JAMES COOK UNIVERSITY AUSTRALIA

ACADEMY

INDIVIDUAL TASK COVER SHEET

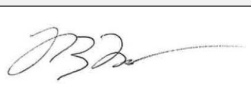
*This cover sheet should be signed and dated by both the **student and their Primary Advisor**, and attached to the front of the Research Dissertation assessment task before submission.*

***NB:** Students should ensure sufficient time for their advisors to review and approve by signature the final version, prior to submission.*

SUBJECT CODE	**RM8502**							
STUDY PERIOD - YEAR	**S2**				**2025**			
SUBJECT TITLE	Research Project							
ASSESSMENT TITLE	**RESEARCH DISSERTATION**							
DUE DATE	3/11/2025							
STUDENT FAMILY NAME	Ross							
STUDENT GIVEN NAME	James							
JCU STUDENT NUMBER	1	4	4	7	2	2	6	6
PRIMARY ADVISOR NAME	Prof Ickjai Lee							

Student Declaration

In preparing this work, I have adhered to the JCU Learning, Teaching and Assessment policies.

Student signature		**Date**	2/11/2025

Advisor Declaration

I have reviewed and approved this work to be submitted for assessment.

Advisor signature		**Date**	30/10/2025

Decentralised Ledger Architecture for Trustworthy Multi-Agent Memory

RM8502 – S2 – Research Dissertation

James Ross (14472266)

Abstract

Autonomous agents often lack reliable shared memory, which hampers effective collaboration. This research introduces a blockchain-inspired, decentralised ledger to fill this gap, focusing on a design that uses cryptographic signatures to verify data provenance. The work provides a quantitative proof of concept by experimentally assessing the architecture's feasibility and evaluating key performance metrics, including transaction throughput, verification latency, and economic cost, across different data loads in a controlled local EVM environment.

The results show a perfect 100% Provenance Verification Accuracy and a predictable linear increase in gas costs. More importantly, the findings highlight a fundamental performance trade-off: on-chain throughput decreased from around 20 to 10 transactions per second, while off-chain verification remained extremely fast and stable at under 3 milliseconds. This demonstrates that the architecture sacrifices write speed for near-instant trust, confirming its potential for high-value, low-frequency data applications where data integrity is critical. The study concludes by identifying the on-chain bottleneck as the key limitation and suggests that future research into Layer 2 scaling solutions is the most promising way to improve performance and expand the architecture's applicability for collaborative AI.

1. Background and Significance of the Research

1.1. The Emergence of Autonomous Agents and the Centrality of Memory

The emergence of the transformer architecture (Vaswani et al., 2017) marked a crucial milestone, sparking the large-language-model (LLM) revolution. By replacing recurrent computation with self-attention mechanisms, transformers unlocked unprecedented parallelism at scale, enabling models to capture long-range dependencies with fewer inductive biases. As researchers gradually increased the number of model layers, expanded token contexts, and used web-scale corpora for training, these models evolved into increasingly capable LLMs that exhibited emergent abilities such as few-shot generalisation and chain-of-thought reasoning. These advanced capabilities prompted a further leap: the development of LLM-powered agents. Such agents can interpret prompts as goals, independently develop plans, interact with external APIs, critically assess their outputs, and refine their responses in real-time. This evolution transforms the once-passive language model into a versatile, goal-

oriented collaborator suitable for a wide range of fields, from software development to scientific discovery (Wiesinger et al., 2024).

A key architectural element driving this evolution is the incorporation of external memory systems. As shown by Figure 1, a general agent framework embeds external memory within the orchestration layer. This memory persistence goes beyond the model's native, often temporary, context window, enabling deeper reasoning, multi-step planning, and learning across extended interactions. As a result, the agent transforms from a stateless conversational tool into a more competent, adaptable, and goal-oriented entity capable of learning and modifying its behaviour over time. It is this external memory, especially its role in multi-agent settings, that is the main focus of this research.

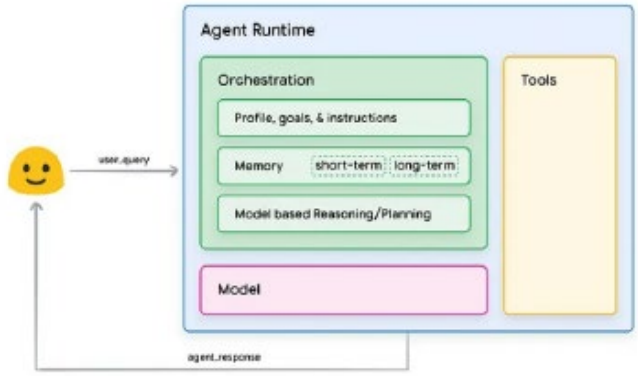


Figure 1: General Agent Architecture and Components adapted from Wiesinger et al. (2024).

1.2. A Survey of Dominant Single-Agent Memory Paradigms

The research landscape for single-agent memory is lively and has established several key architectural paradigms. To grasp the specific challenges in multi-agent systems, it is essential to critically analyse these core approaches and their limitations. This section reviews four main paradigms: parametric, retrieval-augmented, differentiable, and compressive memory.

1.2.1. Parametric Memory

Early LLMs relied purely on parametric memory, where knowledge is implicitly stored within the model's vast array of weights during pre-training (Petroni et al., 2019). This functions as a distributed repository of facts and patterns, providing the key advantage of quick, implicit recall during generation. However, this method has notable limitations. Parametric memory is difficult to recall rare, "long-tail" knowledge and notoriously hard to update; as Kandpal et al. (2023) highlight, knowledge encoded in weights is "neither editable nor scalable once the model is trained." Updating this fixed knowledge requires costly fine-tuning, which risks "catastrophic forgetting" of previously learned information.

When scaled to multi-agent systems, the limitations of parametric memory become quite severe. As Chen et al. (2024) point out, independently trained agents develop misaligned knowledge weights, which lead to contradictions and inconsistent world models. Directly

merging these weights isn't feasible, demanding costly co-training or distillation techniques to align them. This fundamental limitation of parametric memory, its isolated, opaque, and static nature, directly drives the need for external, explicit memory architectures, especially in collaborative multi-agent settings.

1.2.2. Retrieval-Augmented Memory

Retrieval-Augmented Generation (RAG) signals a major shift in approach, enabling large language models (LLMs) to access external sources such as vector databases and real-time retrieval systems (Lewis et al., 2020). This method directly addresses the limitations of static, unscalable parametric memory, enabling agents to access potentially unlimited, regularly updated external information. By separating the knowledge base from the model's parameters, RAG enhances information freshness and helps reduce hallucinations caused by outdated parametric knowledge (Gao et al., 2024).

However, RAG also introduces new integration challenges. Models must effectively combine retrieved evidence with their internal knowledge, and the system’s trustworthiness depends heavily on the quality of the retriever (Hagström et al., 2023). In multi-agent scenarios, RAG is effective for establishing shared factual grounding. Nevertheless, it raises important issues that single-agent architectures do not encounter. As noted in the literature review (Ross, 2025), simultaneous writes from multiple agents to a shared vector database require strong governance and concurrency controls to prevent race conditions and data corruption. Furthermore, the source of retrieved information becomes critical; an agent must trust the origin of a passage to resolve conflicts and identify potential misinformation.

1.2.3. Differentiable and Compressive Memory

Other architectures focus on memory management. Differentiable memory, inspired by concepts such as Neural Turing Machines (Graves et al., 2014), features RAM-like vector banks fully integrated into the neural network's computational graph. This enables end-to-end learning of read-write policies but faces significant challenges in terms of training stability and scalability. In a multi-agent context, using such a system as a shared "neural blackboard" encounters major hurdles, as concurrent writes risk vector interference and the credit assignment problem for shared contributions remains unresolved (Guo et al., 2024).

Compressive and priority-based memory mechanisms manage memory by compressing older information or prioritising important data for retention, inspired by human memory abstraction (Rae et al., 2019; Park et al., 2023). While essential for managing long-term memory growth for individual agents, these approaches pose coordination risks in multi-agent systems. If agents adopt different compression schemes or disagree on the importance of information, their individual memories can diverge, leading to misalignments and coordination failures due to information loss.

1.3. The Critical Research Gap in Multi-Agent Memory

Despite significant advances in single-agent architectures, their use in multi-agent settings presents inherent complexities and ongoing, under-researched challenges. As noted by a systematic literature review (Ross, 2025) and summarised in Table 1 below, the key issues of multi-agent coordination are often acknowledged but rarely become the main focus of research.

Table 1: Coverage of key technical categories in recent memory-augmented LLM studies (Ross, 2025).

#	Technical Category	Zhang etal. (2024)	Khosla etal. (2023)	Gao etal. (2024)	Guo etal. (2024)	L. Wang etal. (2024)	S. Wang etal. (2024)	Yu etal. (2025)	Manzoor etal. (2024)
1	Capacity & Scalability	✓	✓	◐	◐	✓	◐	◐	◐
2	Retrieval & Representa-tion	✓	✓	✓	✓	◐	✓	✓	◐
3	Forgetting & Memory Management	✓	◐			✓	◐		✓
4	Integration & Consistency	◐	◐			◐	◐		◐
5	Evaluation & Benchmarks	◐						✓	
6	Personalisation & Privacy	◐				◐	◐		
7	Multimodal Memory	◐	✓	◐					✓
8	Multi-Agent Coordination	◐							

Note. ✓ = fully addressed; ◐ = partially addressed; blank = not addressed.

Despite extensive research on single-agent memory, memory in multi-agent systems (MAS) is often only briefly mentioned in outlook sections or not examined in detail in existing surveys. This gap is problematic because, in practice, deployments increasingly involve MAS coordination with multiple agents needing to read, write, and reason over shared or interoperable memories. Such situations raise new and complex questions regarding concurrency, consistency, provenance, and collective forgetting that existing single-agent designs do not adequately address.

Further analysis in the literature review (Ross, 2025) of common memory architectures such as parametric, retrieval-augmented, differentiable, and compressive/priority-based highlights that, while these architectures are well-developed for single-agent use, extending them to multi-agent contexts is challenging. As shown in Table 2, specific technical challenges consistently arise when considering shared memory in MAS.

Table 2: Key challenges in Scaling LLM Memory Architectures to Multi-Agent Systems (Ross, 2025).

Memory Architecture	Key Multi-Agent Scaling Challenges
Parametric Memory	**Collaborative Memory Formation & Learning** – independent weight sets diverge and aligning them demands costly co-training, distillation, or weight-merging. **Memory Provenance & Trustworthiness** – facts embedded in weights lack explicit source tags, so agents cannot audit or trust each other's parametric claims. **Benchmarks & Evaluation Frameworks** – no standard MAS benchmark measures cross-agent consistency when every agent depends on private, frozen weights.
Retrieval-Augmented Memory	**Concurrency Control for Shared Memories** – many agents writing/reading the same vector DB require locking, versioning, or merge rules to avoid overwrite and race conditions. **Memory Provenance & Trustworthiness** – retrieved passages must carry source and confidence metadata so agents can resolve conflicts and filter misinformation. **Integrating Memory with Agent Communication & Planning** – agents must decide when to query the store versus ask peers, and continually update joint plans as shared memory evolves.
Differentiable Memory Networks	**Concurrency Control for Shared Memories** – simultaneous writes to a neural blackboard cause vector interference; needs gated or partitioned write protocols. **Credit Assignment in Multi-Agent Learning** – if a shared slot drives group success, the system must credit the writer agent to reinforce helpful contributions and discourage harmful ones. **Theoretical Foundations & Cognitive Models** – we lack formal models for capacity, stability, and convergence of a differentiable memory jointly trained by many agents.
Compressive / Priority-Based Memory	**Collaborative Memory Formation & Learning** – agents must agree on what to compress and how to merge summaries into a shared narrative without losing critical details. **Memory Provenance & Trustworthiness** – compressed summaries can hide omissions or bias; provenance tags and confidence scores are needed to gauge the reliability of each agent's condensed contributions. **Integrating Memory with Agent Communication & Planning** – dialogue and planning protocols must account for peers retaining different granularities of past events and explicitly negotiate misunderstood or missing context.

From this analysis, four pressing gaps were uncovered. Of these, the two most foundational and relevant to this project were:

- Memory Provenance: Existing architectures, especially opaque parametric memory, lack strong methods for agents to verify the origin, reliability, or integrity of shared information. In MAS, where agents might be non-cooperative or depend on distributed knowledge, establishing trust in shared memory is essential.
- Data Integrity: Keeping shared knowledge accurate against accidental errors or deliberate tampering is vital for coordinated action, but remains challenging without a central trusted authority.

1.4. Decentralised Ledger Technology as a Potential Solution

The main challenge in any MAS is building trust in a decentralised setting. How can one agent be sure that information from another is genuine and has not been tampered with, particularly without a central, trusted authority? In examining the literature for potential solutions to issues of memory provenance and data integrity, Decentralised Ledger Technology (DLT), with blockchain as its most well-known example, offers a convincing option. By design, DLT possesses specific properties that are especially helpful in tackling these challenges. This research chose DLT as its core technology because its foundational principles directly provide strong, practical solutions to the problems faced by traditional shared memory systems. The upcoming sections will explore the DLT features utilised in this study.

1.4.1. Verifiable Provenance through Public-Key Cryptography

The most essential requirement for trustworthy memory is verifiable provenance, which is the ability to definitively and computationally verify the origin of a piece of information. DLT achieves this through built-in support for public-key cryptography and digital signatures. Each

participant in the system is represented by a cryptographic key pair: a private key kept secret and a public key that serves as its unique identifier. When an agent records a memory entry in the ledger, it uses its private key to sign the entry. This signature is unique to both the data and the agent's private key; any change to the data, no matter how small, would invalidate the signature (Nakamoto, 2008).

This mechanism offers a robust guarantee called non-repudiation. Because only the holder of the private key could have created a valid signature for a given public key, an agent cannot later deny having authored a message it has signed (Swan, 2015). This directly addresses the provenance gap identified in the literature, where knowledge in opaque systems like parametric memory lacks explicit source tags, making it impossible to verify or attribute claims. In this DLT architecture, every memory entry is inherently linked to its originator, providing an indisputable record of who said what. This forms the foundational layer of trust upon which all subsequent multi-agent coordination is built. As Zyskind, Nathan, and Pentland (2015) note, this enables the creation of a "personal data management platform that provides users with ownership and control over their data," a principle that can be directly applied to autonomous agents.

1.4.2. Data Integrity and Immutability via Chained Hashes

While provenance establishes the origin of data, data integrity guarantees that the data has not been altered since its creation. DLT achieves a strong form of data integrity, often called immutability, through its unique structure of a cryptographically linked chain of blocks. Each block in the chain contains a set of transactions (or memory entries) and, importantly, a cryptographic hash of the entire previous block (Crosby et al., 2016). A cryptographic hash function acts like a "digital fingerprint," converting any amount of data into a fixed-size, unique string.

This chaining of hashes creates a powerful "avalanche effect." If an attacker tries to change a single transaction in a past block, the block's hash would change. This change would then invalidate the hash stored in the next block, causing a chain reaction that invalidates every subsequent block in the entire chain. To make such a change appear legitimate, the attacker would need to recompute the proof-of-work (or other consensus mechanism) for the altered block and all subsequent blocks faster than the rest of the network, an act deemed computationally unfeasible in any reasonably decentralised network (Yli-Huumo et al., 2016). This structure provides a strong defence against covert tampering or falsification of historical records, directly addressing the data integrity gap and offering a shared, immutable history for all participants.

1.4.3. Intrinsic Auditability and Transparency

A direct outcome of verifiable provenance and data integrity is the system's built-in auditability. Since the ledger is an append-only, chronologically ordered, and transparent record of all transactions, any participant (or external observer) can independently reconstruct and verify the entire history of events (Zheng et al., 2018). This transparency itself acts as a security feature. It helps prevent malicious behaviour, as any fraudulent or conflicting entry an agent

adds to the ledger will be permanently visible to all other participants. For multi-agent systems, this provides a particularly effective mechanism for debugging, accountability, and conflict resolution. If two agents hold conflicting information, they can both trace their beliefs back through the unchangeable ledger to identify the sources of those beliefs, enabling a deterministic resolution based on a shared, verifiable history.

1.4.4. Foundation for Concurrency Management through Consensus

A key issue in any distributed system, as first defined by Lamport (1978), is the ordering of events without a global clock. In shared-memory contexts, this presents a challenge for concurrency control: if multiple agents attempt to write to memory simultaneously, which write is accepted first? DLTs resolve this issue through a consensus mechanism. The primary purpose of a consensus algorithm (e.g., Proof-of-Work, Proof-of-Stake) is to establish a deterministic process that enables the distributed network of nodes to agree on a single, canonical order of transactions, and thus a single version of the shared history. This provides a solid foundation for managing concurrent access to shared state (Xu, Weber, & Staples, 2019). Importantly, the consensus mechanisms used by public blockchains are typically designed to be Byzantine Fault Tolerant (BFT). This means they can function correctly and maintain consistency even if a certain percentage of participants act maliciously or are faulty (Cachin & Vukolić, 2017). This offers a much stronger guarantee than traditional distributed database consensus algorithms like Paxos or Raft, which are usually only Crash Fault Tolerant (CFT) and assume a trusted, non-adversarial environment (Cachin & Vukolić, 2017). By inheriting the BFT properties of the underlying ledger, the architecture is inherently resilient to some malicious behaviours that could threaten simpler distributed systems. By leveraging these four DLT characteristics, this research explores a DLT-inspired architectural approach that directly addresses the critical challenges of trust and data integrity in MAS, as identified in the literature review (Ross, 2025) as urgent and underexplored.

1.5. Research Aim, Hypothesis, and Core Question

While blockchain technology presents a compelling theoretical solution for ensuring provenance and data integrity, its practical feasibility as a foundation for multi-agent coordination remained an open question. Therefore, the central aim of this research was to bridge this gap by designing a proof-of-concept architecture and, more critically, by quantitatively evaluating its viability. This evaluation rigorously tested the system's core performance, economic costs, and scalability across varying data loads.

The project specifically investigated the hypothesis that:

H_1: A blockchain-based system could provide a robust and feasible solution for verifiable provenance, thereby enabling trustworthy, shared memory for AI agents.

The resulting null hypothesis that will be experimentally tested will be:

H_0: A blockchain-based system does not provide a robust and feasible solution for verifiable provenance.

1.6. Significance and Contribution

This research project is positioned to make a significant and timely contribution to the underdeveloped area of trustworthy shared memory in multi-agent systems (MAS), a critical bottleneck for the advancement of collaborative artificial intelligence. The significance of this work is not merely incremental. By directly confronting the foundational issues of provenance and integrity with a novel methodological approach, it aims to provide new knowledge and establish a crucial baseline for a nascent field. The anticipated contributions of this project can be articulated across three primary domains: (1) the delivery of a practical architectural blueprint for a problem that is currently addressed only theoretically, (2) the establishment of a comprehensive and novel evaluation framework for quantifying the feasibility of such systems, and (3) its role as a foundational stepping-stone for more complex future research.

1.6.1. An Architectural Blueprint for Verifiable Provenance

A primary contribution of this research will be the design and implementation of a tangible, open-source architectural blueprint for achieving verifiable provenance in a multi-agent context. The existing literature, while rich with designs for single-agent memory, largely theorises about the challenges of multi-agent coordination without providing concrete, testable solutions. Knowledge embedded in opaque parametric memory, for instance, lacks the explicit source tags necessary for agents to audit or trust each other's claims, a problem noted but not solved in current surveys (Petroni et al., 2019; Ross, 2025). This project will address this gap by designing a functional proof-of-concept that operationalises the concept of trustworthy memory. By integrating public-key cryptography at the agent level with a smart contract as an immutable log, the resulting blueprint will serve as a practical reference implementation. This shifts the academic conversation from a theoretical need for trust to a demonstrable, testable artifact, providing a valuable resource for other researchers who can then build on, critique, or design comparative experiments against this model.

1.6.2. A Foundational Evaluation Framework for Feasibility

Perhaps the most significant academic contribution of this research is the establishment of a comprehensive and methodologically sound framework for evaluating the feasibility of DLT-based AI systems. The literature currently lacks a standardised approach for this; multi-agent benchmarks are sparse and often focus on task-specific outcomes rather than the performance and cost of the underlying memory architecture itself (Gao et al., 2024). This project will address this gap by proposing and implementing a framework that defines feasibility not as a single, abstract goal, but as a multi-dimensional concept encompassing reliability, performance, and economic cost.

The novelty of this framework lies in integrating blockchain-specific metrics as first-class citizens alongside traditional performance measures. The planned measurement of Average Gas Cost (AvgGas) will provide a crucial assessment of economic viability, a primary concern for any practical deployment that is often overlooked. Furthermore, the use of varying payload sizes as an independent variable is designed to transform the experiment from a single-point benchmark into a foundational scalability study. By defining and planning to measure Provenance Verification Accuracy, Transaction Throughput, and Verification Latency within this multifaceted framework, this project will provide a robust, pragmatic methodology for assessing the feasibility of DLT-based AI systems. This evaluation framework itself is a key contribution, as future researchers can adopt it to provide a common language for comparing different decentralised architectures.

1.6.3. A Foundational Evaluation Framework for Feasibility

Finally, this project is significant because it is intentionally designed as a foundational stepping-stone for more advanced research. The current literature acknowledges that the challenges of multi-agent coordination, such as concurrency control, credit assignment, and trust, are complex and interconnected (Guo et al., 2024). Attempting to solve all of them at once is intractable. This research makes a strategic contribution by isolating and addressing the most fundamental prerequisite: a reliable and verifiable data layer.

By providing a quantitative baseline for a simple, Layer 1 architecture, this study will produce the essential data needed to justify and guide future investigations into more complex solutions. For instance, the performance limitations identified in this study will provide a direct, data-driven rationale for subsequent research on the effectiveness of Layer 2 scaling solutions such as Arbitrum (Kalodner et al., 2018). Similarly, by establishing a functional 'happy path' with honest agents, it creates a stable foundation upon which future work can build and test more complex adversarial scenarios or integrate intelligent LLMs performing collaborative tasks. In this way, the project is significant not only for the questions it will answer but also for the new, more sophisticated questions it will enable the field to ask.

Methods & Techniques

2. Justification of Research Methodology

The methodology for this research project integrates principles from two established fields of Design Science Research (DSR) and Simulation Modelling and Analysis (Hevner et al., 2004; Law, 2015). The initial phase followed a DSR approach, which was appropriate given the primary aim of creating and refining an architectural artifact to address the practical problem of untrustworthy shared memory in MAS (Hevner et al., 2004). This involved an iterative cycle of designing the core software components, the agent logic, the smart contract, and the interaction protocols.

After designing and implementing this proof-of-concept, the second phase involved its quantitative assessment. To do this, a rigorous experimental framework based on the principles of simulation modelling was used. As recommended by Law (2015), the experiment was set up with enough independent replications (T=30) and a long run length (N=500). This quantitative, simulation-based approach provides precise, reproducible evidence of the system's performance at the proof-of-concept stage.

2.1. Key Metrics for Feasibility

The evaluation of any new blockchain architecture requires a solid set of performance metrics that not only measure its operational efficiency but also confirm its core security and economic feasibility. To do this, our research focused on four key quantitative metrics, each providing a unique yet complementary view of the system's overall potential: Transaction Throughput (TT), Average Gas Cost (AvgGas), Verification Latency (VL), and Provenance Verification

Accuracy (PVA). These metrics, outlined in Table 3, were carefully selected for their established importance in current blockchain research and their direct relevance to assessing the fundamental aspects of our proposed architecture.

2.1.2 Provenance Verification Latency

PVA was assessed to directly test and confirm the fundamental security promise of blockchain technology. The core innovation of blockchain is its use of a chain of digital signatures to establish verifiable ownership and prevent issues like double-spending. While the reliability of this process is often assumed implicitly in many discussions, explicitly validating it is essential for any new architectural proposal. The idea of using cryptographic proofs to create an immutable and verifiable record of transactions was introduced in the foundational Bitcoin whitepaper by Satoshi Nakamoto (2008). Citing this influential paper, our research highlights a commitment to validating the core principles on which the entire technology rests. Therefore, PVA was measured to validate the system's fundamental reliability. This provides direct experimental confirmation of the core principle of using a chain of digital signatures to establish verifiable provenance, a concept first introduced in the foundational design of blockchain systems (Nakamoto, 2008). It is a confirmation of the integrity and security guarantees inherent to the blockchain paradigm, ensuring the system consistently maintains a trustworthy, tamper-proof history of all operations.

2.1.2 Transaction Throughput

TT, often expressed in Transactions Per Second (TPS), is the most common and vital benchmark for any blockchain system, serving as the primary measure of its capacity and scalability. In blockchain performance analysis, TT is the key metric for quantifying how many transactions a network can process and confirm within a specific timeframe. It is the main indicator of a system's ability to handle high volumes of operations, making it essential for evaluating architectures designed for real-world use. Our choice of TT as the main measure of on-chain write speed aligns with the foundational work of Dinh et al. (2017). Their influential "BLOCKBENCH" paper created a standard benchmarking framework for assessing private blockchain systems, explicitly highlighting throughput as a key performance indicator. High throughput is also essential for widespread adoption, especially in enterprise or high-frequency data environments, emphasising its significance in gauging the practical utility of our proposed system.

2.1.1 Verification Latency

While much of the literature on blockchain performance focuses on on-chain metrics like "time to finality," the speed of off-chain cryptographic operations, particularly client-side signature verification, is a critical, often overlooked, component of the overall "time-to-trust" for a client interacting with a blockchain. VL was thus measured to quantify this "off-chain trust speed." The process by which a node validates a transaction before including it in a block is well understood and comprises stages such as creation, signing, propagation, and validation. Our research isolates the specific client-side verification step and measures it from the perspective of an agent, providing a direct assessment of how quickly an individual can independently

verify the authenticity and integrity of a memory entry. General overview papers on blockchain technology, such as the highly cited survey by Zheng et al. (2018), consistently discuss the comprehensive lifecycle of a transaction, which inherently includes these client-side processes. Therefore, VL was measured to quantify the 'off-chain trust speed.' While the overall transaction lifecycle includes on-chain consensus (Zheng et al., 2018), this metric specifically isolates the client-side cryptographic computation, providing a direct measure of how quickly an agent can independently verify the authenticity and integrity of a memory entry after retrieving it from the ledger." This metric is vital for applications requiring rapid client-side validation, ensuring a responsive and trustworthy user experience even before on-chain finality is achieved.

2.1.1 Average Gas Cost

For any system based on the Ethereum Virtual Machine (EVM), gas directly quantifies computational work and, consequently, economic cost (Wood, 2022). The AvgGas metric is therefore essential for assessing the economic viability and operational efficiency of our architecture. Analysing gas consumption is a key aspect of smart contract engineering and the broader economic evaluation of blockchain applications. Research in this area often focuses on identifying costly gas patterns and developing optimisation strategies to reduce the operational overhead of smart contracts. This is highlighted by Chen et al. (2017), who conducted a systematic study of gas-intensive patterns in smart contracts, emphasising the importance of measuring and optimising gas consumption for the long-term sustainability of decentralised applications. By including AvgGas, we aimed to quantify the computational resources needed for on-chain operations, which directly reflect the monetary cost for users interacting with the system. "To assess the economic viability of the system, Average Gas Cost was included as a key metric. The measurement of gas consumption is a standard practice for evaluating the efficiency of smart contracts on EVM-based platforms, as it directly corresponds to the computational work and, by extension, the monetary cost of on-chain operations (Chen et al., 2017)." This metric is particularly relevant for understanding the architecture's sustainability in environments where transaction fees can significantly affect user adoption and operational budgets.

2.2. Measurement Methods

While the initial research proposal outlined metrics for Tamper Detection Rate, False Positive Rate, and Audit Completeness Ratio, these were not assessed in the live blockchain implementation due to architectural complexities beyond the project's core scope. In the mock simulation, tampering with an entry was simply a matter of modifying an in-memory Python object. However, on an immutable ledger such as a blockchain, data cannot be changed once it has been committed. Therefore, realistically measuring the Tamper Detection Rate would require simulating a malicious actor actively crafting and submitting invalid transactions, then verifying their rejection or failure. This would involve designing specific attack vectors rather than just altering existing data. Likewise, measuring the Audit Completeness Ratio would require implementing a robust, separate auditing subsystem capable of efficiently querying, filtering, and parsing historical event logs from the blockchain, which is a distinct technical challenge from the core task of transaction submission and verification. To maintain a clear

focus on the main research question, validating the feasibility of ensuring verifiable provenance and performance for legitimate entries, the development of these complex adversarial and auditing subsystems was postponed. This decision ensures a thorough analysis of the project's essential architectural components within the project's timeframe, laying a strong foundation for future work that could explore more advanced scenarios.

With the decision to refine the experimental focus to more directly address the fundamental feasibility aspects, two new, highly relevant metrics were introduced: AvgGas and Performance Under Varying Payload Sizes. These metrics directly reinforce the central hypothesis by shifting the analysis from purely theoretical security to tangible, real-world viability. The gas cost provides a crucial measure of the economic feasibility of storing memory entries, a primary concern for any practical deployment. The analysis of varying payload sizes serves as a foundational scalability study, assessing how the system's performance and cost are impacted by different workloads. Therefore, while the scope was narrowed by omitting the adversarial metrics, the inclusion of economic and scalability analyses provides a more robust and pragmatic validation of the architecture's overall feasibility, arguably offering a more immediate and practical answer to the core research question for a proof-of-concept system.

The four key metrics measured were (see Table 3):

- PVA: For each of the *T*=30 simulation runs, PVA was calculated as the total number of legitimate entries that returned True from the cryptographic signature verification function, divided by the total number of legitimate entries that were submitted (*N*=500). A final score of 1.0 indicates that every signature was successfully validated for that run.
- TT: For each run, a timer was started immediately before the submission of the first of *N*=500 transactions. The timer was stopped only after the script received the final transaction receipt from the blockchain for the 500th transaction. The TT was then calculated as the total number of successfully confirmed transactions (*N*) divided by the total elapsed duration in seconds, yielding a final unit of transactions per second (*t* x *n* / sec).
- VL: This metric was measured *after* the transaction submission phase. For each of the *N*=500 entries in a run, a high-precision timer was started immediately before the verify signature function was called and stopped immediately after it returned a result. This individual duration was recorded in milliseconds. The final VL for that run is the statistical average (mean) of all 500 of these individual time measurements, representing the average computational time for a single off-chain verification.
- AvgGas: For every successful transaction within a run, the transaction receipt object returned by the *web3.py* library contains a gasUsed field. This value, representing the exact amount of computational work consumed by each transaction on the EVM, was collected for *N*=500 transactions. The final AvgGas metric for that run is the statistical average (mean) of the 500 individual gas values.

.

Table 3: Key Metrics for Feasibility.

Metric	Definition Example	Measurement Method
Provenance Verification Accuracy (PVA)	1.0 or 100%	Measures system reliability. Total number of legitimate entries that passed their cryptographic signature verification, divided by the total number of legitimate entries that were submitted. A score of 1.0 (100%) indicates that the verification process was perfectly reliable for that run.
Transaction Throughput (TT)	5/s	Measures on-chain write speed. Total time in seconds from the start of the first transaction submission to the confirmation of the last. The TT is then calculated as the total number of successfully confirmed transactions divided by the total time.
Verification Latency (VL)	3.2 ms	Measures off-chain trust speed. During the verification phase of each run, the time taken for each signature check is recorded in milliseconds. The VL for that run is the statistical average (mean) of all these individual time measurements. A lower VL indicates that agents can verify and trust data more quickly.
Average Gas Cost (AvgGas)	41,400	Assesses the economic viability. Every time a transaction is successfully confirmed, the blockchain returns a receipt showing the exact amount of computational work, or 'gas,' used. This gasUsed value was collected for every transaction in a run. The final metric is the average of all the individual gas values, providing a stable measure of the computational cost of storing a single memory entry.

2.3. Simulation Environment

The complete, self-contained setup and code are available via the project's GitHub repository. This can be accessed at the following link: https://github.com/jimy-r/RM8502---Research-Project-

As shown in Figure 2 and in Table 4, the experiment was conducted in a controlled local environment to ensure reproducibility. The system consisted of Python-based agents that signed data. These agents used the standard Web3 library to communicate with a local Anvil-powered blockchain node, which provided a complete Ethereum Virtual Machine environment.

On this blockchain, a simple *Solidity* smart contract was deployed, serving as the append-only ledger that logged signed data via events. The core action was the *addEntry* function, which

logged the signed data to the chain. This complete, self-contained setup enabled precise, repeatable collection of all performance and cost data.

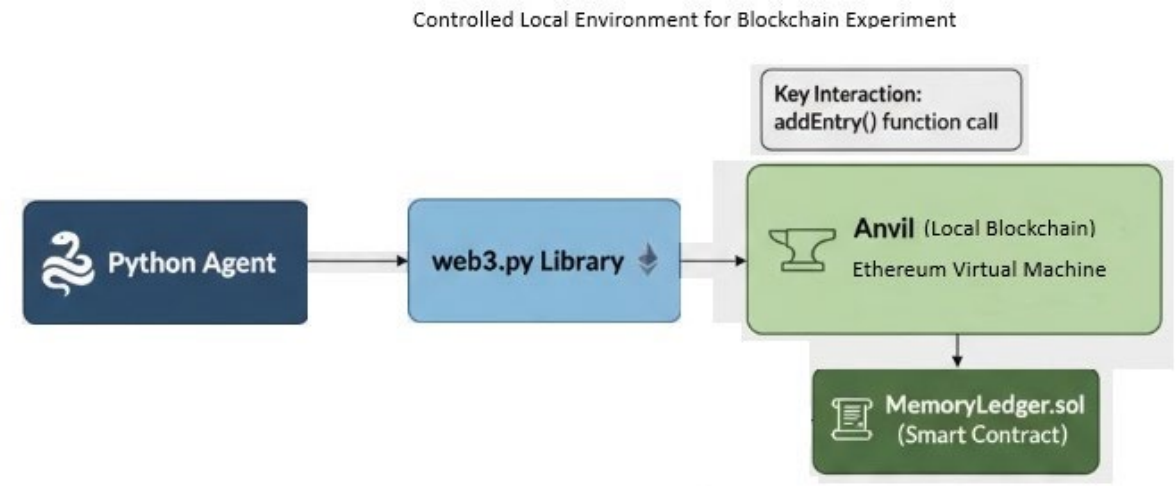


Figure 2: Simulation Design.

The system consisted of the following components (see Table 4 below):

- Agent Runtime: Python *asyncio* was utilised (Python Software Foundation, 2025). It is lightweight, well-suited for I/O-bound operations (such as interacting with a ledger), and facilitates easy wallet operations for signing and transaction submission through libraries like *Web3.py*.
- LLM Backbone: LLM integration will be stubbed or minimal. The core research focus is on the ledger architecture and its trustworthiness properties, not on complex agent reasoning or natural language processing.
- Ledger: An *Anvil* fork of the OP Stack (Paradigm, 2023), running as a private development network, was used. *Anvil* provides a fast, local Ethereum node ideal for development and testing. It provides full EVM (Ethereum Virtual Machine) control, enabling easy deployment and testing of smart contracts.
- Smart Contracts: *Solidity* (version 0.8.x or later) was used for a contract-based state management approach for ledger entries (Ethereum Foundation, n.d). *Solidity* is the standard language for EVM-compatible blockchains. Smart contracts were used to define the on-chain protocol logic, including functions such as *addSignedEntry* and *verifyEntrySignature*.
- Cryptography: Standard, well-vetted cryptographic libraries were used, such as Python's *hashlib* for hashing (Python Software Foundation, n.d), and *ecdsa* package (Danger & et al., n.d) via *web3.py* for elliptic curve digital signatures, commonly used in Ethereum. The use of established cryptographic primitives is crucial for ensuring the foundational security of provenance via signatures.

Table 4: System Components.

Layer	Choice	Rationale
Agent Runtime	Python asyncio	Lightweight, good for I/O (interacting with ledger), easy wallet ops via Web3.py.
LLM Backbone	Stubbed / Minimal	The core focus is on ledger architecture & trust, not on complex agent reasoning. LLM integration was out of scope for this project.
Ledger	Anvil fork of OP Stack (private devnet)	Full EVM control for design/simulation; allowed easy deployment/testing of smart contracts.
Smart Contracts	Solidity 0.8.x (if contract-based state management is chosen for entries)	Standard for EVM; used to define on-chain protocol logic for addSignedEntry, verifyEntrySignature.
Cryptography	Standard libraries (e.g., Python's hashlib, ecdsa via web3.py)	For signing/verifying agent transactions/entries. Required for provenance.

2.3. Experimental Design

2.3.1 Design Flow

This experimental design is based on a single-factor experiment informed by simulation modelling best practice and structured as a series of nested processes to ensure that the collected data is both robust and comprehensive (Montgomery, 2019; Law, 2015).

- Phase 1 - Setup & Initialisation: This initial phase sets the static parameters for the entire experiment. The number of simulations (T=30) is chosen to ensure statistical significance under the Central Limit Theorem. The number of transactions per run (N=500) is chosen to ensure each run reaches a "steady state," providing an accurate measurement of sustained performance. The three distinct payload sizes are defined to serve as the independent variable for the scalability analysis.
- Phase 2 - Execution & Data Collection: This is the core of the experiment, structured as a nested loop. The nested loop consists of:
    - The Outer Loop iterates through each payload size (small, medium, large). This ensures that a complete, statistically valid experiment is run for each condition.
    - The Inner Loop iterates T=30 times. This is the statistical sampling loop. Each of the 30 passes provides one independent data point for each metric (e.g., one value for average TT, one for average VL, etc.).
    - Inside a Single Simulation, a sub-process runs N=500 times to submit transactions, collecting the Gas Cost for each. After the submission loop, all entries are verified to collect VL and PVA. Finally, the metrics for this single run are calculated and aggregated.

- Phase 3 - Per-Payload Statistical Analysis: Once the inner loop completes all 30 runs for a specific payload (e.g., "small"), the script immediately performs a full statistical analysis on that set of 30 data points. It calculates descriptive statistics (mean, standard deviation), confidence intervals, and runs the necessary hypothesis tests. It then generates and saves the individual plots for that specific payload size before the outer loop continues to the next size.
- Phase 4 - Final Comparative Analysis: After the outer loop has completed all three payload experiments, this final phase aggregates the results from all three runs. It generates the crucial final summary table and the comparative boxplots, which directly visualise how performance and cost scale with data size. This is the ultimate output of the experiment, providing the data needed to draw the main conclusions of the research.

This structured design successfully ensures that the experiment is reproducible, statistically robust, and directly addresses the research questions regarding feasibility, performance, and scalability.

2.3.2 Preliminary Testing Phase

The final experimental design was directly informed by a two-phase preliminary study, a critical methodological precursor to the main data collection effort. The purpose of the pilot studies was to assess the feasibility of the main experiment, refine the methodology, and gain initial estimates of key parameters (Law, 2015). Initially, a purely Python-based mock simulation was developed, enabling rapid iteration and validation of the statistical analysis pipeline without the overhead of blockchain interaction. This 'test-in-a-vacuum' approach proved invaluable, as it confirmed the correctness of the analytical framework before introducing the complexities of the *Web3* stack. The insights gained from this phase, particularly regarding initial performance estimates and the robustness of the script, were then used to define the statistically significant and methodologically sound parameters for the full, formal experiment detailed in Section 2.3.1.

During the initial development and validation phase of the project, simulation parameters were intentionally set to a reduced scale, typically using T=3 simulation runs with N=50 transactions per run. This approach was not intended for final data collection or statistical inference, but rather to facilitate an efficient, iterative workflow for refining the script's functionality. The primary advantage of these smaller parameters was the rapid execution time, which allowed for quick cycles of coding, testing, and debugging. This rapid feedback loop was instrumental in verifying the script's core logic, including the successful connection to the Anvil blockchain, the deployment and interaction with the smart contract, the integrity of the data collection pipeline, and the functional correctness of the final analysis and visualisation functions. By ensuring the script was robust and error-free on a smaller scale, it established the necessary confidence to proceed with the final, more time-intensive data collection phase, which utilised larger, statistically significant parameters for the formal analysis.

2.3.3 Determining Run Length (N) for Accurate Measurement

For the final data collection phase of the experiment, a significantly high number of transactions per run (*N*=500) was selected to ensure the statistical robustness and accuracy of the key performance metrics. The choice of a long run length is a critical best practice in simulation modelling, primarily to mitigate the effects of initialisation bias (Law, 2015). A simulation with a small number of transactions is susceptible to measurement noise, where the non-trivial overhead of script initialisation and termination can disproportionately influence the final calculation, creating a "transient" or biased result.

By utilising a large *N*, the simulation is forced to operate for a prolonged period, allowing the system to reach a steady state. As described by Law (2015), a system is in a steady state when its performance metrics are no longer influenced by the initial starting conditions. In this state, the impact of the initial overhead is amortised across many operations, ensuring that the measured throughput accurately reflects the ledger's sustained transaction-processing capacity rather than transient startup effects.

Furthermore, a high volume of transactions yields a more comprehensive dataset for Verification Latency (VL), enabling a detailed analysis of its distribution through a well-populated histogram. This is critical for identifying not just the average latency, but also outliers or performance inconsistencies that would be statistically invisible with a smaller sample size. Therefore, this methodical approach ensures that each simulation run produces a high-quality, reliable set of metrics, thereby strengthening the validity of the final statistical analysis across all runs.

2.3.4 Experimental Parameters

The final experimental parameters were:

- Independent Simulation Runs (*T*): 30. This choice was methodologically driven by the Central Limit Theorem, ensuring that the distribution of the sample means for each metric would be approximately normal, thereby validating the use of *t*-tests and the calculation of reliable 95% confidence intervals.
- Number of Transactions per Run (*N*): 500. This large number was chosen to ensure each run reached a "steady state," where the impact of initialisation and termination overhead is minimised, providing an accurate measurement of sustained performance.
- Independent Variable (Payload Size): To test scalability, the entire experiment was run three times, once for each of three data payload sizes:
    - Small: Approx. 90 bytes, representing a short status update.
    - Medium: Approx. 190 bytes, representing a descriptive observation.
    - Large: Approx. 350 bytes representing a detailed report.

The final total workload for this experiment was 45,000 transactions.

2.4. Data Collection & Analysis Pipeline

A key part of the methodology was the automated analysis pipeline. As soon as the simulation runs were complete for a given payload, the blockchain_simulation.py script didn't just stop but immediately processed the 30 data points it had collected:

- Data Collection: For each of the N=500 transactions in a run, the *gasUsed* value was collected from the transaction receipt. Following the submission phase, all 500 entries were cryptographically verified, and the individual latency for each verification was recorded.
- Statistical Analysis: The script then performed a full statistical analysis on the set of 30 data points for each metric, calculating the mean, standard deviation, and 95% confidence intervals. A one-sided *t*-test was also performed for the PVA metric.
- Outputs: All statistical results were printed to a console log. The script also automatically generated and saved all visualisations, including per-payload boxplots, latency histograms, and the final comparative plots as PNG files, and saved the final summary data to a CSV file.

2.5. Statistical Validation

After completing 30 simulation runs for each payload condition, the combined data for each metric underwent a thorough statistical validation process to extract meaningful insights from the raw outputs. For each of the four key metrics, PVA, TT, VL, and AvgGas, the sample mean (μ) and standard deviation (σ) were calculated across the 30 independent runs to assess the central value and variability of the results. To estimate the range within which the true population mean for each metric is likely to fall, a 95% confidence interval was constructed using a t-distribution, appropriate for the sample size. Additionally, a formal hypothesis test was performed on the critical metric, PVA, to determine whether the system met its predefined reliability target. A one-sided t-test was conducted at a significance level (α) of 0.05, testing the null hypothesis (H_0) that the true mean PVA was less than 0.99 against the alternative hypothesis (H_1) that it was at least 0.99. The p-values and confidence intervals obtained from this analysis provided a statistically solid basis for the conclusions in the Results and Discussion sections of this report.

2.6. Ethical Considerations

While this research project did not require a formal ethics review for human or animal subjects, it was nonetheless conducted in accordance with the established ethical principles governing the computing profession and responsible research. The 'agents' central to the experiment are purely computational constructs, and the ethical framework for this study is therefore based on the principles of professional conduct, data integrity, and responsible innovation.

1. No Human or Animal Subjects: The study's methodology is entirely computational and does not involve human participants or animal subjects. As a result, ethical protocols related to human subject research (such as informed consent) and those concerning animal welfare are not applicable.

2. Data Management and Integrity: In line with the principle to "be honest and trustworthy" (ACM, 2018), all data generated throughout the project, including design documents, analytical notes, and simulation logs, were managed securely to ensure their integrity. Since the simulation used only synthetic data, there were no privacy concerns about the data itself. The automated analysis pipeline further ensured that the results were a direct, unaltered representation of the raw experimental data.

3. Responsible Design: The primary aim of this research is to improve trustworthiness and provenance in shared memory systems, a goal that positively contributes to the principles of responsible AI development. The design of mechanisms for verifiable data origins aligns with ethical frameworks that call for AI systems to be beneficial and reliable (Floridi & Cowls, 2019). By establishing the foundation for more accountable multi-agent systems, this project adheres to the professional obligation to "contribute to society and human well-being" (ACM, 2018, Principle 1.1. 1).

4. Openness and Transparency: The methods, code, and non-proprietary findings of this research are documented and publicly available in the project's GitHub repository. This commitment to openness and transparency ensures the work is scrutable and reproducible, and contributes to open scientific discourse, fulfilling the professional responsibility to "be transparent and to give a comprehensive account of system functionality" (ACM, 2018, Principle 3.5.5).

3. Results

3.1 Results Summary

The final experiment, conducted across 90 simulation runs and 45,000 transactions, provided a clear and thorough understanding of the proposed DLT-based architecture. Table 5 offers a detailed breakdown of the average performance for each of the three payload conditions, while Table 6 presents a summary of the descriptive statistics for key metrics aggregated across all 90 experimental runs.

Overall, the data show several definitive quantitative results. As shown in both tables, the Provenance Verification Accuracy (PVA) was consistently 1.00 (100%) with a standard deviation of 0.00 across all runs. The scalability analysis in Table 5 reveals that the Average Gas Cost (AvgGas) increased steadily with data size, from 37,578.25 for small payloads to 46,905.36 for large payloads. Conversely, the Transaction Throughput (TT) demonstrated a clear inverse relationship with payload size, dropping from an average of 20.10 txn/sec for small payloads to 10.25 txn/sec for large payloads. As shown in Table 6, the overall mean TT across all experiments was 10.25 txn/sec, though this value is largely influenced by the large

payload runs. In stark contrast, the Verification Latency (VL) remained highly stable across all conditions. As detailed in Table 6, the overall mean VL was 2.81 ms, with a very low standard deviation of 0.08 ms and a tight 95% confidence interval of [2.78, 2.85], confirming the high consistency and efficiency of this off-chain operation across workloads.

Table 5: Results Summary.

Metrics	**Small Payload**	**Medium Payload**	**Large Payload**
AvgGas	37578.25	41400.99	46905.36
PVA	1.00	1.00	1.00
VL (ms)	2.80	2.81	2.81
TT (txn/sec)	20.10	13.73	10.25

Table 6: Descriptive Statistics Across All Runs.

Metric	**Mean**	**Std**	**95% C.I (t-dist)**
AvgGas	46905.36	0.49	46905.18, 46905.55
PVA	1.00	0.00	nan, nan
VL	2.81	0.08	2.78, 2.85
TT	10.25	1.99	9.50, 10.99

The complete data output is available in the project's GitHub repository (Section 2.3).

3.2. Flawless Provenance Verification

The primary goal of achieving verifiable provenance was accomplished with clear success. Out of all 45,000 transactions carried out during the 90 simulation runs, the Provenance Verification Accuracy (PVA) reached a perfect 1.0 (100%). This complete reliability is illustrated in Figure 3 below, where the PVA boxplot appears as a flat line at 1.0, consistent across all three payload sizes. The descriptive statistics in Table 6 further support this, showing that the mean PVA over all 90 runs was 1.0 with a standard deviation of 0.00, indicating no variance in the system's accuracy. This provides, with high statistical confidence, evidence that the architecture's cryptographic basis is perfectly reliable. The formal hypothesis test for PVA confirmed this with a p-value of 0.0000, leading to a decisive rejection of the null hypothesis and strong statistical backing for the system's ability to ensure verifiable provenance.

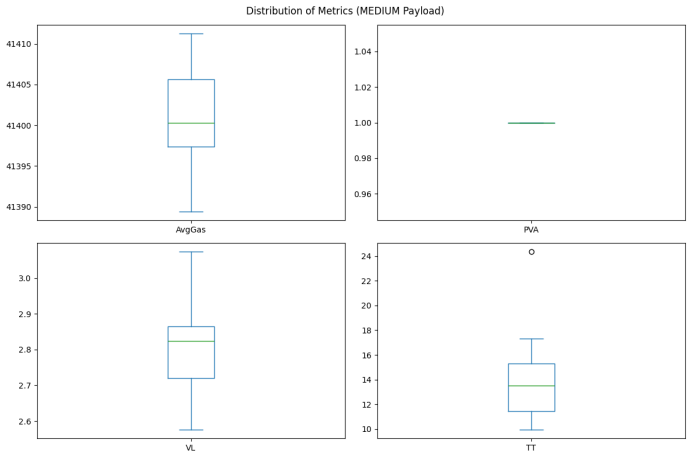


Figure 3: Distribution of Metrics (Medium Payload).

3.2. Predictable and Linear Economic Cost

The analysis of Average Gas Cost (AvgGas) provides a clear view of the architecture's economic viability. As shown in the comparative boxplot in Figure 4 below, the results reveal a direct, statistically significant, and linear relationship between the data payload size and the gas needed for the transaction. The final average costs rose steadily from 37,578 gas for small payloads, to 41,401 gas for medium, and ultimately to 46,905 gas for large payloads. The statistical analysis in Table 6 highlights the stability of these costs; for the large payload scenario, the mean gas cost was 46,905.36 with a standard deviation of just 0.49. This very low variance, as visually confirmed by the tight boxplots in Figure 4, shows that the economic cost of storing a memory entry is not only scalable but also highly predictable and consistent.

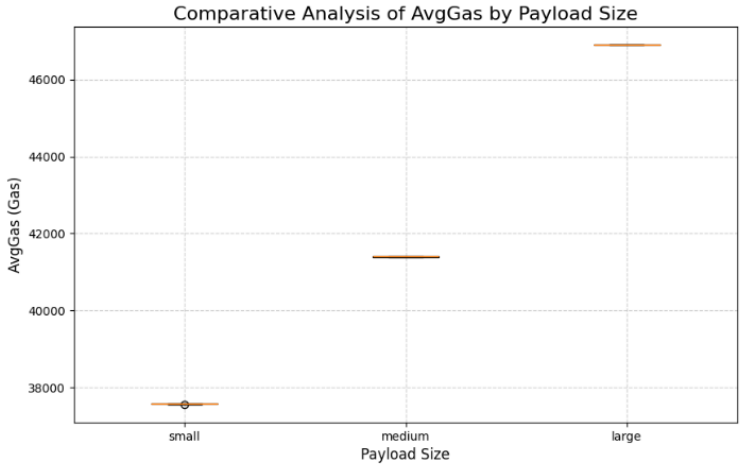


Figure 4: Comparative Analysis of AvgGas by Payload Size.

3.3. The Core Performance Trade-off

The analysis of Transaction Throughput (TT) and Verification Latency (VL) revealed the study's most important finding: a fundamental trade-off between the system's on-chain write performance and its off-chain verification speed.

The system's throughput was significantly below the initial target of 100 transactions per second and exhibited a strong inverse correlation with payload size. As illustrated in Figure 5, the median throughput declined noticeably as the data payload increased, dropping from approximately 20.1 transactions/sec for small payloads to just 10.2 transactions/sec for large payloads. The statistical analysis (Tables 5 & 6) supports this, with the mean throughput for the large payload condition at only 10.25 transactions/sec and a 95% confidence interval of [9.50, 10.99]. This indicates that the on-chain consensus and confirmation process acts as a substantial performance bottleneck.

In stark contrast, the off-chain verification speed was extremely fast and highly consistent. As illustrated in Figure 6, the median latency remained below 3 milliseconds across all payload sizes, indicating that data size has little impact on performance. This stability is supported by statistical analysis (Table 6), which shows that, even for large payloads, the mean verification latency was 2.81 ms with an exceptionally low standard deviation of 0.08 ms. The narrow 95% confidence interval of [2.78, 2.85] provides strong statistical evidence that the verification process remains both fast and very predictable.

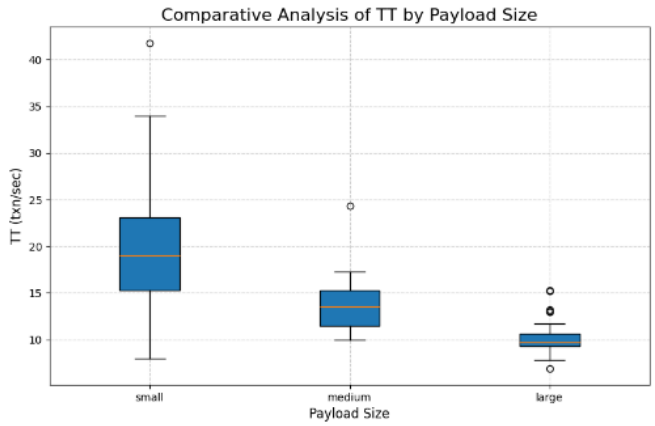


Figure 5: Comparative Analysis of TT by Payload Size.

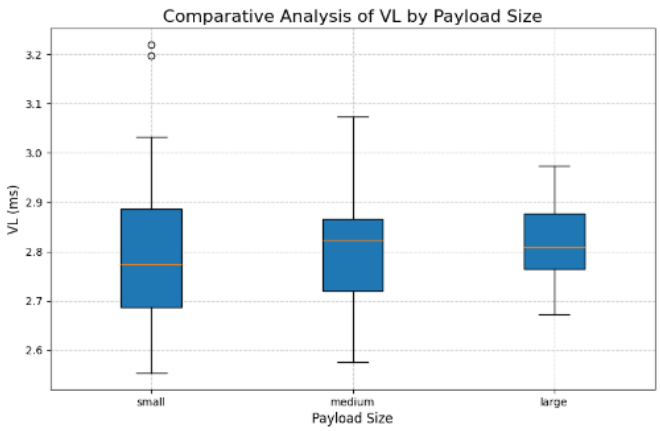


Figure 6: Comparative Analysis of VL Payload Size.

4. Discussion

The experimental results provide a clear, compelling, and multifaceted account of the feasibility, performance, and scalability of the proposed Decentralised Ledger Technology (DLT) - based architecture. This section interprets these key findings, discusses their importance within the wider academic and practical context, critically evaluates the study's limitations, provides recommendations, and explores future research directions.

4.1. Interpretation of Key Findings

4.1.1 Core Hypothesis Validation

The primary goal of this research was to determine whether a DLT architecture could ensure verifiable provenance, which was achieved with unequivocal success, with the PVA measured at 1.00 (100%) across all 45,000 transactions. This finding is substantiated across multiple data representations. Visually, this is shown in Figure 3, where the PVA metric boxplot is flat, indicating zero variance in the results. Numerically, the summary in Table 5 confirms that this result held true for all three payload sizes. Furthermore, the detailed descriptive statistics in Table 6 quantify this by showing a mean of 1.0 and, critically, a standard deviation of 0.00 across the 90 independent runs. This zero-variance outcome is the basis for the high statistical confidence in the system's reliability. It directly led to the formal hypothesis test for PVA, yielding a p-value of 0.0000 and a definitive rejection of the null hypothesis, providing powerful statistical support for the architecture's core function.

The perfect 1.00 (100%) PVA achieved in the local simulation is a testament to the mathematical integrity of the Elliptic Curve Digital Signature Algorithm (ECDSA) underlying it. Under ideal conditions, a correctly signed message will never fail verification. However, in a real-world decentralised network, the PVA metric would inevitably degrade below 100%. This degradation would not stem from a failure of the cryptography itself, but from failures in the complex, multi-step process of transaction submission and retrieval. The measured PVA in a live environment is a composite measure of both cryptographic validity and the reliability of the entire data pipeline. Several real-world conditions could cause a legitimate, correctly signed transaction to fail verification, leading to a drop in measured PVA.

In a real-world decentralised network, the successful inclusion of a transaction in the blockchain is not guaranteed. As documented in empirical studies of the Ethereum transaction market, the mempool, the waiting area for pending transactions, operates as a dynamic marketplace where transactions compete for limited block space (Leonardos et al., 2022). During periods of network congestion, the base gas fee can spike suddenly. If an agent broadcasts a transaction with a maximum fee below this new market rate, validators will ignore it, leading to the transaction being indefinitely delayed or ultimately dropped from the network altogether (Leonardos et al., 2022). This scenario leads to a data availability failure. When an auditing agent later attempts to retrieve and verify the entry from the blockchain, it will not find it, as the transaction was never successfully mined. This would be logged as a verification failure because the expected entry is absent from the immutable ledger. While this is not a cryptographic signature failure, it nonetheless lowers the overall end-to-end reliability of the system and would directly degrade the measured PVA.

Decentralised networks are not instantly consistent; instead, they follow a model of "eventual consistency." After an agent's transaction is successfully included in a block, that block must propagate through the global peer-to-peer network. Empirical studies of the Ethereum network have shown that this propagation, although generally quick, is not immediate and can experience delays depending on network topology and geographic distance (Gencer et al., 2018). When an agent submits a transaction, which is then added to the latest block, a second agent might query the network for this data but be connected to a node that, due to propagation delay, has not yet received the new block. Consequently, the querying agent's request for the data will fail because the entry does not yet exist in its ledger version, resulting in a temporary verification failure. Although this problem will resolve itself once the node synchronises, in a system that demands real-time verification and coordination, this brief period of inconsistency can lead to intermittent PVA degradation. This underscores a key trade-off in decentralised systems: sacrificing perfect, real-time consistency in exchange for decentralisation and resilience.

A subtle but critical threat to the measured PVA is the risk of data encoding errors, which occur when data isn't properly canonicalised before signing and submission. Cryptographic hash functions are deterministic; even a single-byte difference between two inputs results in entirely different outputs. Thus, for a signature to be valid, the exact byte representation of the data signed must match the byte representation stored on-chain and later verified. This threat could arise if a software bug or an inconsistency in the agent's libraries causes the JSON serialisation used in local signing to differ slightly from that in the transaction payload. Variations might include how whitespace is handled, the order of keys in the dictionary, or the encoding of special characters. Creating a standard, or canonical, representation for structured data before cryptographic operations is a well-established principle, underpinning standards such as JSON-LD (Sporny et al., 2020). A failure in this process, even a single-byte mismatch between the signed message and on-chain data, will cause cryptographic verification to fail catastrophically. The signature remains valid for the original message but becomes invalid for the slightly altered data stored on-chain. This constitutes a verification failure and directly impacts the measured PVA, emphasising that end-to-end data integrity relies not only on the blockchain but also on disciplined client-side software.

In a more advanced or adversarial setting, including a transaction is not just a technical issue but also depends on the strategic and economic choices of the network's validators (miners/proposers). Validators are not neutral judges; they are economically rational actors who decide which transactions to include in the blocks they produce. This can cause two types of data availability failures:

- Censorship: A validator might intentionally exclude an agent's transaction from a block, effectively censoring that agent from the shared memory.
- Maximal Extractable Value (MEV): A more subtle and widespread problem is Maximal Extractable Value (MEV), a concept explained by Daian et al. (2020). MEV is the profit that block producers can gain by manipulating the order or inclusion of transactions. For example, a validator might prioritise transactions with higher fees or those involved in profitable arbitrage, while ignoring an agent's less profitable (but still valid) addEntry transaction.

In both censorship and MEV cases, the result is the same as a standard transaction drop. The legitimate transaction is never recorded on-chain. This causes a verification failure due to data unavailability, which would directly and adversely affect the measured PVA. This shows that, in a real-world decentralised network, transaction success depends not only on technical

validity and paying sufficient fees but also on navigating the complex, often adversarial game theory within the underlying consensus layer.

In conclusion, while the cryptographic core of the architecture is theoretically perfect, its real-world PVA would reflect the entire system's end-to-end reliability. The 100% PVA in the simulation should be interpreted as the best-case scenario, confirming the soundness of the design's logic. In a live deployment, the PVA would become a critical Key Performance Indicator (KPI) for the health and reliability of the agent's connection to the decentralised network itself.

4.1.1 Economic Feasibility

The economic analysis of Average Gas Cost (AvgGas) provides a clear and compelling view of the architecture's viability. The results show a highly predictable linear cost model. This is best illustrated in the comparative boxplot in Figure 4, which displays three distinct, non-overlapping boxplots, indicating a statistically significant increase in cost that directly aligns with the rise in payload size.

The numerical data in Table 5 quantifies this linear trend, showing the average cost increased from 37,578 gas for small payloads to 41,401 gas for medium payloads, and up to 46,905 gas for large payloads. Importantly, the variation within each experiment was very low. As shown in the statistical analysis for the large payload condition in Table 6, the mean gas cost of 46,905.36 had a standard deviation of just 0.49. This impressive consistency, seen in the very tight boxplots in Figure 4, proves that storing a memory entry is not only scalable but also highly predictable.

The finding that the computational cost, measured in gas, scales predictably and linearly with data size is a very positive outcome for practical implementation. It enables developers to perform a straightforward cost-benefit analysis of their application’s computational needs, accurately estimating the resources required based on the data they plan to secure. However, a key distinction must be made between this fixed computational effort and the variable monetary cost of a transaction.

On a live EVM network, the final transaction fee is set by a dynamic fee market, which includes a baseFee that adjusts algorithmically according to network congestion and a priorityFee or 'tip' to incentivise validators. As detailed in the dynamical analysis of Ethereum's fee market by Leonardos et al. (2022), this mechanism creates a highly volatile environment in which the transaction cost can vary widely. Consequently, a transaction costing around 47,000 gas, as seen in this experiment, might be worth a few cents during quiet periods but could jump to several dollars during peak demand. This shifts the economic perspective, as the system's viability depends not only on its efficiency but also on the target network's real-time economic conditions. This important caveat makes the architecture potentially unsuitable for applications that require frequent, low-value data entries, even though the underlying computational cost structure remains sound, transparent, and predictable.

4.1.2 Performance & Scalability

The analysis of Transaction Throughput (TT) and Verification Latency (VL) is the most revealing part of the study, a fundamental trade-off between on-chain performance and off-chain verification speed. On one side, the experiment uncovered a clear on-chain bottleneck.

The system's throughput was notably lower than the initial target of 100 txn/sec and showed a steep, inverse relationship with payload size. This is clearly shown in the comparative boxplot in Figure 5, where the median throughput drops from about 20.1 txn/sec for small payloads to only 10.2 txn/sec for larger ones. The numerical summary in Table 5 accurately quantifies this decline. Additionally, the detailed statistical analysis for the large payload condition in Table 6 confirms this limitation, with a mean throughput of just 10.25 txn/sec and a 95% confidence interval of [9.50, 10.99]. This key finding indicates that the sustained throughput is fundamentally limited by the on-chain transaction confirmation process. While the high variance and outliers in the boxplots suggest that the local node's performance can fluctuate, the overall downward trend is clear, showing that the design is not suitable for high-frequency data logging. In stark contrast, the off-chain efficiency was outstanding. The VL was remarkably low and stable, averaging around 2.8 ms across all payload sizes. The comparative analysis in Figure 6 confirms this stability, with three boxplots showing nearly identical medians and distributions, indicating payload size has little effect on verification time. The statistical data in Tables 5 and 6 further support this. For the large payload, the mean VL was 2.81 ms, with a very low standard deviation of 0.08 ms. This is a major success for the architecture, demonstrating that off-chain cryptographic verification is highly efficient. This allows agents to query and access trust ledger information nearly in real time, which is vital for coordination and decision-making tasks that require trust. Overall, this experiment successfully offers a comprehensive answer to the research question on the feasibility of a DLT architecture for trustworthy memory. The main finding can be summarised as the core trade-off: the proposed architecture sacrifices high on-chain write performance for fast, near-instantaneous off-chain trust and verification. This makes the system particularly well-suited for applications where the value of the data and the need for tamper-proof provenance are high, but the data submission frequency is relatively low. Examples include recording critical agent decisions, logging completed high-level tasks, or storing foundational knowledge that all agents in a system must trust. The project's goals have been met, providing clear, data-backed insights into the architecture's performance, costs, and scalability, setting a strong basis for future research in collaborative AI systems.

4.2. Significance in the Context of Current Literature

The importance of this study's findings goes beyond a simple performance benchmark; it needs to be understood within the wider academic fields of multi-agent systems, distributed systems, and blockchain technology to fully recognise their contribution. This research makes a significant contribution by offering a quantitative, empirical link between the theoretical potential of Decentralised Ledger Technology (DLT) and the real-world challenges of multi-agent memory. In particular, the main discovery of the "slow write, fast trust" trade-off serves as an important heuristic for a field dealing with the complexities of decentralised trust, and it both builds on and distinguishes itself from the traditional distributed systems literature.

4.2.1 Differentiating from Classic Distributed Systems

A key aspect of this research's importance is its positioning of blockchain-based systems relative to traditional distributed systems. The field of distributed systems has examined consensus and replicated state machines for decades, with foundational protocols such as Paxos (Lamport, 1998) and its more intuitive successor, Raft (Ongaro & Ousterhout, 2014), providing reliable solutions for maintaining consistency within a known group of servers. These algorithms are highly efficient and form the foundation of modern cloud computing

infrastructure. However, they are mainly designed for permissioned environments where the set of participating nodes is fixed and mutually trusting. Their security models are primarily Crash Fault Tolerant (CFT), meaning they can handle node failures but are not built to withstand malicious, coordinated attacks from within the set of consensus participants.

In contrast, the blockchain-based architecture assessed in this study is intended for permissionless, 'trust-less' environments. This is a fundamentally different and more demanding operational domain, typical of many open multi-agent systems where participants can be anonymous, ephemeral, and potentially adversarial. The architecture inherits the Byzantine Fault Tolerance (BFT) of the underlying blockchain, a concept originating from the work of Lamport, Shostak, and Pease (1982). This means the system is designed to preserve consistency and withstand failure even when some of its participants are malicious and trying to subvert the protocol. By providing a quantitative performance analysis in this BFT context, this research provides a valuable data point for a class of systems that require stronger security and decentralisation guarantees than traditional distributed systems typically offer, even at the cost of lower throughput.

4.2.2 Refining the Concept of Trust in Multi-Agent Systems

Furthermore, this research adds to the multi-agent systems (MAS) literature by clarifying the often-vague idea of "trust." In MAS, trust is a complex, layered concept, often understood as a social-cognitive phenomenon involving reputation, previous performance, and assumptions about intent (Sabater & Sierra, 2005). While these higher-level trust models are influential, they frequently depend on subjective or game-theoretic beliefs. This project makes a notable contribution by quantitatively assessing a fundamental, non-negotiable layer, cryptographic trust.

The architecture proposed here tackles the most basic yet vital question of trust: "Can I be certain that this information genuinely comes from the stated source and hasn't been tampered with since it was created?" By achieving a perfect 100% PVA, this study shows that DLT can offer an exceptionally dependable "ground truth" for agent interactions. This unchangeable, verifiable record of who said what and when forms the essential basis for constructing more complex, higher-level social trust models. For example, an agent reputation system becomes significantly more resilient when it can rely on a permanent, unforgeable history of an agent's past statements and actions. Therefore, this work does not aim to replace intricate trust models but rather to establish a solid, technical foundation that enhances their reliability and security.

4.2.3 Contextualising Performance within the Blockchain Scalability Landscape

Finally, the identified on-chain bottleneck must be considered within the rapidly changing landscape of blockchain scalability. The throughput limitations seen in this experiment (10-20 txn/sec) reflect a monolithic, Layer 1 (L1) blockchain environment, where all transactions are processed by a global network of nodes. This performance pattern directly illustrates the "Scalability Trilemma," which states that a blockchain system can only optimise two of three properties: Decentralisation, Security, and Scalability (Buterin, 2017). The low throughput of the emulated L1 in this study is an expected outcome of prioritising decentralisation and security.

However, the advent of Layer 2 (L2) scaling solutions intends to specifically address this issue. As detailed by Kalodner et al. (2018) in their foundational paper on Arbitrum, optimistic rollups handle transactions in a separate, faster execution environment and only post compressed data back to the main L1 chain for security. This enables L2s to achieve significantly higher throughput and much lower transaction costs. Therefore, while this study's results accurately characterise the performance trade-offs on a basic L1, they also serve as an important

benchmark that underscores why L2 solutions are not just an enhancement, but a vital requirement for many practical MAS applications. The findings of this paper provide the quantitative basis for focusing future efforts on L2 deployments, with a high likelihood that such architectures would greatly boost transaction throughput, potentially making the system suitable for a broader range of medium-to-high-frequency applications. In this way, the importance of this study's results is twofold: it establishes the boundaries of a basic approach, while directly pointing to its most promising evolutionary pathway.

4.3. Strengths and Limitations of the Study

The primary requirement of the experiment was to quantitatively evaluate the feasibility of a DLT-based architecture for trustworthy multi-agent memory. The design achieved this with distinction in several key areas:

- Effective Isolation of Key Variables: The design was highly successful at isolating and measuring the most critical variables: provenance verification (PVA), economic cost (AvgGas), on-chain performance (TT), and off-chain performance (VL). By focusing on these core metrics, the experiment provided a clear, straightforward answer to the central research question.
- Adaptive and Iterative Methodology: A key strength of the design was its two-phase, adaptive approach. It started with a mock simulation that allowed for swift development and logical validation without the complexities of a live blockchain. The move from this mock setup to a live Anvil simulation was a vital step. Additionally, the choice to exclude the practically unfeasible TDR/FPR metrics and replace them with more relevant feasibility metrics, such as Gas Cost and Payload Scaling, demonstrated a mature research process. This approach aimed to deliver the most meaningful and realistic insights possible within the project's scope.
- Statistically Sound Final Framework: The final experimental design, incorporating T=30 simulation runs and N=500 transactions, was methodologically robust. The use of T=30 satisfied the conditions of the Central Limit Theorem, providing strong statistical validity to the final analysis and confidence intervals. The choice of N=500 effectively enabled the measurement of "steady state" performance, ensuring the gathered metrics genuinely reflected the system's sustained capacity. This rigorous framework elevated the project from a simple proof-of-concept to a credible performance evaluation.

The execution of the experimental design uncovered several key lessons about the gap between theoretical planning and practical implementation in blockchain research. These lessons directly highlight the main limitations of this study, including the use of an idealised local testnet, the simplicity of the agents and data, and the intentional focus on legitimate, or 'happy path,' transactions:

- The Idealism of a Local Testnet: By design, the experiment was carried out on a local Anvil node. This is a major limitation since it depicts a "best-case" scenario with zero network latency, no network congestion, and a fixed, zero-cost gas price. While necessary for creating a controlled and reproducible experiment, this setup cannot reflect the performance degradation, cost variability, and transaction uncertainty (e.g., dropped transactions) that occur in a real-world public environment network.
- Simple Agents and Dummy Data: A second significant limitation of this study is its use of simple, scripted agents and synthetic, or 'dummy,' data. The Python Agent in the experiment was designed to perform a single, repetitive task, creating a data packet of a predefined size and cryptographically signing it. While this approach was highly

effective for demonstrating the reliability and performance of the core cryptographic mechanism, it did not establish the architecture's usefulness in a real-world, goal-oriented context. The data being stored had no semantic meaning, and the agents were not using this shared memory to reason, plan, or coordinate their actions to achieve a collective goal. Therefore, while the experiment successfully demonstrates that trustworthy memory can be created, it does not yet answer the more complex question of how, or if, access to this trustworthy memory improves the efficiency, robustness, or emergent collaborative capabilities of an intelligent multi-agent system. This distinction between a functional mechanism and validated utility marks a key boundary of the current research and serves as a primary motivation for the future work proposed in the following section.

- Simulating Adversarial Conditions: The most important lesson learned was the practical challenge of measuring metrics like Tamper Detection Rate (TDR) and False Positive Rate (FPR). The initial design assumed that "tampering" could be simulated by altering committed data, which is fundamentally impossible on an immutable ledger. This led to a “happy path” simulation for this research. A more advanced design would have required simulating active adversaries programmed to craft and submit invalid transactions. This realisation highlights a key challenge in blockchain research, that testing security properties often requires building complex attack simulators rather than simply modifying data.

In summary, the experimental design was highly successful for its intended purpose of providing a foundational, statistically sound feasibility analysis of the core architecture. Its greatest strengths were adaptability and the ability to deliver clear, quantifiable results for the most critical metrics. Simultaneously, its deliberate limitations, the use of a controlled local testnet, the focus on simple agents, and the exclusion of adversarial testing, are not failures. Rather, they are carefully defined boundaries that clearly illuminate the path for future, more complex investigations now possible.

4.4. Recommendations

Based on the research's conclusive findings, two main recommendations can be made for practitioners and researchers seeking to develop reliable multi-agent systems using similar DLT-based architectures.

1. Prioritise High-Value, Low-Frequency Data: The experimental results clearly show a trade-off where on-chain write performance is compromised to enable near-instant off-chain trust. Therefore, it is advisable to apply this architecture to use cases where data integrity and immutable provenance are of importance, but where data submission frequency is relatively low. Suitable applications include logging critical agent decisions, storing foundational knowledge trusted by all participants, or recording the completion of major, verifiable tasks, rather than for high-frequency data streams such as raw sensor readings.
2. Strongly Consider Layer 2 Solutions for Initial Deployment: Given that the on-chain throughput was identified as the main performance bottleneck, it is highly recommended that any real-world deployment of this architecture should focus on a Layer 2 scaling solution (e.g., Arbitrum) rather than a Layer 1 mainnet. As this study's findings are typical of an L1 environment, deploying on an L2 is the most direct and effective way to address the primary performance limitation, offering the potential for

significantly higher throughput and lower economic costs, thereby expanding the range of feasible applications.

4.5 Future Research Directions

The foundational work laid out in this project provides a clear, data-driven pathway for future, more complex investigations. The findings, especially the evident trade-off between on-chain throughput and off-chain verification speed, naturally suggest a series of compelling next steps to test real-world applicability, push boundaries, and explore broader socio-technical and ethical issues.

4.5.1. Enhancing the Experimental Environment and Architecture

The most immediate and critical expansion is to move beyond the idealised local testnet and subject the architecture to the rigours of a live, decentralised environment.

To address the limitations of the local testnet, the crucial next step is to conduct a comparative analysis by deploying this architecture on a public Layer 2 network (e.g., an Arbitrum testnet) and a public Layer 1 testnet like Ethereum's Sepolia (Ethereum Foundation, n.d). A comparative study would quantify real-world transaction speeds, the monetary cost of transactions under fluctuating gas prices, and the impact of network latency, providing a far more realistic measure of the system's performance and economic feasibility in both environments.

This study deliberately concentrated on the 'happy path.' A key future development should be the introduction of an adversarial testing phase. This would entail creating malicious agents that deliberately attempt to corrupt the shared memory by submitting transactions with invalid signatures. By doing so, researchers can quantitatively evaluate the originally proposed Tamper Detection Rate, providing a direct and robust assessment of the system's security and resilience against active threats.

Alongside improving the experimental environment and agent complexity, a key area for future research is expanding the architectural design to directly tackle the identified issues of cost and throughput. One promising route is to explore off-chain storage with on-chain hashes, using systems like the Interplanetary File System (Benet, 2014) to handle large data payloads while committing only the unchangeable data hash to the blockchain. This hybrid method could significantly reduce costs but would require a new experimental setup to evaluate trade-offs between data retrieval delay and the challenges of maintaining off-chain data availability. For applications where the on-chain throughput bottleneck is the main constraint, future work should investigate alternative architectures designed for high-frequency interactions. Two promising avenues are the use of state channels or dedicated sidechains, both established off-chain scalability techniques (Zheng et al., 2018). As initially formalised for payment channels by Poon and Dryja (2016), state channels would enable a known group of agents to exchange a large volume of signed memory updates off-chain, with the main chain settling the final state. This method would effectively address the throughput issue for contained interactions but would raise new questions about channel security, management, and the

game-theoretic assumptions required for safe operation. Similarly, a dedicated sidechain could provide high throughput for a specific multi-agent application, but it would pose different challenges, including the security of the bridge connecting the sidechain to the mainnet and the finality of batched settlements.

4.5.2. Increasing Agent and Task Complexity

The current experiment employs simple, reactive agents that, although effective for performance benchmarking, do not assess the architecture's utility in real-world AI scenarios. A key advancement would be to integrate more complex, cognitive agents to evaluate the system's performance with semantically rich data. These agents could be upgraded from basic scripted entities to generative agents powered by LLMs, as demonstrated by Park et al. (2023), to enable believable, autonomous behaviour. Using a state-of-the-art model via an API, such as those offered by OpenAI (2023), this change would facilitate testing more sophisticated use cases.

A compelling demonstration of the architecture's value would involve designing a specific, goal-oriented task that requires agents to collaboratively construct and rely on the shared knowledge base. For example, a "collaborative summarisation" task could be created in which multiple LLM agents read different parts of a document and record their validated summaries in the ledger. This would test how the architecture handles more variable and semantically complex data generated by these intelligent agents, as well as enable measurement of task-specific success metrics, providing a direct evaluation of the architecture's practical utility.

4.5.3 Addressing Broader Socio-Technical and Ethical Challenges

As these technologies advance, future research must extend beyond merely technical performance to address the complex social and technical implications of deploying autonomous, immutable memory systems in real-world settings. This requires an interdisciplinary approach to address challenges in accountability, privacy, and governance.

A vital area for future investigation is the development of Accountability Frameworks. While the non-repudiation provided by this architecture enables information attribution, it does not resolve the "responsibility gap" that arises when autonomous systems make critical decisions (Matthias, 2004). Future research should explore how agents or their creators are held legally and ethically responsible for data they commit to an immutable ledger, especially if that data influences real-world outcomes.

Moreover, if such systems were to store sensitive or personal data, the inherent immutability and transparency of DLT could pose a significant privacy risk. Future research must therefore investigate the integration of advanced privacy-preserving architectures. This may involve structural modifications to include zero-knowledge proofs, a robust cryptographic method that enables one party (the prover) to demonstrate to another (the verifier) that a statement is true, without revealing any information beyond the statement’s validity. As shown in the foundational design of privacy-preserving systems like Zerocash (Sasson et al., 2014), this technique can be applied to a blockchain to verify on-chain actions while keeping the underlying data entirely private. For multi-agent memory, this would allow an agent to provide

an on-chain attestation that it possesses a piece of off-chain data that meets certain criteria, thereby balancing the need for cryptographic verification with the right to privacy.

Lastly, as these multi-agent systems become more autonomous, questions of Decentralised Governance become critical. How is the DLT network itself managed, and more importantly, how are disputes, errors, or harmful yet validly signed information handled in a system with no central authority? The study of Decentralised Autonomous Organisations (DAOs) offers a starting point for this inquiry (Ante et al., 2021). Future research should focus on designing and testing robust on-chain and off-chain governance models to ensure these autonomous ecosystems operate safely, fairly, and predictably.

Conclusion

This research project began with a significant, often overlooked challenge in collaborative artificial intelligence: the lack of a fundamental mechanism for trustworthy shared memory in multi-agent systems. The main goal was to go beyond theoretical ideas by developing a proof-of-concept DLT-based architecture and, more critically, to quantitatively assess its practicality in the real world. By systematically evaluating its performance, cost-efficiency, and scalability, this study aimed to provide a clear, data-backed answer to whether a blockchain could act as a reliable 'source of truth' for AI agents.

The results of the experiment, conducted across 45,000 transactions, were clear-cut. The architecture proved to be highly dependable, achieving 100% Provenance Verification Accuracy and showing a straightforward, linear increase in economic costs. However, the most important contribution of this research is the clear demonstration of a key performance trade-off. The system's on-chain write speed was identified as a significant bottleneck, with a low throughput of 10-20 transactions per second, indicating that the architecture is unsuitable for high-frequency data logging. In contrast, off-chain verification was extremely fast, taking less than 3 milliseconds, proving that agents can trust ledger information in near real-time.

This main finding that the architecture sacrifices on-chain write speed for near-instantaneous off-chain trust directly supports the research hypothesis that the system is feasible, while also pinpointing where that feasibility lies. The architecture is highly suitable for applications where data integrity and immutable provenance are critical, and update frequency is low, such as recording important decisions or foundational knowledge.

While limited by its use of a controlled local testnet and a focus on legitimate transactions, these boundaries clearly outline the most promising directions for future research. The logical next steps include deploying this architecture as a Layer 2 scaling solution to directly address throughput limitations and integrating it with intelligent LLM agents collaborating on a task to test its practical utility. Ultimately, this thesis offers a validated, quantitative baseline for the field. It effectively illustrates both the potential and the inherent constraints of using DLTs for multi-agent memory, thereby laying a solid, data-driven foundation for the future development of more complex, scalable, and genuinely collaborative AI systems.

References

Association for Computing Machinery (ACM). (2018). *ACM code of ethics and professional conduct*. ACM. https://www.acm.org/code-of-ethics

Benet, J. (2014). IPFS - Content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561*.

Buterin, V. (2021, April 2). The scalability trilemma. Vitalik Buterin's website. https://vitalik.ca

Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*.

Chen, T., Li, Z., & Zhou, Y. (2017). A systematic study of gas-costly patterns in smart contracts. In *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)* (pp. 189–199).

Chen, J. C.-Y., Saha, S., Stengel-Eskin, E., & Bansal, M. (2024). Magdi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. *Proceedings of the 41st International Conference on Machine Learning*.

Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V. (2016) Blockchain Technology: Beyond Bitcoin. *Applied Innovation Review*, 2, 71.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., & Juels, A. (2020). Flash Boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 942–960). IEEE.

Danger, T., & et al. (n.d.). *python-ecdsa* [Computer software]. GitHub. Retrieved October 26, 2025, from https://github.com/tls-in-a-box/python-ecdsa

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017). BLOCKBENCH: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data* (pp. 1085–1100).

Ethereum Foundation. (n.d.). Sepolia testnet. *Ethereum.org*. Retrieved October 26, 2025, from https://github.com/eth-clients/sepolia

Ethereum Foundation. (n.d.). *Solidity documentation*. Soliditylang.org. Retrieved October 26, 2025, from https://docs.soliditylang.org/

Floridi, L., & Cowls, J. (2019). A unified framework of five principles for AI in society. *Harvard Data Science Review*, *1*(1). https://doi.org/10.1162/99608f92.8cd550d1

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey *[Preprint]. arXiv*. https://arxiv.org/abs/2312.10997

Gencer, A. E., Basu, S., Eyal, I., van Renesse, R., & Sirer, E. G. (2018). Decentralization in Bitcoin and Ethereum networks. In *Financial Cryptography and Data Security* (pp. 439–457).

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. https://arxiv.org/abs/1410.5401

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. https://arxiv.org/abs/2402.01680

Hagstrom, L., Saynova, D., Norlund, T., Johansson, M., & Johansson, R. (2023, December). The effect of scaling, retrieval augmentation and form on the factual consistency of language models. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 5457–5476). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.332

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly,* 28(1), 75–105.

Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S. M., & Felten, E. W. (2018). Arbitrum: Scalable, private smart contracts. In *27th USENIX Security Symposium (USENIX Security 18)* (pp. 1353–1370). USENIX Association.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., & Raffel, C. (2023, 23–29 Jul). Large language models struggle to learn long-tail knowledge. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 15696–15707, Vol. 202). PMLR.

Khosla, S., Zhu, Z., & He, Y. (2023). *Survey on memory-augmented neural networks: Cognitive insights to ai applications.* https://arxiv.org/abs/2312.06141

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM, 21(7),* 558–565.

Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems*, *16*(2), 133–169.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, *4*(3), 382–401

Law, A. M. (2015). *Simulation modeling and analysis* (5th ed.). McGraw-Hill Education.

Leonardos, S., Reijsbergen, D., & Piliouras, G. (2022). Dynamical analysis of the EIP-1559 Ethereum fee market. *ACM Transactions on Economics and Computation*, *10*(4), 1–40.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., K¨uttler, H., Lewis, M., Yih, W.-t., Rockt¨aschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems, 33*, 9459-9474.

Manzoor, M. A., Albarri, S., Xian, Z., Meng, Z., Nakov, P., & Liang, S. (2023). Multimodality representation learning: A survey on evolution, pretraining and its applications. *ACM Trans. Multimedia Comput. Commun. Appl., 20(3)*.

Matthias, A. (2004). The responsibility gap: Ascribing responsibility for the actions of autonomous systems. *Ethics and Information Technology*, *6*(3), 175–183.

Montgomery, D. C. (2019). *Design and analysis of experiments* (10th ed.). Wiley.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* Retrieved from https://bitcoin.org/bitcoin.pdf

Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)* (pp. 305–319). USENIX Association.

OpenAI. (2023). *GPT-4 Technical Report*. arXiv preprint arXiv:2303.08774. https://arxiv.org/abs/2303.08774

Paradigm. (2023). *Anvil (Version v1.2.1)* [Computer software]. Foundry. https://book.getfoundry.sh/anvil/

Park, J. S., O' Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3586183.3606763

Petroni, F., Rockt¨aschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019, November). Language models as knowledge bases? In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 2463–2473). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1250

Poon, J., & Dryja, T. (2016). The Bitcoin lightning network: Scalable off-chain instant payments. Retrieved from https://lightning.network/lightning-network-paper.pdf

Python Software Foundation. (2025). *asyncio (Version 3.13.3)* [Computer software]. Python. https://docs.python.org/3/library/asyncio.html

Python Software Foundation. (n.d.). *hashlib — Secure hashes and message digests*. In *The Python Standard Library*. Retrieved October 26, 2025, from https://docs.python.org/3/library/hashlib.html

Rae, J.W., Potapenko, A., Jayakumar, S. M., & Lillicrap, T. P. (2019). *Compressive transformers for long-range sequence modelling.* CoRR, abs/1911.05507. http://arxiv.org/abs/1911.05507

Ross, J. (2025). *Remembering at scale: Memory mechanisms for large language model–based agents* [Unpublished literature review]. James Cook University.

Sabater-Mir, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, *24*(1), 33–60.

Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy* (pp. 459–474). IEEE.

Sporny, M., Kellogg, G., & Lanthaler, M. (Eds.). (2020, July 16). *JSON-LD 1.1*. W3C Recommendation. https://www.w3.org/TR/json-ld11/

Swan, M. (2015). *Blockchain: Blueprint for a new economy.* O'Reilly Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need.* In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Wang, L., Zhang, X., Su, H., & Zhu, J. (2024). A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(8),* 5362–5383.

Wang, S., Zhu, Y., Liu, H., Zheng, Z., Chen, C., & Li, J. (2024). Knowledge editing for large language models: A survey. *ACM Computing Surveys, 57(3)*, 1-36..

Wiesinger, J., Marlow, P., & Vuskovic, V. (2024, November). *Agents* (White paper) (Retrieved from the Internet Archive). Google. Retrieved May 4, 2025, from https://archive.org/details/google-ai-agents-whitepaper

Wood, G. (2022). *Ethereum: A secure decentralised generalised transaction ledger* (Berlin Version e101a88). Ethereum Foundation. https://ethereum.github.io/yellowpaper/paper.pdf

Xu, X., Weber, I., & Staples, M. (2019). *Architecture for Blockchain Applications.* Springer.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—A systematic review. *PLOS ONE, 11(10)*, e0163477.

Zhang, R., Wu, Q., & Wang, P. (2020). A survey on blockchain-based privacy-preserving techniques. *Computer Networks*, *181*, 107455.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2018). An overview of blockchain technology: Architecture, consensus, and future trends. *2017 IEEE International Congress on Big Data (BigData Congress)*, 557–564.

Zyskind, G., Nathan, O., & Pentland, A. (2015). Decentralizing privacy: Using blockchain to protect personal data. *2015 IEEE Security and Privacy Workshops*, 180–184.