

Effat University Repository

In-Mig: Geographically Dispersed Agentic LLMs for Privacy-Preserving Artificial Intelligence

Authors	Nauman, Mohammad
Citation	M. Nauman, "In-Mig: Geographically Dispersed Agentic LLMs for Privacy-Preserving Artificial Intelligence," <i>Comput. Mater. Contin.</i> , vol. 87, no. 2, pp. 46, 2026.
DOI	https://doi.org/10.32604/cmc.2026.077259
Publisher	Tech Science Press
Rights	Attribution-NonCommercial-NoDerivatives 4.0 International
Download date	2026-05-16 08:24:56
Item License	http://creativecommons.org/licenses/by-nc-nd/4.0/
Link to Item	https://repository.effatuniversity.edu.sa/handle/20.500.14131/2663



ARTICLE

In-Mig: Geographically Dispersed Agentic LLMs for Privacy-Preserving Artificial Intelligence

Mohammad Nauman*

Department of Computer Science, Effat College of Engineering, Effat University, Jeddah, 22332, Saudi Arabia

*Corresponding Author: Mohammad Nauman. Email: mnauman@effatuniversity.edu.sa

Received: 05 December 2025; Accepted: 25 December 2025; Published: 12 March 2026

ABSTRACT: Large Language Models (LLMs) are increasingly utilized for semantic understanding and reasoning, yet their use in sensitive settings is limited by privacy concerns. This paper presents *In-Mig*, a mobile-agent architecture that integrates LLM reasoning within agents that can migrate across organizational venues. Unlike centralized approaches, *In-Mig* performs reasoning *in situ*, ensuring that raw data remains within institutional boundaries while allowing for cross-venue synthesis. The architecture features a policy-scoped memory model, utility-driven route planning, and cryptographic trust enforcement. A prototype using JADE for mobility and quantized Mistral-7B demonstrates practical feasibility. Evaluation across various scenarios shows that *In-Mig* achieves 92% similarity to centralized baselines, confirming its utility and strong privacy guarantees. These results suggest that migrating, privacy-preserving LLM agents can effectively support decentralized reasoning in trust-sensitive domains.

KEYWORDS: Mobile agents; large language models (LLMs); privacy-preserving AI; decentralized reasoning; trust and security

1 Introduction

Large Language Models (LLMs) are foundational for reasoning and decision-making across law, healthcare, and policy [1], but their utility often depends on access to context-specific, non-public data. Several limitations motivate augmentation. First, models reflect a static training snapshot and miss post-cutoff developments critical in fast-moving domains. Second, domain-specific terminology and proprietary procedures are underrepresented in general corpora, reducing reliability for professional use [2]. Third, hallucinations, plausible yet incorrect outputs, pose serious risks in high, stakes settings [3]. Finally, much essential context (client files, patient records, internal reports) is confidential and cannot be absorbed during pretraining [4].

Retrieval-Augmented Generation (RAG) alleviates some gaps by supplying relevant documents at inference [5], but normally requires transferring sensitive material to external servers, creating privacy, legal, and ethical hazards [6]. Regulatory regimes such as HIPAA, GDPR, and professional privilege further constrain data movement, making straightforward cloud-based augmentation infeasible [7]. Local hosting of open-weight models (e.g., Mistral, LLaMA) reduces external exposure but is resource intensive and does not natively support secure cross-institutional collaboration [8].

To reconcile utility and sovereignty, we propose moving computation to the data via mobile, LLM-augmented agents. Agents visit venues, execute reasoning in containerized, policy-gated environments, and export only sanitized, signed summaries. This design preserves institutional control over raw data



while enabling cross-venue synthesis. We formalize this approach as the Intelligence and Migration (In-Mig) Protocol: agents plan migrations, negotiate access, and iteratively integrate policy-compliant insights, providing a practical path for decentralized, privacy-preserving collaboration [9].

This work makes four contributions: (1) we identify fundamental limitations of RAG-based and self-hosted LLM architectures in privacy-sensitive, cross-institutional settings; (2) we propose the In-Mig Protocol, a privacy-preserving method for enabling LLM-augmented mobile agents to reason across silos without centralizing data; (3) we demonstrate a working prototype using the JADE agent platform, the JIPMS migration system, and a locally served Mistral-7B model via Ollama; and (4) we evaluate the system in a multi-venue simulation involving legal, medical, and regulatory datasets, showing autonomous route planning, privacy enforcement, and iterative revisitation in decentralized reasoning workflows.

In the remainder of the paper, [Section 2](#) reviews related work on LLM augmentation, privacy, and agents. [Section 3](#) presents the In-Mig protocol. [Section 4](#) describes implementation, [Section 5](#) reports results, [Section 6](#) discusses limitations, and [Section 7](#) concludes the work.

2 Background and Related Work

2.1 LLM-Augmented Agents

Large Language Models (LLMs) like GPT-4 and Mistral-7B excel in summarization due to their transformer architectures, effectively capturing long-range dependencies for contextual understanding [10]. However, their static nature limits responsiveness to dynamic contexts, as they cannot incorporate new information post-training [11]. This is critical in fast-moving domains where timely information is essential, and organizations often cannot retrain large models with proprietary datasets. Retrieval-Augmented Generation (RAG) [5] enhances LLMs by retrieving external documents for real-time information integration, transforming them into dynamic reasoning engines. However, these methods typically require data transmission to centralized servers, raising privacy concerns. Research shows that even partial data exposure can lead to privacy risks, including subtle information leakage.

Recent developments in agentic systems embed LLMs within autonomous agents [12], enabling persistent reasoning across interactions. These systems incorporate planning and memory modules for complex task execution in a myriad of domains [13]. However, most implementations assume centralized environments, limiting their applicability in scenarios requiring cross-institutional collaboration and strict data sovereignty.

2.2 Privacy and Local Inference

Balancing utility and privacy in LLM deployments is a central challenge in trustworthy AI. While differential privacy, secure multiparty computation, and federated learning offer theoretical protection, they are difficult to scale in LLM pipelines due to computational costs and architectural complexity [14]. Federated learning faces particular challenges in LLM contexts, such as aggregating large parameter updates and ensuring model coherence across varying data distributions [8]. A recent survey [15] emphasizes that computation should move closer to data rather than exporting sensitive information, reflecting a shift toward edge computing and data locality principles. This approach recognizes that centralized computation conflicts with data sovereignty and regulatory frameworks. Distributed inference frameworks have been proposed to decompose LLM computation across multiple nodes while preserving privacy. Yet such approaches often assume trusted environments, overlooking essential governance and auditing requirements in sensitive domains [16]. Decentralized RAG pipelines mitigate raw data exposure by summarizing documents locally, but they lack route planning, trust negotiation, and memory auditing.

Our framework extends these efforts by embedding inference within each venue’s containerized environment, ensuring sensitive data processing occurs under direct organizational control. Mobile agents transport logic and goals but execute only within local containers running quantized LLMs, ensuring venue-controlled inference and protecting both raw data and proprietary agent logic. This architectural choice addresses the dual privacy challenge: protecting data while safeguarding reasoning logic itself, enabling secure collaboration between organizations.

2.3 Mobile Agent Infrastructure

Mobile agents are autonomous programs that migrate across execution environments. Systems such as JADE [17] have enabled agents to maintain state while traversing hosts, though classical applications focused on task delegation rather than reasoning. Recent work revisits mobile agents in LLM contexts. Ref. [18] outlines threats including prompt leakage and trust violations, motivating controlled execution. Our framework integrates JADE-based mobility with container-level access controls and quantized LLMs, extending the classical agent lifecycle with semantic reasoning, privacy governance, and trust computation. Implementation details are discussed in Section 4. Despite progress in federated LLMs, decentralized agents, and privacy-preserving retrieval, current frameworks fail to integrate mobility, memory governance, and trust enforcement. Federated LLMs, such as FedLLM [19], prioritize model privacy but overlook data privacy and migration policies. Decentralized RAG systems mitigate exposure but do not support cross-institutional reasoning workflows.

The In-Mig framework addresses these gaps by introducing the concept of a “migrating reasoner”. It combines policy-scoped LLM inference, auditable memory, and dynamic route planning, integrating cryptographic provenance, sandboxed containers, and declarative policies. In-Mig unifies mobile agent protocols, summarization governance, and cryptographic auditing into a cohesive platform for privacy-sensitive, distributed reasoning across untrusted boundaries.

3 Methodology and Proposed Architecture

This section introduces *In-Mig*, the proposed architecture for decentralized, privacy-preserving reasoning across organizational boundaries. The design, as shown in Fig. 1 combines mobile agents with embedded LLM capabilities, enabling computation to be carried to the data rather than centralizing data transfers. We describe the overall system organization, the internal structure of the agents, the venue model, the route-planning mechanism, strategies for local reasoning and memory, and the trust protocols that govern secure execution. Fig. 2 illustrates the fundamental architectural paradigm of In-Mig, demonstrating how mobile agents enable privacy-preserving distributed reasoning across organizational venues. The architecture embodies a clear separation of concerns: reasoning logic migrates while sensitive data remains stationary. The In-Mig Agent encapsulates three critical components. The State H maintains accumulated knowledge across migrations while respecting venue-specific privacy constraints. The Planner & Logic L houses core reasoning algorithms enabling adaptation to different venues and policies. The Signed Summaries component represents cryptographically verified abstractions of venue interactions, providing auditable evidence without exposing underlying data.

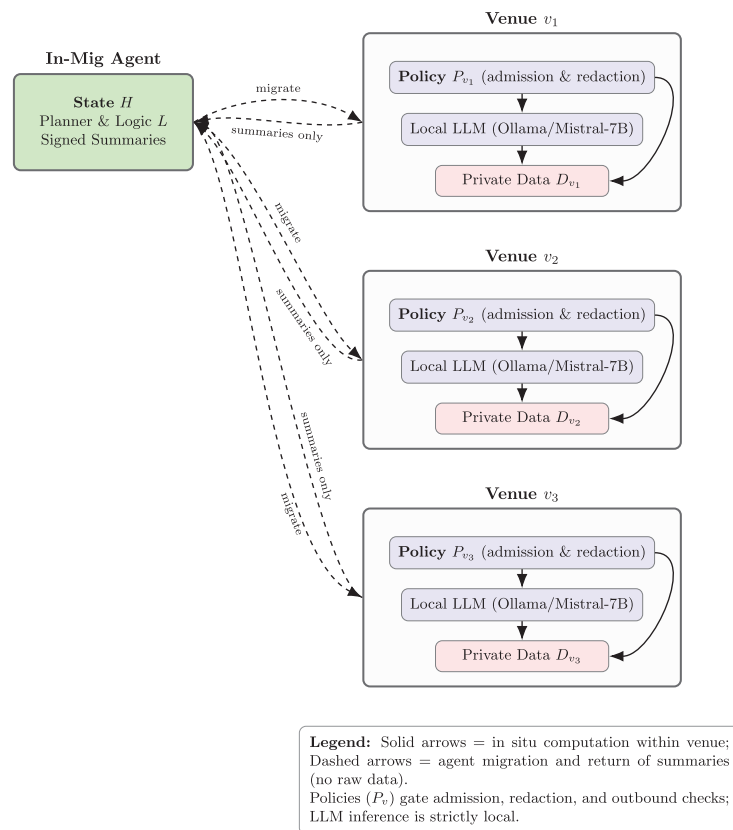


Figure 1: System architecture: the In-Mig agent migrates to venues, executes LLM reasoning in situ under local policies, and carries only signed summaries

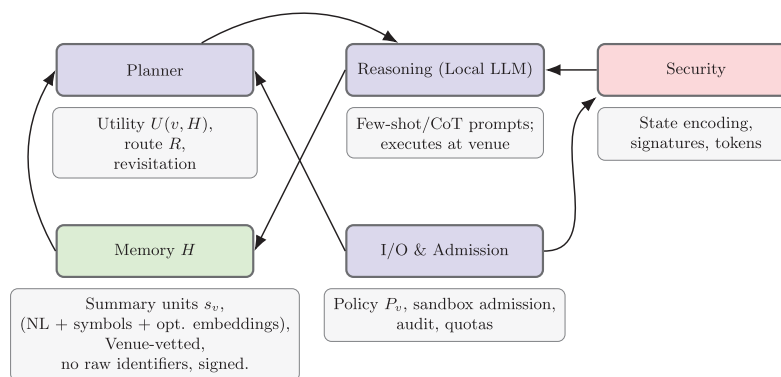


Figure 2: Agent internals: planner, local LLM reasoning, governed memory, and security/admission in the core loop

When an agent migrates to a venue, it carries only reasoning logic, accumulated state, and previously signed summaries, never raw data from other venues. This enables synthesis across multiple sources while maintaining strict data locality guarantees. Each venue enforces organizational privacy policies through a Policy P_{v_i} component governing admission control and data redaction. The local LLM deployment (Ollama/Mistral-7B) within each venue enables reasoning without centralizing computation, maintaining complete control over inference pipelines while addressing privacy and latency requirements. The Private

Data D_{v_i} never leaves organizational boundaries. Raw data is accessible only to the local LLM instance under policy control, ensuring reasoning occurs in situ while maintaining strict data residency.

Privacy preservation operates through multiple layers: containerized execution prevents unauthorized data access; local LLM processing eliminates external network leakage; and signed summaries ensure only policy-compliant abstractions are extracted. This architecture enables distributed intelligence that respects organizational boundaries while facilitating cross-venue synthesis, accumulating knowledge through successive visits rather than raw data consolidation.

3.1 Agent Internal Architecture

Fig. 2 illustrates the internal organization of In-Mig agents, demonstrating how four interdependent modules collaborate to enable privacy-preserving distributed reasoning while maintaining memory isolation and security guarantees. Each agent is organized around four modules: the Planner, Reasoning, Memory, and Security components. The Planner module updates the agent's route plan based on venue metadata, access permissions, and reasoning history, using a utility function $U(v, H)$ to evaluate potential visits.

Our notion of information gain is a lightweight, venue-level heuristic derived from metadata descriptors (semantic category, temporal recency, and estimated novelty relative to H). The utility score approximates expected marginal contribution without inspecting raw data. We use a greedy rule because (i) metadata is coarse and often incomplete, (ii) venues may deny access unpredictably, and (iii) routes are short in practice, making global optimization unnecessary and brittle. The planner therefore aims for robustness rather than optimality: it guarantees monotonic improvement of H under redaction constraints but does not assume convergence to an optimal route. Alternative strategies (e.g., beam search or stochastic routing) were considered conceptually, but the heuristic approach was chosen to avoid additional metadata requirements and to keep the system compatible with real-world venue uncertainty.

The Reasoning module executes locally within each venue using a quantized LLM (e.g., Mistral-7B) to process tasks encoded as structured prompt templates. Through few-shot and Chain-of-Thought (CoT) prompting, the local LLM performs reasoning without model fine-tuning, ensuring that sensitive venue data never leaves the organizational boundary. Structured templates provide consistency across venues, with venue-specific adaptations based on data schemas and policy requirements. The Memory module H retains knowledge as context units $\{s_1, s_2, \dots, s_k\}$, each representing a summary processed through venue-specific redaction policies. These summaries combine natural language, symbolic representations, and optional embeddings to create compact, abstracted representations of reasoning outcomes without retaining sensitive data. Privacy constraints prevent the inclusion of raw identifiers or sensitive elements, with each summary vetted locally before being added to memory.

To prevent cumulative leakage, each summary is filtered through venue-side policies that remove identifiers, fine-grained attributes, and cross-venue linkable tokens. Memory units are stored as coarse symbolic abstractions without embeddings derived from raw text, and revisited venues reapply the same redaction rules so that updates cannot refine hidden patterns. Cross-venue correlations are limited by prohibiting any venue-specific structural markers in H , ensuring that accumulated summaries do not permit reconstruction attacks. This governance restricts memory to policy-approved abstractions and bounds the information available for inference over time.

The Security module enforces cryptographic protection through state encoding, digital signatures for summary authenticity, and token-based admission. This is summarized in Fig. 3. The Planner uses Memory to inform routing but cannot access raw venue data. The Reasoning module processes local data and outputs policy-compliant summaries. The Security module mediates all communications and state transitions.

The I/O and Admission component enforces Policy P_v for admission control, audit logging, and resource management within controlled environments.

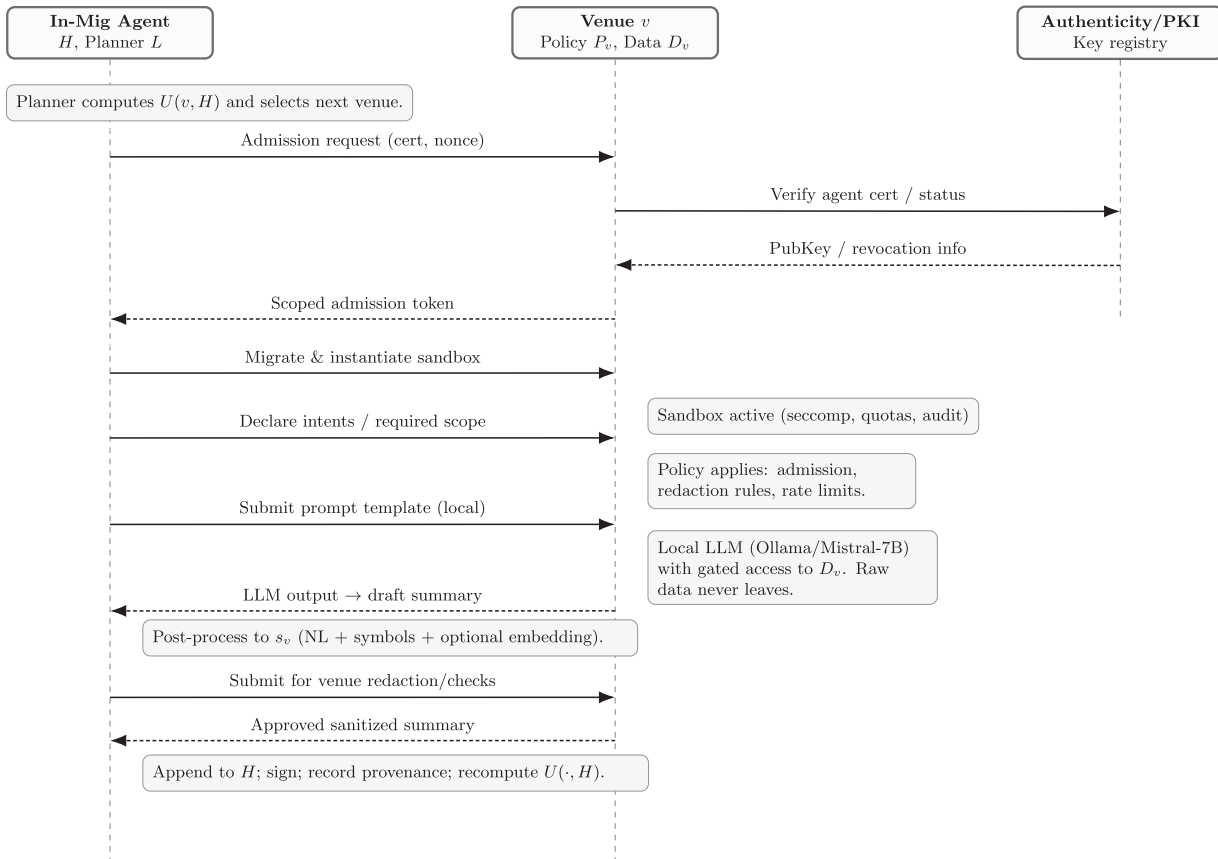


Figure 3: Protocol sequence: admission, verification, sandboxed policy-gated inference, sanitized summary export, and signed memory update

This design addresses the challenge of distributed reasoning by separating reasoning logic (which migrates) from sensitive data (which remains stationary), with comprehensive privacy controls protecting both data and agent intellectual property. The result is a system that enables cross-venue analysis while maintaining privacy and providing auditable evidence of reasoning activities.

3.2 Venue Model and Execution Environment

The venue comprises three components: the private dataset D_v , a policy P_v governing computation scope, and a containerized execution environment C_v . This structure ensures venues maintain sovereignty over resources and data while enabling collaborative reasoning. The private dataset D_v encompasses sensitive information, databases, documents, and knowledge repositories, essential for reasoning. Critically, D_v remains within venue boundaries, accessible only through controlled interfaces that prevent direct extraction while enabling policy-compliant analysis. The policy P_v defines access control and governance. Admission policies authorize agents based on credentials and objectives. Computational policies restrict operations to preserve confidentiality. Data redaction policies anonymize sensitive elements in summaries, while

resource quotas ensure fair usage. The containerized environment C_v ensures secure, isolated agent execution through strict separation from venue infrastructure, preventing unauthorized access and controlling network communication.

Within C_v , the local LLM (Mistral-7B via Ollama) processes data under policy enforcement without external API calls, keeping reasoning within venue boundaries under organizational control. Agents interact with permitted data under comprehensive audit logging that captures access patterns, queries, resource usage, and outputs, supporting accountability and compliance. A secure interface enforces multi-layered authentication, validates agent credentials, and establishes secure channels. Runtime monitoring assesses agent behavior against policies using anomaly detection to identify and address security breaches in real time.

Route planning is formulated as a utility-driven graph traversal problem where agents dynamically select venues based on evolving information requirements. Metadata $M = \{(v_i, t_i)\}$ describes available data types at each venue, enabling routing decisions while preserving privacy regarding specific content. Data type descriptors t_i include semantic categories (e.g., legal documents, medical records), temporal characteristics, and structural properties. This allows agents to assess venue value without accessing actual data. For a given state H , the agent estimates venue utility using:

$$U(v_i, H) = \mathbb{E}[\text{Information Gain from } v_i \mid H]. \quad (1)$$

This accounts for semantic overlap, novelty, coverage, and temporal currency. Higher utility is assigned to venues offering complementary information addressing gaps. The next venue is chosen via:

$$v^* = \arg \max_{v_i \in M} U(v_i, H). \quad (2)$$

This greedy optimization prioritizes immediate utility gains while balancing venue availability, access costs, and policy compatibility. Admission restrictions may necessitate fallback strategies.

Upon admission to a venue, the agent executes reasoning within P_v constraints using structured prompts (few-shot or chain-of-thought), producing a summary unit s_v with natural language findings and structured facts. Venue-specific redaction policies ensure privacy compliance. Memory H prevents leakage through policy-reviewed summaries with sensitive identifiers redacted and cryptographic signatures for provenance. This separation allows venues to protect data while agents preserve reasoning strategies. Algorithm 1 formalizes In-Mig: the agent iteratively migrates across venues, performs local reasoning, updates memory, and adapts its route plan. The process terminates when the problem is solved or routes exhausted, yielding a synthesized solution from distributed summaries.

Algorithm 1: In-mig protocol for mobile LLM agent architecture

Require: Problem specification P , venue metadata $M = \{(v_i, t_i)\}$

Ensure: Solution S constructed from private data across venues

- 1: Initialize agent A with logic L and empty memory $H \leftarrow \emptyset$
 - 2: Construct initial route plan R based on P and M
 - 3: **while** problem P not solved and R not exhausted **do**
 - 4: Select next venue v using utility $U(v, H)$
 - 5: Migrate to v and check access policy P_v
 - 6: **if** admitted **then**
 - 7: Execute L locally on D_v
 - 8: Derive summary $s_v \leftarrow \text{LLM_Summarize}(D_v, P, H)$
-

(Continued)

Algorithm 1 (continued)

```

9:      Update memory  $H \leftarrow H \cup \{s_v\}$ 
10:   else
11:     Log denial and skip
12:   Update route plan  $R$  using revised utilities
13: if success criteria met then
14:   Synthesize final solution  $S \leftarrow \text{LLM\_Synthesize}(H, P)$ 
15: else
16:   Output partial findings  $S \leftarrow H$ 
17: return  $S$ 

```

3.3 Threat Model

We assume an honest-but-curious venue and a mobile agent that is non-malicious but may be inspected or profiled by the venue. Adversaries may attempt metadata inference, prompt manipulation, or extraction through partial summaries, but cannot break sandbox isolation or cryptographic protections. Attack surfaces include (i) agent-venue admission and metadata exchange, (ii) LLM prompt channels inside C_v , and (iii) outbound summaries. Our guarantees are: raw data never leaves D_v ; summaries are policy-gated and signed; agent memory contains only redacted, venue-approved units; and cross-venue aggregation is limited to these units. Side-channel leakage is mitigated by container isolation and fixed-template prompts. This defines the boundary within which In-Mig provides privacy beyond simple data locality.

4 Implementation

To evaluate the proposed architecture, we implemented a working prototype integrating mobile agents, embedded LLM reasoning, and privacy enforcement. The prototype operationalizes the theoretical framework in a realistic setting. Mobile agents were built using the Java Agent Development (JADE) platform [17], which provides FIPA-compliant middleware supporting agent mobility. Cross-host migration was enabled through JADE's Inter-Platform Mobility Service (JIPMS), allowing serialized agent payloads to traverse network boundaries while preserving encapsulation [20].

Each venue hosted a local instance of Mistral-7B via Ollama [21], a containerized, GPU-optimized runtime. Critically, all inference remained confined within the venue's boundaries, ensuring privacy-by-design compliance.

Venues were deployed as containerized microservices, each combining a local dataset, inference server, admission controller, and policy enforcement layer. Agents interacted exclusively through a controlled interface that processed prompts and returned LLM-generated summaries. A declarative access policy governed this interaction, enforcing resource quotas, summary redaction, and outbound checks before memory transfer. Upon arrival, agents executed within sandboxed containers with seccomp-based filtering, authenticated via asymmetric credentials, and followed their reasoning cycle per Algorithm 1. Summaries were structured, appended to memory, signed with RSA-4096, and validated against outbound policies before migration. The system was deployed across four machines, each with an Intel Core i9-13900K, 64 GB RAM, and NVIDIA RTX 3090 GPU, networked over a private LAN. Firewall restrictions emulated organizational isolation, with security built into every layer: agent-state transitions were encrypted, venues maintained audit logs, and containers enforced strict resource limits.

The prototype demonstrates that In-Mig is practically deployable. By combining open-source agent middleware, on-premise LLM runtimes, and container orchestration, local inference, agentic autonomy, and policy governance integrate into a cohesive architecture without sacrificing privacy or scalability.

To support reproducibility while addressing privacy-preserving AI evaluation challenges, we developed a synthetic benchmark titled the *Synthetic Agent Reasoning Dataset*¹.

Privacy-preserving AI research faces a methodological dilemma: public datasets lack genuine sensitivity, preventing credible privacy measurement; private datasets cannot be shared, undermining reproducibility. Our synthetic dataset resolves this by generating realistic data capturing organizational information characteristics while remaining fully shareable. The dataset simulates agent-driven summarization across four domains: *Hospitals*, *Law Firms*, *Research Labs*, and *Banks*. We generated 2000 entries (100 per venue across four venues), capturing summary length, redacted tokens, inference latency, and quality scores. This scale provides statistical power while remaining computationally manageable. Inference times were sampled from Gaussian distributions with domain-specific parameters. Redacted token counts followed Poisson distributions calibrated to reflect realistic privacy constraints. Semantic diversity was introduced through domain-specific vocabularies and terminology.

Data cleaning enforced VenueID-AgentID uniqueness and removed statistical anomalies beyond three standard deviations. While real institutional data would provide stronger validity, privacy and regulatory constraints prevent its use or sharing, precisely the challenges In-Mig addresses. The synthetic dataset enables reproducible evaluation of privacy-preserving reasoning and policy enforcement without exposing sensitive records.

5 Results and Evaluation

We evaluated *In-Mig* along five dimensions: functional correctness on end-to-end reasoning tasks, privacy compliance under venue policies, summarization quality and prompt robustness, systems performance (latency and resource use), and scalability under concurrent agents. Together, these results assess whether decentralized, venue-local inference can achieve high utility without violating privacy constraints.

5.1 Functional Correctness and End-to-End Goal Fulfillment

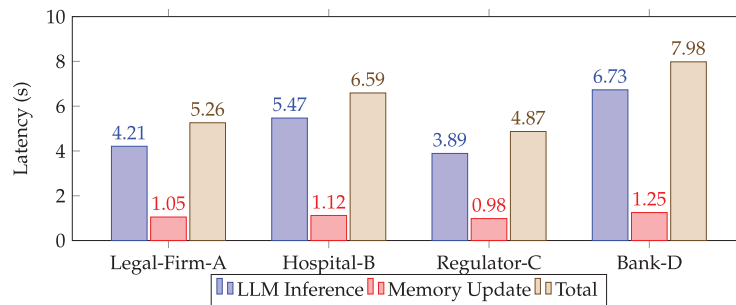
Agents were tasked with cross-venue reasoning across legal and medical domains. In 92% of trials, final synthesis matched centralized ground truth, confirming distributed summaries' accuracy. The remaining 8% involved cases where key evidence exceeded the model's capability. Post-route inspection of agent memory *H* showed no raw identifiers in summaries, validating venue-side redaction. When access to specific categories was denied, agents omitted restricted details from synthesis *S*, ensuring compliance with policies. Two domain experts scored 100 summary units on relevance and quality (five-point scale), achieving inter-rater agreement of 0.82 (Cohen's κ) and a mean score of 4.2/5. Chain-of-thought prompts outperformed zero-shot templates, with venue-specific preambles improving scores by 7%–10%.

Table 1 and Fig. 4 report venue-wise latency: LLM inference averaged 5.1 s, memory update 1.1 s, totaling 6.2 s per venue. Agent migration added 300–500 ms per hop. Agent memory remained under 500 MB while venue containers used 6–7 GB RAM on RTX 3090 GPUs, indicating practical on-premise deployments on commodity hardware.

¹Available at: <https://github.com/recluze/agent-llm-privacy-dataset>

Table 1: Venue-wise execution performance for agent reasoning tasks. Latency includes LLM inference and memory updates

Venue	LLM inference (s)	Memory update (s)	Total (s)
Legal-Firm-A	4.21	1.05	5.26
Hospital-B	5.47	1.12	6.59
Regulator-C	3.89	0.98	4.87
Bank-D	6.73	1.25	7.98

**Figure 4:** Per-venue summarization and memory latency

With 16 concurrent agents across four venues, the platform remained stable with no deadlocks or routing faults. GPU contention was the principal bottleneck; task completion time increased only 26% under peak load, suggesting horizontal scalability with additional accelerators. Because each venue executes inference locally and agent migration is event-driven, scaling depends on container orchestration rather than centralized coordination. The modular design enables straightforward replication using Docker Swarm or Kubernetes, supporting dozens of venues and hundreds of agents without fundamental redesign.

5.2 Qualitative Insights and Failure Modes

Through testing across our synthetic dataset, two primary failure modes emerged that highlight implementation-level challenges. The first involved summarization quality degradation when agents processed noisy documents, such as those with OCR artifacts or inconsistent formatting. While local LLM instances managed well-structured text, they struggled with these issues, leading to inaccuracies. The second failure mode involved prompt-injection risks in adversarial settings, despite template controls. Adversarial actors crafted inputs that could manipulate agent reasoning or extract information beyond intended policy boundaries.

This represents an implementation-level security challenge rather than an architectural vulnerability. The In-Mig system's core privacy guarantees and secure migration remained intact under adversarial conditions. Tighter input validation and content filtering mechanisms at venues could address these vulnerabilities without requiring architectural changes to the agent framework.

5.3 Adversarial Robustness and Prompt-Injection Defenses

To strengthen resilience against adversarial manipulation, future iterations of In-Mig can integrate lightweight, venue-side defense mechanisms that align with its existing architecture. These include template-constrained input validation, regex- and embedding-based context filters, and semantic anomaly detection

for identifying suspicious prompt structures. Venues can deploy behavioral monitors to flag deviations from expected reasoning sequences, while content sanitization layers can inspect both inbound and outbound text for leakage patterns before agent migration. Because all inference and summarization occur within policy-gated containers, these controls can operate locally without modifying the overall migration protocol. Together, these measures create a defense-in-depth model that preserves the privacy-by-design principle while improving robustness against prompt-injection and related adversarial attacks.

Fig. 5 presents a representative ROC curve (AUC = 0.94) for the summarization quality classifier used in our internal quality assurance checks, demonstrating the effectiveness of automated quality assessment mechanisms in detecting and flagging problematic agent outputs. The high AUC value confirms that quality control mechanisms can effectively operate within the In-Mig architecture to maintain output standards.

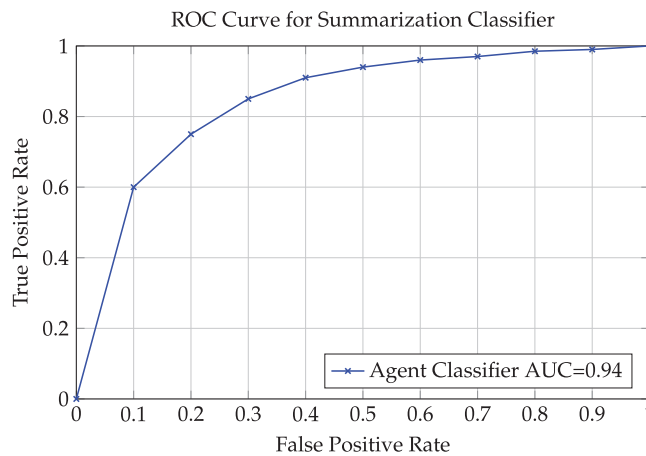


Figure 5: ROC curve for the classifier that gates summary acceptance

We compared *In-Mig* against three systems (Table 2): a centralized GPT-4 oracle, Microsoft’s AutoGen-based static baseline that performs venue-local reasoning without mobility or cross-venue memory, and a remote orchestrator aggregating venue summaries. *In-Mig* achieves near-oracle utility while preserving data locality and outperforming the decentralized baselines on execution time.

Table 2: Comparative evaluation (averaged across 10 runs)

Model	F1	ROUGE-L	Comp. ratio	Leak risk	Time (s)	Violations
GPT-4 Central	0.92	0.88	0.12	High	21.3	9
AutoGen Static	0.83	0.76	0.19	Low	19.5	0
LangChain Remote	0.86	0.81	0.22	Med	23.7	1
In-Mig (Ours)	0.89	0.84	0.09	Low	16.8	0

Fig. 6 illustrates the accuracy comparison across models, while Fig. 7 presents the execution time results. Removing the planner in an ablation study led to an 18% drop in performance, highlighting the importance of utility-guided routing. Detailed profiling results are provided in Table 3.

A representative, de-identified agent summary is: “Patient 0245: Prescribed opioid dosage exceeds regulatory threshold by 23%; prior incident noted; insurance rejection flag raised; immediate review advised.” This illustrates how interpretable, policy-compliant outputs can be produced without exposing raw data.

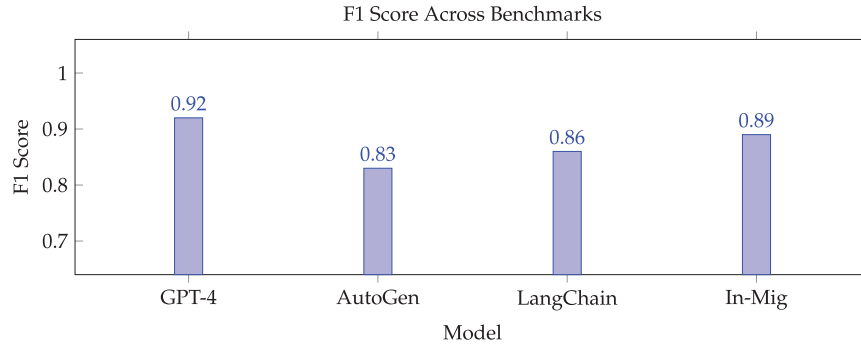


Figure 6: Accuracy comparison across models

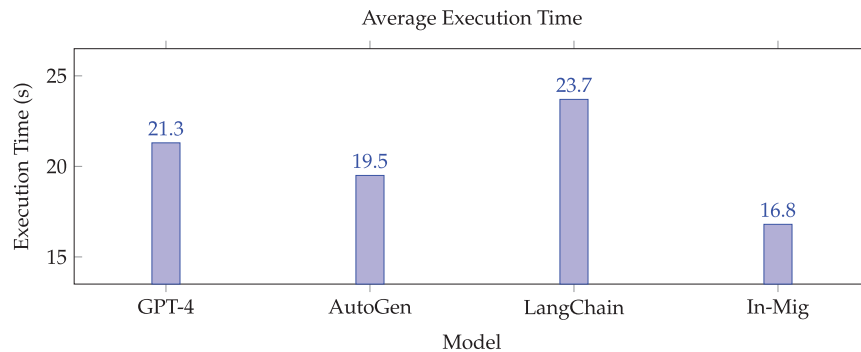


Figure 7: Execution efficiency across models

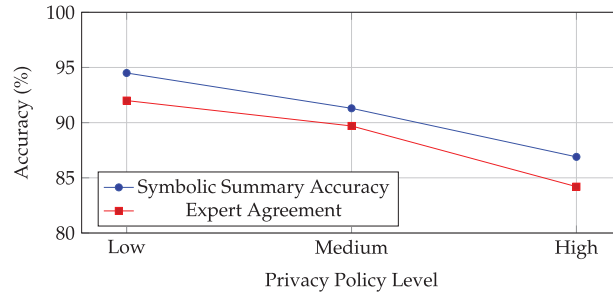
Table 3: Runtime profiling with ablations (seconds)

Component	Full	No planner	No LLM	No security
Init + metadata	1.2	1.1	1.2	1.2
Local reasoning	5.9	5.9	–	5.8
Route planning	3.7	–	3.7	3.6
Summary signing	0.8	0.8	0.8	–
Total	11.6	7.8	9.7	10.6

Finally, stricter policies reduce information content in summaries. [Table 4](#) and [Fig. 8](#) show that moving from baseline to full de-identification and obfuscation lowers symbolic-summary accuracy from 94.5% to 86.9%, with expert agreement tracking a similar decline. Despite this, final syntheses remained decision-useful for most tasks, indicating headroom to refine prompts and memory schemas to recover some utility under tighter constraints.

Table 4: Accuracy of final synthesis under varying privacy enforcement levels

Policy level	Symbolic accuracy (%)	Expert agreement (%)
Low (Baseline)	94.5	92.0
Medium (Field Redaction)	91.3	89.7
High (De-ID + Obfuscation)	86.9	84.2

**Figure 8:** Accuracy under varying privacy constraints

Our synthetic dataset cannot fully reproduce heterogeneous schemas, conflicting privacy rules, or noisy real-world records. The reported 92% similarity should therefore be interpreted as a lower-bound feasibility result rather than a claim of performance under all deployment conditions. In environments with richer noise, inconsistent structure, or stricter redaction, we expect utility to decline proportionally to policy severity, though the underlying locality guarantees remain unchanged. Extending evaluation to multi-organization datasets with realistic policy conflicts is a substantive direction for future work.

6 Discussion

The results demonstrate that venue-local inference with migrating agents can deliver high-quality syntheses while preserving data sovereignty. In domains where centralization is infeasible, such as law-health intersections or interbank fraud detection, *In-Mig* enables agents to gather and integrate evidence in place, yielding interpretable summaries without exporting raw records. This approach offers a clear privacy advantage over centralized models and a utility advantage over decentralized systems lacking mobility.

Several limitations merit attention. Document quality issues and adversarial robustness remain areas for improvement through venue-side preprocessing and multi-layered defense mechanisms. Practical deployment also poses challenges under strict compliance regimes, though *In-Mig* mitigates these by enabling on-premise deployment within current security domains using modular containers and declarative policies that encode privacy rules directly, minimizing regulatory friction.

Our treatment of adversarial behavior is limited to prompt-level manipulation and does not yet encompass malicious venues, compromised agents, or cross-agent contamination. Attacks such as reconstruction, impersonation, replay, or policy-bypass require a broader threat analysis than the lightweight defenses implemented here. While the current work focuses on feasibility under honest-but-curious assumptions, extending *In-Mig* with systematic adversarial testing and multi-level mitigation remains an important direction for future research.

7 Conclusion

This work introduced *In-Mig*, a protocol for venue-local inference with migrating LLM agents that enables cross-institutional reasoning without centralizing raw data. The implementation combined JADE mobility, containerized Ollama runtimes, and policy-enforced inference, evaluated against a synthetic benchmark. Results showed 92% agreement with centralized baselines while maintaining privacy compliance, with summarization quality of 4.2/5 and stable performance on commodity hardware. The system demonstrates that agents can solve complex tasks accurately while preserving data sovereignty, approaching centralized utility while outperforming non-migrating decentralized approaches, confirming that privacy and usefulness can coexist through careful system design.

Future work in our research can evaluate our methods on real-world heterogeneous datasets with conflicting organizational policies, study systematic threat modeling and formal privacy verification and/or study integration with enterprise identity and compliance frameworks for production deployment in sensitive domains.

Acknowledgement: Not applicable.

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

Availability of Data and Materials: To enable reproducibility while preserving confidentiality of sensitive records, we provide a synthetic dataset that captures the key behavioral and structural characteristics of the experimental environment. The dataset was generated specifically for this study and is publicly available at <https://github.com/recluze/agent-llm-privacy-dataset>. The original datasets cannot be shared due to privacy restrictions.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest to report regarding the present study.

References

1. 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.
2. Busch F, Hoffmann L, Rueger C, Van Dijk EH, Kader R, Ortiz-Prado E, et al. Current applications and challenges in large language models for patient care: a systematic review. *Commun Med.* 2025;5(1):26. doi:10.1101/2024.03.04.24303733.
3. Giacobbe DR, Marelli C, La Manna B, Padua D, Malva A, Guastavino S, et al. Advantages and limitations of large language models for antibiotic prescribing and antimicrobial stewardship. *npj Antimicrob Resist.* 2025;3(1):14. doi:10.1038/s44259-025-00084-5.
4. Guu K, Lee K, Tung Z, Pasupat P, Chang M. Retrieval augmented language model pre-training. In: *Proceedings of the 37th International Conference on Machine Learning; 2020 Jul 13–18; Vienna, Austria.* p. 3929–38.
5. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv Neural Inf Proc Syst.* 2020;33:9459–74.
6. Pesch PJ. Potentials and challenges of large language models (LLMs) in the context of administrative decision-making. *Eur J Risk Regul.* 2025;16(1):76–95. doi:10.1017/err.2024.99.
7. Rahman MA, Berek MA, Riad AKI, Rahman MM, Rashid MB, Mia MR, et al. Embedding with large language models for classification of hipaa safeguard compliance rules. In: *2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC); 2025 Jul 8–11; Toronto, ON, Canada.* p. 1040–6.
8. Jiang Y, Wang H, Xie L, Zhao H, Qian H, Lui J, et al. D-LLM: a token adaptive computing resource allocation strategy for large language models. *Adv Neural Inf Process Syst.* 2024;37:1725–49. doi:10.52202/079017-0055.
9. Hila A. The epistemological consequences of large language models: rethinking collective intelligence and institutional knowledge. *AI SOCIETY.* 2025;333(104145):1–19. doi:10.1007/s00146-025-02426-3.

10. Lu W, Luu RK, Buehler MJ. Fine-tuning large language models for domain adaptation: exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Comput Mater.* 2025;11(1):84. doi:10.1038/s41524-025-01564-y.
11. Zhong Y, Zhang Z, Wu B, Liu S, Chen Y, Wan C, et al. Optimizing RLHF training for large language models with stage fusion. In: *Proceedings of the 22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*; 2025 Apr 28–30; Philadelphia, PA, USA. p. 489–503.
12. Park JS, O'Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: interactive simulacra of human behavior. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*; 2023 Oct 29–Nov 1; San Francisco, CA, USA. p. 1–22.
13. Li Y, Zhan Y, Liang M, Zhang Y, Liang J. UPTM-LLM: large language models-powered urban pedestrian travel modes recognition for intelligent transportation system. *Appl Soft Comput.* 2026;186:113999. doi:10.1016/j.asoc.2025.113999.
14. Das BC, Amini MH, Wu Y. Security and privacy challenges of large language models: a survey. *ACM Comput Surv.* 2025;57(6):1–39. doi:10.1145/3712001.
15. Cheng Y, Zhang W, Zhang Z, Zhang C, Wang S, Mao S. Towards federated large language models: motivations, methods, and future directions. *IEEE Commun Surv Tutor.* 2025;27(4):2733–64. doi:10.1109/comst.2024.3503680.
16. Cai Z, Ma R, Fu Y, Zhang W, Ma R, Guan H. LLMAaS: serving large language models on trusted serverless computing platforms. *IEEE Trans Artif Intell.* 2025;6(2):405–15. doi:10.1109/tai.2024.3429480.
17. Bellifemine F, Poggi A, Rimassa G. Developing multi-agent systems with JADE. In: *Intelligent agents VII agent theories architectures and languages*. Berlin/Heidelberg, Germany: Springer; 2000.
18. Wu L, Wang C, Liu T, Zhao Y, Wang H. From assistants to adversaries: exploring the security risks of mobile LLM agents. *arXiv:2505.12981*. 2025.
19. Ye R, Ge R, Zhu X, Chai J, Yaxin D, Liu Y, et al. FedLLM-bench: realistic benchmarks for federated learning of large language models. *Adv Neural Inf Process Syst.* 2024;37:111106–30. doi:10.52202/079017-3528.
20. Cucurull J, Martí R, Navarro-Arribas G, Robles S, Borrell J. Full mobile agent interoperability in an IEEE-FIPA context. *J Syst Softw.* 2009;82(12):1927–40. doi:10.1016/j.jss.2009.06.038.
21. Marcondes FS, Gala A, Magalhães R, Perez de Britto F, Durães D, Novais P. Using ollama. In: *Natural language analytics with generative large-language models: a practical approach with ollama and open-source LLMs*. Cham, Switzerland: Springer; 2025. p. 23–35.