kPL-6-9MVyA&t=267s.
52. Onobhayedo P, Igah C, Okonkwo A. Using artificial intelligence techniques in the requirement engineering stage of traditional SDLC Process. J Artif Intell. 2024;6(1):379–401. doi:10.32604/jai.2024.058649.
53. Lin F, Kim DJ, Chen T. When LLM-based code generation meets the software development process. arXiv:2403.15852. 2024.
54. Kearney M. Bots of the SOC. 2024 [cited 2025 Nov 13]. Available from: https://www.youtube.com/watch?v=WLP5eEqmbxQ.
55. Splunk GitHub. botsv3. 2020 [cited 2025 Nov 13]. Available from: https://github.com/splunk/botsv3.
56. Pan MZ, Cemri M, Agrawal LA, Yang S, Chopra B, Tiwari R, et al. Why do multiagent systems fail? In: Proceedings of the ICLR 2025 Workshop on Building Trust in Language Models and Applications; 2025 Apr 28; Singapore.
57. Ding Y, Shi T. Sustainable LLM serving: environmental implications, challenges, and opportunities: invited paper. In: 2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC). Piscataway, NJ, USA: IEEE; 2024. p. 37–8.
58. Hinds PS, Bedinger Miller A. Our words and the words of artificial intelligence: the accountability belongs to us. Cancer Care Res Online. 2023;3(2):e041. doi:10.1097/cr9.0000000000000041.
59. Kapania S, Wang R, Li TJJ, Li T, Shen H. I'm Categorizing LLM as a productivity tool': examining ethics of LLM Use in HCI research practices. Proc ACM Hum-Comput Interact. 2025;9(2):CSCW102:1–26. doi:10.1145/3711000.
60. Jiao J, Afroogh S, Xu Y, Phillips C. Navigating LLM ethics: advancements, challenges, and future directions. AI Ethics. 2025;5(6):5795–819. doi:10.1007/s43681-025-00814-5.
61. MIT. NANDA: the internet of AI agents. [cited 2025 Nov 13]. Available from: https://nanda.media.mit.edu.

62. Raskar R, Chari P, Zinky J, Lambe M, Grogan JJ, Wang S, et al. Beyond dns: unlocking the internet of AI agents via the nanda index and verified agentfacts. arXiv:2507.14263. 2025.
63. Topsakal O, Akinci TC. Creating large language model applications utilizing langchain: a primer on developing llm apps fast. In: Proceedings of the International Conference on Applied Engineering and Natural Sciences; 2023 Jul 10–12; Konya, Turkey. p. 1050–6.
64. Darmon T. Unleashing the Power of langchain expression language (LCEL): from proof of concept to production. 2024 [cited 2025 Nov 13]. Available from: https://medium.com/artefact-engineering-and-data-science/unleashing-the-power-of-langchain-expression-language-lcel-from-proof-of-concept-to-production-8ad8eebdcb1d.
65. Jeong C. A study on the implementation method of an agent-based advanced RAG system using graph. arXiv:2407.19994. 2024.
66. Wu Q, Bansal G, Zhang J, Wu Y, Zhang S, Zhu E, et al. Autogen: enabling next-gen LLM applications via multi-agent conversation framework. arXiv:2308.08155. 2023.
67. Khan A. Microsoft copilot studio. In: Introducing microsoft copilot for managers: enhance your team's productivity and creativity with generative AI-powered assistant. Heidelberg, Germany: Springer; 2024. p. 621–94.
68. Amazon. Amazon bedrock agents. [cited 2025 Nov 13]. Available from: https://aws.amazon.com/bedrock/agents.
69. Hou X, Zhao Y, Wang S, Wang H. Model context protocol (mcp): landscape, security threats, and future research directions. arXiv:2503.23278. 2025.
70. Microsoft. The AI app and agent factory. [cited 2025 Nov 13]. Available from: https://ai.azure.com.
71. Eloundou T, Manning S, Mishkin P, Rock D. GPTs are GPTs: labor market impact potential of LLMs. Science. 2024;384(6702):1306–8.
72. Kumari Vaddepalli R. AutoSchema: a self-learning framework for detecting and adapting to schema drift in real-time data streams. Eur J Adv Eng Technol. 2023;10(7):94–100.
73. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: synergizing reasoning and acting in language models. arXiv:2210.03629. 2022.
74. Aryal S, Do T, Heyojoo B, Chataut S, Gurung BDS, Gadhamshetty V, et al. Leveraging multi-AI agents for cross-domain knowledge discovery. arXiv:2404.08511. 2024.
75. Du H, Thudumu S, Vasa R, Mouzakis K. A survey on context-aware multi-agent systems: techniques, challenges and future directions. arXiv:2402.01968. 2024.
76. Cambon A, Hecht B, Edelman B, Ngwe D, Jaffe S, Heger A, et al. Early LLM-based tools for enterprise information workers likely provide meaningful boosts to productivity. Microsoft Res. 2023.
77. Lewkowicz J. While most companies focus on data, only about 16% are 'data-driven'. 2024 [cited 2025 Nov 13]. Available from: https://sdtimes.com/data/while-most-companies-focus-on-data-only-about-16-are-data-driven/.
78. Bakis N, Aouad G, Kagioglou M. Towards distributed product data sharing environments—progress so far and future challenges. Autom Constr. 2007;16(5):586–95. doi:10.1016/j.autcon.2006.10.002.
79. Dunning T, Friedman E. Time series databases: new ways to store and access data. Sebastopol, CA, USA: O'Reilly; 2015.
80. Date CJ. A Guide to the SQL standard. Hoboken, NJ, USA: Addison-Wesley Longman Publishing Co., Inc.; 1989.
81. IBM. What are the key differences between structured and unstructured data? 2025 [cited 2025 Nov 13]. Available from: https://www.ibm.com/think/topics/structured-vs-unstructured-data.
82. Gharehchopogh FS, Khalifelu ZA. Analysis and evaluation of unstructured data: text mining versus natural language processing. In: 2011 5th International Conference on Application of Information and Communication Technologies (AICT). Piscataway, NJ, USA: IEEE; 2011. p. 1–4.
83. Li I, Pan J, Goldwasser J, Verma N, Wong WP, Nuzumlalı MY, et al. Neural natural language processing for unstructured data in electronic health records: a review. Comput Sci Rev. 2022;46(1):100511. doi:10.1016/j.cosrev.2022.100511.
84. Mernik M, Heering J, Sloane AM. When and how to develop domain-specific languages. ACM Comput Surv. 2005;37(4):316–44. doi:10.1145/1118890.1118892.

85. Python Software Foundation. Using the python interpreter. 2025 [cited 2025 Nov 13]. Available from: https://docs.python.org/3/tutorial/interpreter.html.
86. Ferrag MA, Tihanyi N, Debbah M. From LLM reasoning to autonomous AI agents: a comprehensive review. arXiv:2504.19678. 2025.
87. Fu-Hinthorn W. Benchmarking multi-agent architectures. 2025 [cited 2025 Nov 13]. Available from: https://blog.langchain.com/benchmarking-multi-agent-architectures/.
88. PromptForward Team. MCP overload: why your LLM agent doesn't need 20 tools. 2025 [cited 2025 Nov 13]. Available from: https://promptforward.dev/blog/mcp-overload.
89. Yang B, Tang C, Zhao K, Xiao C, Lin C. Effective distillation of table-based reasoning ability from llms. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Stroudsburg, PA, USA: ACL; 2024. p. 5538–50.
90. Leung K. Handoffs-in-langgraph-multi-agent-systems. 2025 [cited 2025 Nov 13]. Available from: https://github.com/kennethleungty/Handoffs-in-LangGraph-Multi-Agent-Systems.
91. Reid A, O'Callaghan S, Carroll L, Caetano T. Risk analysis techniques for governed LLM-based multi-agent systems. arXiv:2508.05687. 2025.
92. Verma M, Bhambri S, Kambhampati S. On the brittle foundations of react prompting for agentic large language models. arXiv:2405.13966. 2024.
93. Moore DJ. A taxonomy of hierarchical multi-agent systems: design patterns, coordination mechanisms, and industrial applications. arXiv:2508.12683. 2025.
94. Hue M. A review of enterprise architecture use in defence. Edinburgh, SA, Australia: Defence Systems Integration Technical Advisory, Joint and Operations Analysis Division, Defence Science and Technology Organisation; 2016.
95. Kapoor M. The evolving role of human-in-the-loop evaluations in advanced AI systems. Eur J Comput Sci Inform Techn. 2025;13(9):115–26. doi:10.37745/ejcsit.2013/vol13n9115126.
96. Microsoft. Retrieval-augmented generation (RAG) evaluators. 2026 [cited 2025 Nov 13]. Available from: https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/evaluation-evaluators/rag-evaluators?view=foundry-classic.
97. Crowley C. Common and best practices for security operations centers: results of the 2019 SOC survey. North Bethesda, MD, USA: SANS Institute; 2019.
98. Zidan K, Alam A, Allison J, Al-sherbaz A. Assessing the challenges faced by security operations centres (SOC). In: Future of Information and Communication Conference. Cham, Switzerland: Springer; 2024. p. 256–71.
99. Loy M. How much data is generated per day. 2025 [cited 2025 Nov 13]. Available from: https://www.techbusinessnews.com.au/blog/402-74-million-terrabytes-of-data-is-created-every-day/.
100. Stolovitch HD, Keeps EJ. Handbook of human performance technology: principles, practices, and potential. Hoboken, NJ, USA: John Wiley & Sons; 2006.
101. Cristiano F, Kurowska X, Stevens T, Hurel LM, Fouad NS, Cavelty MD, et al. Cybersecurity and the politics of knowledge production: towards a reflexive practice. J Cyber Policy. 2023;8(3):331–64. doi:10.1080/23738871.2023.2287687.
102. Contreras JO, Ballera MA, Lagman AC, Raviz JG. Lexicon-based sentiment analysis with pattern matching application using regular expression in automata. In: Proceedings of the 6th International Conference on Information Technology: IoT and Smart City. New York, NY, USA: ACM; 2018. p. 31–6.
103. Sari R, Rifa'i AM, Ahsan MS, Pahlevi MR, Arief MI. The systematic literature review of the spiral development model: topics, trends, and application areas. Int J Res Appl Technol. 2022;2(2):154–71. doi:10.34010/injuratech.v2i2.8372.
104. Păvăloaia VD, Necula SC. Artificial intelligence as a disruptive technology—a systematic literature review. Electronics. 2023;12(5):1102. doi:10.3390/electronics12051102.
105. Simeon Yates AH, McClure S. AI, data analytics, and digital technology use: a survey of the UK workforce. 2025 [cited 2025 Jul 15]. Available from: https://ddrc.uk/wp-content/uploads/2025/02/20250123_DDRC_Report_O_-v11.pdf.

106. Brown PA. The impact of AI on the past, present, and future of leadership—part 1. 2025 [cited 2025 Jul 15]. Available from: https://www.oxford-group.com/insights/the-impact-of-ai-on-leadership/.
107. Mathew ES. Enhancing security in large language models: a comprehensive review of prompt injection attacks and defenses. J Artif Intell. 2025;7(1):347–63.