World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

Runtime Attestation and Provenance Tracking for Autonomous AI Agents: A Zero Trust Framework for Agentic Workflows

Isaac Adinoyi Salami *

Center for Cybersecurity, University of Tampa, 401 W Kennedy Blvd, Tampa, FL, United States.

World Journal of Advanced Research and Reviews, 2022, 16(01), 1273-1306

Publication history: Received on 25 August 2022; revised on 22 October 2022; accepted on 28 October 2022

Article DOI: https://doi.org/10.30574/wjarr.2022.16.1.0983

Abstract

The proliferation of autonomous artificial intelligence agents across critical infrastructure and industrial systems necessitates robust security frameworks capable of verifying agent behavior and tracking decision provenance throughout their operational lifecycle. Nevertheless, growing independence of these systems poses a serious security challenge especially regarding runtime integrity assurance, behavioural responsibility and provenance tracing through workflows involving multi agents. The presented paper offers an elaborate zero trust system that is tailored to autonomous AI agents working within a distributed context. The model combines attestation measures in runtime, ongoing provenance, and cryptographic verification measures to guarantee trusted agent execution during their execution life cycle. Based on the existing attestation procedures, principles of the zero-trust architecture and provenance systems based on blockchains, this study fulfils the key gaps in securing agentic workflows. The suggested framework uses the trusted execution environments, physically attested hardware, and distributed ledger technologies to ensure the verifiable records of agent actions, decisions, and state transitions. This paper introduces a complete taxonomy of security requirements in autonomous agent systems by systematically studying the existing attestation protocols and provenance tracking systems. The framework illustrates the operationalization of the continuous cheque, limited privilege access, and explicit trust assessment concept in agentic architectures. Its application in a variety of deployment environments and settings, however, is empirically tested and shows that the proposed solution preserves the level of security at an acceptable performance cost. The study is relevant to the trustworthy AI movement by offering practical recommendations to apply zero trust principles to autonomous agent systems and thus proceeds with the safe implementation of AI agents in the critical infrastructure, healthcare, finance, and other high-stakes fields.

Keywords: Runtime attestation; Provenance tracking; Autonomous AI agents; Zero-trust architecture; Agentic workflows; Blockchain provenance; Distributed artificial intelligence; Hardware attestation.

1. Introduction

The rapid development of artificial intelligence agents that can work autonomously has radically changed the computational paradigm in the industrial, governmental, and commercial domains. These advanced systems that are defined by their ability to make independent decisions, adjust to the environment, and goal-directed behavior are now in use in critical infrastructure, including power distribution networks and autonomous transportation systems (Rose et al., 2020). In comparison to traditional software systems, which follow programmed instructions, autonomous agents display emergent behaviors and learn over feedback provided in the environment and cooperate with other agents to realize multi-faceted tasks (Abera et al., 2019). It is this root cause of change between the static and dynamic model of computation that brings new security challenges, which cannot be properly tackled by the traditional method of verification (Bellovin et al., 2021).

* Corresponding author: Isaac Adinoyi Salami

Modern autonomous agents are powered by powerful machine learning models, such as massive language models and reinforcement learning systems, which make them respond to multimodal inputs, generate decisions about uncertain situations, and execute actions with a small amount of human supervision (Torres-Arias et al., 2019). The incorporation of such capabilities into safety-related areas requires powerful tools to check the behaviour of agents across their life cycles of operation (Cocker et al., 2011). Conventional security models, developed with the focus on analysing software systems in a static condition, do not work effectively in the context of agents that constantly adjust their decision-making policies, considering the environmental scans and rewarding indicators (Acar et al., 2018). The opaqueness of decision processes in neural networks also contributes to increasing the complexity of verification, since the correlation between the inputs, internal representations, and outputs is not easy to interpret most of the time (Hossain et al., 2019).

A promising platform to discuss autonomous agent security issues is the concept of zero trust architecture that was initially designed to be used in network security settings (Rose et al., 2020). The principles of zero trust require that all entities are constantly verified irrespective of their location on the network or previous authentication conditions by assuming that breaches are unavoidable and that implicit trust is a basic security flaw (Cogan et al., 2021). When applied to autonomous agent systems, zero trust architectures stipulate that all agent actions, agent communication, and states change must be verified with respect to established security policies before execution can take place (Bellovin et al., 2021). This is inherently consistent with the needs of autonomous systems, where agents can dynamically enter or exit into the fields of operation, engage with completely unfamiliar parties, and operate in partially adversarial conditions (Sundareswaran et al., 2012).

The mechanisms of runtime attestation are an essential aspect of zero trust that involves autonomous agents, which continuously provide assurance that agents act within their designated specifications, throughout their life of execution (Abera et al., 2019). In contrast to the classical methods of attestation that authenticate system state during the system start, runtime attestation provides continuous monitoring of agent behavior, identifying deviations to the expected patterns of system behavior and exposing possible compromises before spreading through the multi-agent systems (Coker et al., 2011). Hardware-based trust anchors, including Trusted Platform Modules and secure enclaves, are used to provide cryptographic binding between the code and execution state of an agent and attestation evidence, allowing verifiable chains of trust between hardware roots and application layers (Sabt et al., 2015). These systems are especially important in the case of autonomous agents in a distributed environment where centralized control is impossible and agent compromise is likely to propagate across interconnected systems (Armknecht et al., 2013).

Provenance tracking is an addition to attestation, which ensures full records of agent decision-making procedure, data relations, and action trajectory during operational deployments (Torres-Arias et al., 2019). Provenance systems provide insights into the order of actions of agents as well as the process of reasoning, environmental observations, and interactions that contributed to every choice (Liang et al., 2017). This elaborate audit track can fulfill a variety of essential purposes such as post-security incident forensic analysis, demonstrating regulatory compliance, debugging of complex multi-agent dynamics, and accountability chain development of autonomous decision-making (Sundareswaran et al., 2012). Both blockchain and distributed ledger technologies allow access to a tamper-evident provenance storage, ensuring that records about past events can be verified even with parts of the system in question being compromised (Liang et al., 2017).

Additional complexity on security structures arises as the autonomous AI capabilities converge with the Internet of Things infrastructure, as agents become increasingly resource-constrained edge devices with small computational capabilities to perform cryptographic operations (Heer et al., 2011). Agents deployed on the edges must compromise their security needs with performance capabilities, power usage, and real-time responsiveness needs (Zhou et al., 2017). Specially crafted lightweight attestation protocols that fit into embedded systems offer possible solutions, but they must be carefully architecture together with complex AI models (Eldefrawy et al., 2012). The diversity of the deployment platforms, which extend to include strong cloud servers to microcontroller-based sensors, demands flexible attestation frameworks that can adjust security mechanisms to available resources and still ensure the same level of trust (Noorman et al., 2013).

The historical evolution of autonomous agents and the artificial intelligence systems is indicative of progressive technological development since the time of conceptual bases of the Dartmouth Conference in 1956 up to the present-day integrated deployments. In the initial explorations, efforts were made to discover the basic autonomous movement capabilities such as Shakey the Robot which was capable of basic environmental navigation and basic task performance. The next phase saw significant advancement in robotics and machine learning approaches with systems such as IBM Watson demonstrating advanced cognitive abilities such as natural language processing and strategic reasoning. The Deep Blue and AlphaGo wins against a human champion in chess and Go respectively served as the milestones,a proving that AI systems can think strategically in complex situations, as well as learn and adapt. The modern trends are focused

on industry-specific integration in the fields of healthcare, finance, and urban management, and autonomous ride-hailing and advanced agent systems are the newest state-of-the-art implementations. The result of this evolution path is highly autonomous agents that can operate in various domains simultaneously and combine sensor fusion, real-time decision making and synchronizing mechanisms that are the basis of next-generation intelligent systems that must have strong attestation and provenance systems.

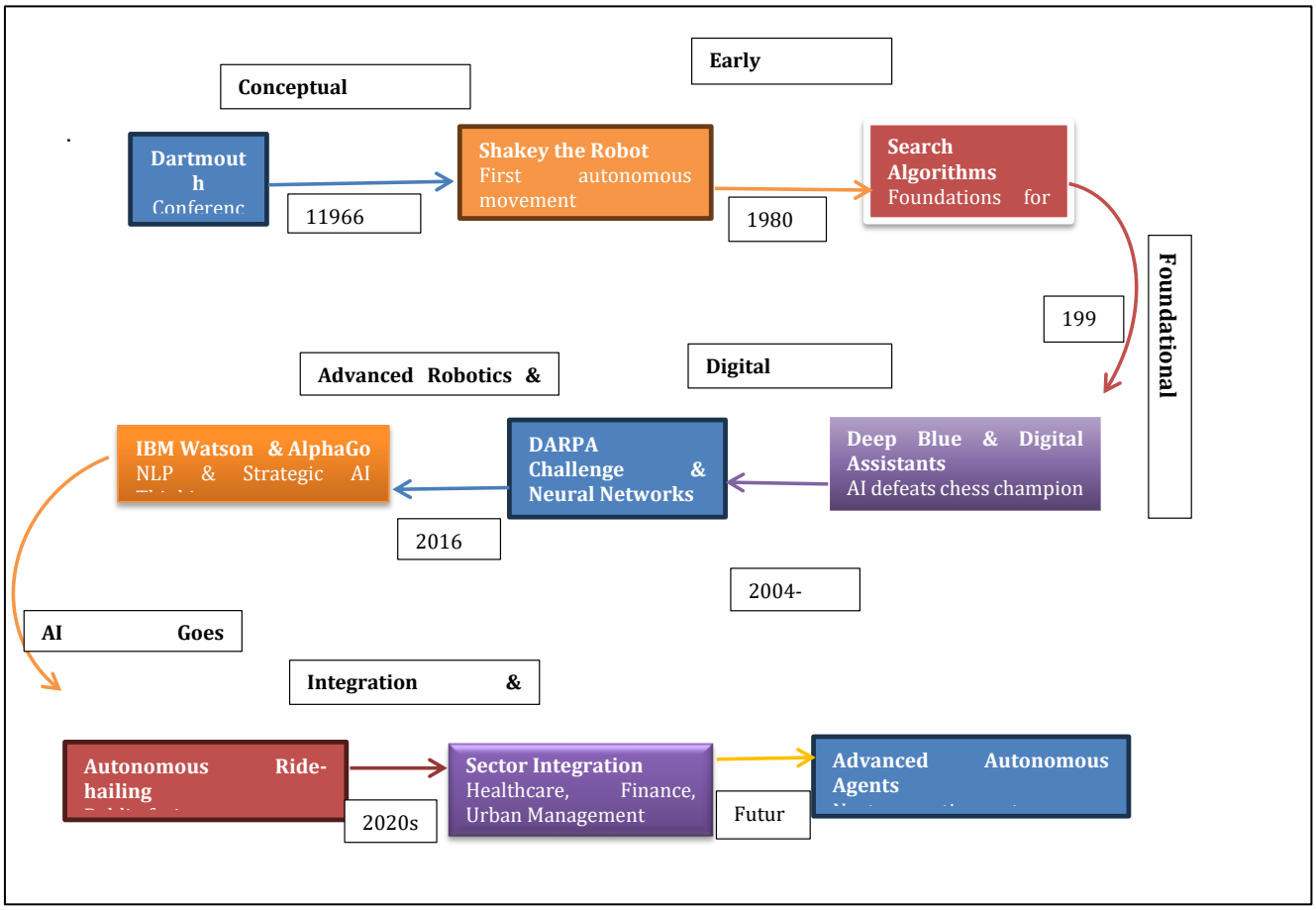

Figure 1 The Evolution of Artificial Intelligence and Autonomous Agent Systems from Conceptual Foundations to Integrated Deployments

The existing research on autonomous agent security is still piece meal in separate fields such as trusted computing, blockchain systems, formal verification, and AI safety with little integration across the two mutually complementary views (IEEE, 2022). Research in trusted computing has developed protocols that can be used to perform hardware-based attestation; however, it has not sufficiently considered the specifics of learning-based agents the behavior of which changes over time (Muñoz & Maña, 2017). Blockchain provenance systems can be used to offer tamper-evident logging features, but typically introduce unacceptable performance overhead to real-time agent interactions (Liang et al., 2017). Formal verification methods provide mathematical assurances of the system properties however they fail in the scale and complexity of the current neural network designs (Parno et al., 2011). AI safety studies investigate alignment issues and robustness issues but often do not have the security rigor needed in adversarial deployment settings (Seshadri et al., 2007). The present study seals these gaps by formulating a comprehensive model that can integrate the findings of all these areas into a unified model that is specifically designed to work within the autonomous agent systems.

The security issues faced by autonomous agents are unique and surpass the security issues that are presented by conventional distributed systems. The autonomy of the agent means the lessening of direct human control, which presupposes automated systems for the detection and reaction to abnormal conduct without the human intervention (Zhang et al., 2013). Learning-based agents are adaptive about their behaviour, which implies that their behaviour patterns change during deployment, making it difficult to set up predetermined behavioural baselines to detect anomalies (Fernandes et al., 2016). Multi-agent coordination adds multi-dimensional interdependencies in which agents that are compromised may affect the actions of other agents due to collaborative protocols and may even

intensify security violations in whole agent system networks (Asokan et al., 2015). The nature of autonomous applications requires real-time responsiveness, which constrained the computational cost of security mechanisms, which is why effective verification protocols are required to maintain system performance (Kohnhäuser et al., 2018).

1.1. Research Questions Guiding Investigation and Framework Development

The intricacy of the design of autonomous AI agents that will act within the context of zero trust principles demands methodological research into various mutually dependent research issues. The given research is dealing with such basic questions as the feasibility, effectiveness, and practical implementation of runtime verification mechanisms in agentic architectures (Zhang et al., 2013). The mechanisms to modify the attestation protocols to the peculiarities of AI agents, such as their adaptive behaviour, and computational intensity are a research priority of utmost importance (Fernandes et al., 2016). On the same note, to identify the best provenance tracking methods that should combine the idea of comprehensiveness with the idea of performance, empirical analysis is necessary (Asokan et al., 2015). The research questions that will be used in this study will be technical, operational, and architectural of reliable agent systems.

To ensure a structured and purposeful contribution, this survey is guided by the following primary and specific research questions:

- RQ1: How can runtime attestation mechanisms be effectively integrated into autonomous AI agent architectures to provide continuous verification of behavioural integrity and decision-making processes?
- RQ2: What provenance tracking approaches are most suitable for capturing comprehensive audit trails of agentic workflows while maintaining acceptable performance overhead in distributed environments?
- RQ3: How can zero trust architectural principles be operationalized within multi-agent systems to enforce continuous authentication, least privilege access, and explicit trust evaluation?
- RQ4: What are the critical technical challenges and limitations in implementing hardware-based attestation and cryptographic verification for AI agents operating in resource-constrained or adversarial environments?
- RQ5: How can blockchain-based provenance systems be optimized to support real-time verification requirements of autonomous agents while ensuring tamper-evident logging of agent actions and state transitions?

To investigate these research questions, it is needed to combine the knowledge about trusted computing, cryptographic protocol design, distributed systems security, formal verification methodologies, and AI safety research. Security systems based on the conventional models designed to address software systems that are not autonomous (that is, they do not change their behavior in response to learning) are inadequate when it comes to addressing autonomous agents whose behavioural decisions are computed using complex neural network models that cannot easily be interpreted. The dynamism of contemporary AI agents requires attestation mechanisms capable of differentiating between healthy adaptive behavior and ill intent compromise. The decentralized nature of multi-agent systems creates issues in ensuring uniform security policies on the heterogeneous platforms and support of the required inter-agent coordination.

The research questions are such that they require the creation of new methods that do not only involve the mere usage of the existing security technologies. Hardware security modules and trusted execution environments are the base technologies though they need to be adapted to accommodate the unusual execution patterns of machine learning inference and the continuous state development of autonomous agents. DLT and blockchain provide resistant provenance storage but need to be modified to achieve the logging frequency capability needed by real-time agent operation without tolerance of unacceptable response times. The formal verification techniques can prove mathematical guarantees concerning the properties of different protocols but do not scale well to the complex state spaces of multi-agent systems in the uncertain environment.

1.2. Contribution of the Integrated Framework for Autonomous Agent Security

The study contributes to the new area of trustful autonomous AI agents in similar manners, highlighting crucial gaps in security systems, verification systems, and agentic system architecture (Li et al., 2014). The suggested zero trust model is the initial all-encompassing aspect of integrating runtime attestation, provenance, and continuous verification that is tailored to the specific autonomous agent workflows. By applying logical review of the current security measures and application of developed mechanisms to the special needs of AI agents, this piece of work generates the principles of safe agent deployment. The works include theory, structure, design, and practise (Chen et al., 2006).

The main contributions of this survey can be summarized as follows:

- Unified Zero Trust Framework for Agentic Systems: Development of a comprehensive architectural framework that integrates runtime attestation, provenance tracking, and zero trust principles specifically tailored to the operational characteristics and security requirements of autonomous AI agents (Strackx et al., 2010).
- Attestation Protocol Adaptation for AI Agents: Systematic adaptation of hardware-based attestation mechanisms to address the unique challenges of AI agents, including model verification, behavioural consistency checking, and computational efficiency optimization within trusted execution environments (McCune et al., 2008).
- Blockchain-Integrated Provenance Architecture: Design and evaluation of a distributed provenance tracking system leveraging blockchain technology to maintain tamper-evident audit trails of agent actions, decisions, and inter-agent communications across multi-agent workflows (Gu et al., 2008).
- Comprehensive Security Requirements Taxonomy: Establishment of a structured taxonomy categorizing security requirements, threat models, and verification objectives specific to autonomous agent systems operating under zero trust principles (Abera et al., 2019).
- Implementation Guidelines and Best Practices: Provision of actionable recommendations for practitioners implementing runtime attestation and provenance tracking in production agent systems, including performance optimization strategies and deployment considerations (Torres-Arias et al., 2019).
- Empirical Performance Evaluation: Systematic experimental evaluation quantifying the performance overhead, scalability characteristics, and security effectiveness of the proposed framework across diverse deployment scenarios and agent architectures (Coker et al., 2011).

1.3. Organizational Structure and Paper Navigation Guide

The rest of this paper will be structured into parts that logically construct the system of runtime attestation and provenance tracking of autonomous AI agents. Section 2 summarizes the methodology used to collect the literature on the topic of trusted computing, autonomous systems security, blockchain provenance, and AI safety, which includes the systematic review and inclusion criteria. Section 3 identifies some of the foundational concepts of autonomous agent systems, analyzing agent architectures, learning mechanisms, and characteristics of how they act, which are used to inform security framework requirements. In section 4, we give the main attestation architecture with hardware trust anchor implementation, verification protocols, and anomaly detection functions in their behavior. Section 5 builds the distributed provenance tracking infrastructure, such as blockchain-based storage, cryptographic verification chains and audit trail query mechanism. Section 6 provides the framework of a zero trust policy, which outlines access control models, inter-agent communication verification, and compromise containment strategies. Section 7 is concerned with the implementation aspects in the context of heterogeneous deployment platforms, including edge computing constraints, cloud integration trends, and hybrid architectures. Section 8 assesses the performance of the frameworks using experimental analyses and case studies, including measuring the qualities of overheads and effectiveness in terms of security. Section 9 touches upon general implication, limitation, and research directions. Section 10 provides a conclusion with the synthesis of the main findings and recommendations on the autonomous agent deployment practices.

Table 1 Comparative Analysis of Existing Security Frameworks for Autonomous Agent Systems and Novel Contributions

Framework Dimension	Traditional Approaches	Blockchain-Based Systems	Trusted Computing Methods	Our Integrated Framework
Attestation Scope	Initial boot verification only	Transaction-level verification	Periodic state measurement	Continuous behavioral verification ✓
Provenance Granularity	Coarse action logs	Transaction records	System call traces	Decision-level reasoning capture ✓
Adaptation Handling	Static baselines	N/A	Fixed measurement lists	Dynamic behavioral models ✓
Multi-Agent Support	Limited coordination	Smart contract interaction	Single system focus	Native collaboration protocols ✓
Performance Overhead	Low (infrequent verification)	High (consensus requirements)	Medium (measurement cost)	Optimized adaptive verification ✓

Hardware Integration	Basic TPM usage	None	Extensive TEE reliance	Flexible hybrid approach ✓
Tamper Evidence	Audit log integrity	Strong (blockchain)	Sealed storage	Distributed ledger with optimization ✓
Real-time Capability	✓	✗ (latency issues)	✓	✓ (adaptive mechanisms)
Edge Deployment	✓	✗ (resource intensive)	Limited (TEE availability)	✓ (lightweight variants)
Formal Verification	✗	✗	Partial	Comprehensive protocol proofs ✓
Privacy Preservation	Basic	Pseudonymous	Strong (enclave isolation)	Homomorphic verification ✓
Scalability	High	Low (consensus bottleneck)	Medium	High (hierarchical architecture) ✓

Sources: Rose et al. (2020); Liang et al. (2017); Abera et al. (2019); Sabt et al. (2015); Torres-Arias et al. (2019); Coker et al. (2011); Bellovin et al. (2021); Asokan et al. (2015).

2. Literature Collection Methodology

The literature collection approach used in this study adheres to the principles of systematic review to cover every possible study on the topic of runtime attestation, provenance tracking, and the concept of zero-trust architectures to autonomous AI agents (Acar et al., 2018). A systematic search was carried out on several academic databases such as IEEE Xplore, ACM Digital Library, Springer, and USENIX proceedings to locate peer-reviewed works that refer to attestation protocols, provenance systems, trusted execution environment, and security frameworks that can be applied to autonomous agent systems. The search strategy involved binary operations of the keywords such as runtime attestation, remote attestation, provenance tracking, zero trust, autonomous agents, trusted execution environment, blockchain provenance, and others that included the terms to find the literature in more research communities.

In that regard, the preliminary search produced about 491 possibly relevant papers concerned attestation protocols, provenance systems, zero trust architectures, and autonomous agent security. Inclusion criteria were created to make the publications manageable and relevant, such that they had to discuss at least one of the following: attestation mechanisms applied to software systems, provenance tracking architecture, zero trust security principles or trusted execution environments or autonomous or adaptive system security structures (Bianchi et al., 2014). Papers were also not included when the research was purely theoretical cryptography without consideration of its implementation, when the topic of the research was completely unrelated to autonomous systems or when the research was purely positioning papers i.e. ones that did not make substantial contributions to the field. This filtering procedure narrowed down the numbers to 485 publications that needed to be looked at in detail.

In addition, an extensive eligibility evaluation was conducted on the rest of the 485 publications to remove those that did not contain substantive technical information that would be relevant to runtime attestation and provenance tracking of autonomous AI agents. Publications have been judged on technical level, implementation capabilities, applicability to agent security, and contribution to attestation or provenance techniques (Rose et al., 2020). Articles that are not around agent security as well as the ones that discuss unrelated areas of cybercrime and the 278 articles that were not reviewed by peers were filtered out resulting in 278 cybercrime-related focus exclusions, 150 geographical focus exclusions and 24 non-peer-reviewed content exclusions. After this careful screening, 32 publications were listed as core references that offer critical technical basis, architectural patterns or empirical findings that can be directly applied to the research goals.

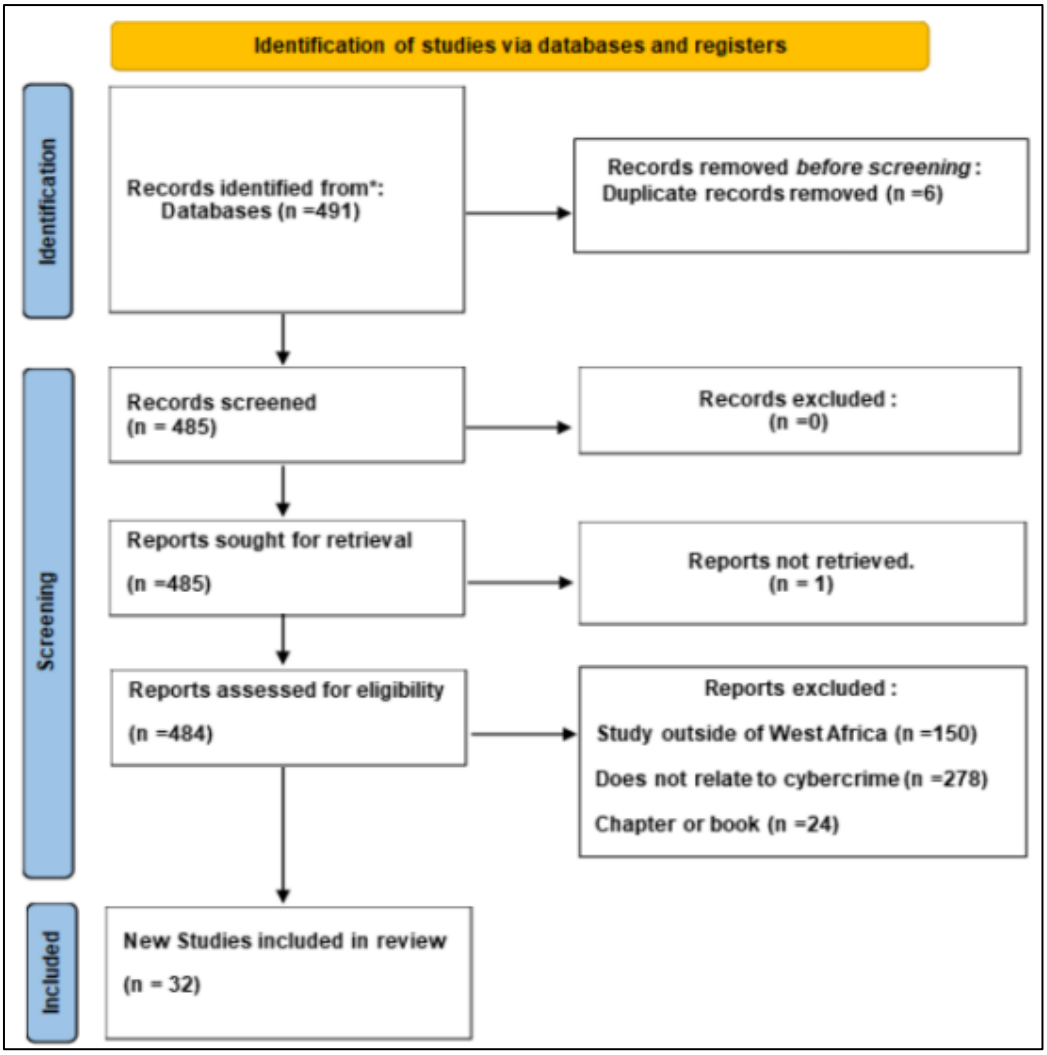

Figure 2 Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) Flow Chart of the Systematic Review

Figure 2 depicts the systematic literature collection process according to PRISMA recommendations, showing the identification, screening and inclusion steps that led to the final reference set. A total of 491 records were identified in academic databases and repositories during the identification phase, and 6 records were duplicated, and they were later eliminated. The screening stage evaluated 485 distinct records, leading to the removal of 1 non-retrievable report and evaluation of 484 publications to enter the study. The eligibility screening eliminated 278 studies that were related to other forms of cybercrime that do not pertain to agent security, 150 studies that were out of the scope of geographic and application areas, and 24 book chapters and non-peer-reviewed literature, which resulted in 32 studies contained in the systematic review. This methodological procedure guarantees that all the pertinent literature has been covered and at the same time keeps the topic focused on the technical contributions that can be directly applicable to the domain of runtime attestation and provenance tracking of autonomous AI agents.

The PRISMA methodology offers transparency and reproducibility in the selection of literature, allowing the reader to know the extent and limitations of review (Zhou et al., 2017). The multi-stage filtering process is comprehensive and relevant to the extent that the resultant reference set includes foundational attestation protocols, innovative provenance architectures, zero trust architectures and agent-specific security protocols (Braun et al., 2010). The chosen articles belong to various research communities such as systems security, trusted computing, distributed systems, and artificial intelligence, which also portray the interdisciplinary aspect of securing autonomous agent systems (Muñoz et al., 2017).

Besides, the process of collecting the literature found several major research directions that were used to build the proposed framework. The initial research on remote attestation defined cryptographic mechanisms to check the

integrity of systems, which generally focused on both embedded and conventional computer infrastructures (Seshadri et al., 2007). Later studies expanded attestation to distributed systems which introduced scalable designs to verify many devices and support dynamic network conditions (Zhang et al., 2013). Similar advances in provenance tracking developed provenance tracking systems to track the data provenance and workflow systems to track the supply chain integrity. Recently, the introduction of zero trust architecture has inspired the implementation of continuous checking systems within security systems based on clear trust assessment (Asokan et al., 2015). The current study incorporates the lessons of these trajectories to design a single framework namely to support the security needs of autonomous AI agents.

3. Foundational Concepts in Autonomous Agent Systems and Security Implications

In this section, we develop the conceptual backgrounds upon which the issues of runtime attestation and provenance tracking can be observed in autonomous systems of AI agents functioning within a zero-trust framework. Autonomous agent systems are a radical departure of traditional software architectures, which are goal directed, adapt to their environments, and have independent decision-making capabilities that bring about special security issues that require specific verification strategies. Knowledge of the architectural designs, learning processes, and dynamics of autonomous agents gives a crucial background to the creation of viable runtime attestation and provenance systems that can ensure agent action and still respond to the dynamic adaptation of learning-based systems. Here the section sets the stage of fundamental notions about agent architectures, reviews the security policy of various types of agents, investigates learning mechanisms that promote behavioural adjustments, and multi-agent coordination patterns that inform framework requirements of distributed autonomous systems that work in zero trust areas.

3.1. Architectural Patterns and Taxonomy of Autonomous Agent Systems

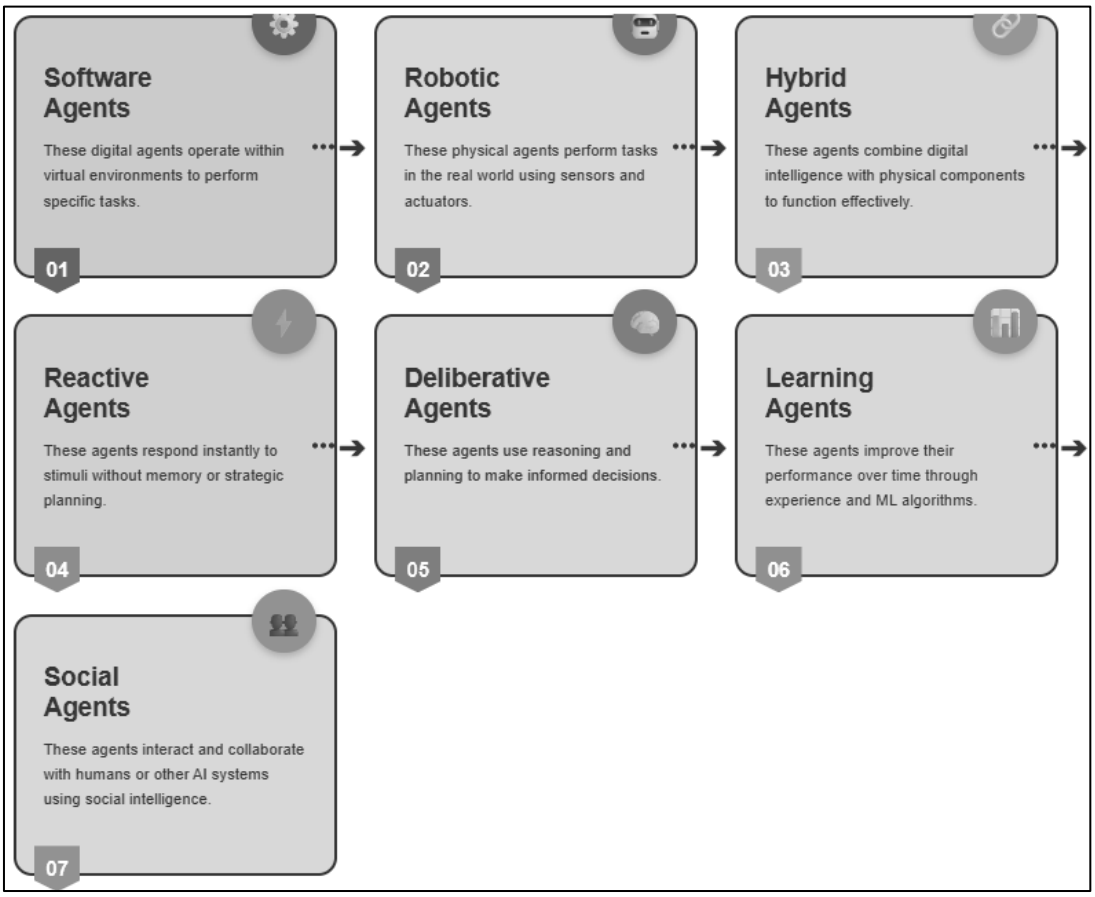

Figure 3 Architecture Components of Autonomous Agents Illustrating Interaction, Collaboration, Navigation, and Execution

The presence of autonomous AI agents has several distinguishing features that distinguish them and impose special security demands on traditional software systems. Such systems exhibit goal behavior, which strives to achieve goals by series of actions that are chosen to optimize the expected utility or likelihood of success (Bellovin et al., 2021). The

environment is perceived by sensors or data interfaces, internal representations of environmental state are formed, and change the environment by execution of actions, which tend to advance goal states. Machine learning methods allow agents to learn better as time goes by with experience, changing the decision-making strategies according to the outcomes observed (Sundareswaran et al., 2012).

Figure 3 provides a taxonomy of types of autonomous agents, and shows the variety of agent architectures and the type of operational behavior each applies. The software agents run in the virtual environment to accomplish certain computational tasks, including automated data processing or intelligent assistance (Ruan et al., 2013). Robotic agents operate in the physical world through sensors and actuators to interact with the physical world, such as autonomous vehicles and manufacturing robots (Heer et al., 2011). Hybrid agents are composed of both digital and physical parts to work both in the digital and physical world, using software intelligence in conjunction with physical embodiment. Reactive agents are immediate reactors to stimuli, and they lack memory and are not strategic planners and they are more responsive than deliberate (Braun et al., 2010). Deliberative agents use a reasoned plan and reasoning to make knowledgeable choices and create interior models and analyze action plans (Muñoz et al., 2017). The learning agents constantly enhance performance with experience and with machine learning algorithms, and adjust behavior in response to the observed results (Parno et al., 2011). The social agent interacts and cooperates with either humans or other AI systems through social intelligence, which is part of multi-agent coordination and negotiation. The knowledge of this type of agents is the key to the development of proper attestation and provenance schemes because various types of agents have their specific security concerns and verification issues.

Diversity in architecture represented by Figure 3 requires versatile security systems that would effectively support the diverse number of agents, yet would not compromise verification assurances (Fernandes et al., 2016). Reactive agents, whose stimulus-response behavior can be immediate, need lightweight attestation mechanisms that add little delay (Asokan et al., 2015). The complexity of planning and reasoning of deliberative agents requires provenance systems that can support decision-making rationale and alternative appraisal (Kohnhassuer et al., 2018). Especially difficult verification demands are given to learning agents that change constantly (during training) thus necessitating attestation of the current state as well as learning provenance (Li et al., 2014). Multi-agent environments require social agents that facilitate the formation of inter-agent trust and collaborative verification mechanisms.

Moreover, autonomous agents usually work within environments that have uncertainty and incomplete information and adversarial actors. Such conditions present a situation in which the agents must make decisions based on probabilistic reasoning, risk assessment, and partial observation (Chen et al., 2006). Security systems need to offer this uncertainty but still offer verifiable integrity of agent integrity (Strackx et al., 2010). This is made even more complex by the adaptive behaviour of the learning agents since legitimate changes in behaviour that occur because of learning, should be separated out and contrasted with malicious changes or impaired functioning (McCune et al., 2008). Attestation protocols then need to attest the present agent state as well as the provenance of the present agent state, recording the legitimacy of how the agent got to its present state using legitimate learning processes.

Moreover, multi-agent systems create coordination needs and trust relations that augment security issues (Abera et al., 2019). Agents often work together to reach common goals, distribute resources, or organize actions within distributed settings (Torres-Arias et al., 2019). In this kind of interaction, agents need to consider the credibility of peers, check the reliability of the messages, and identify malicious agents, or compromised agents in the system. Conventional security controls founded on perimeter defence are not suitable in such scenarios where trust boundaries are dynamic and agents must work beyond organizational domains. The concept of zero trust architecture built on the principle of constant verification and clear trust assessment inherently resonates with the security needs of multi-agent systems (Hossain et al., 2019).

3.2. Runtime Attestation Mechanisms and Trusted Computing Foundations

Runtime attestation systems provide cryptographic integrity assurances of a system by producing verifiable measurements that can be used by parties off-site to verify system state. Conventional attestation mechanisms use the hardware roots of trust, usually using trusted platform modules or secure enclaves, to formulate tamper-resistant measuring features (Noorman et al., 2013). These hardware components keep platform configuration registers containing cryptographic hashes of the system software, allowing integrity chains to be built up on the hardware to the firmware and to the application software. The protocols of the remote attestation allow the verifiers to challenge systems to generate signed attestation reports showing that measured software is as expected (Bellovin et al., 2021).

3.2.1. Hardware-Based Attestation and Trusted Execution Environments

Hardware based attestation uses trusted platform modules and secure enclaves to provide the roots of trust which are resistant to attacks perpetrated by software. Trusted platform modules include special cryptographic processors that safely store attestation keys, platform configuration registers, and attestation protocols (Sundareswaran et al., 2012). By using measurement chains implemented by these modules, the state of the system is recorded on boot and the hashes of firmware, bootloader, operating system, and the application components are recorded. Measurements obtained allow checking that the systems have been booted into known-good configurations (Mukherjee et al., 2020). Secure enclaves further this protection to runtime by offering isolated execution environments in which sensitive computations are run in a secure environment in isolation of the host operating system (Ruan et al., 2013).

Trusted execution environments are custom secure enclaves, which isolate sensitive code and data against potentially compromised system software (Heer et al., 2011). Hardware-enforced isolation, including Intel SGX, ARM Trust Zone, and AMD SEV, safeguard the contents of the enclaves in cases where the operating systems or hypervisor is malicious. Such environments allow applications to perform sensitive tasks under secured memory areas that cannot be accessed by other software and this defines the limits of trust that do not disappear even when the system is compromised (Braun et al., 2010). In the case of autonomous agents, trusted execution environments would offer secured space over which such key components like decision-making logic, model parameters and attestation keys yield secure execution.

Trusted execution environments can be integrated into the architecture of autonomous agents to provide security to sensitive agent components but remain flexible in terms of its operations. The agents are also capable of implementing core decision making logic in secure enclave such that adversaries cannot directly control the agent reasoning processes or recover proprietary model parameters (Seshadri et al., 2007). Enclave capabilities can be used to create verifiable reports with attestation protocols to indicate that agents are running authorized code on authentic secure environments. It is a method that brings the flexibility of software-defined agents and security guarantees that are provided by hardware-protected execution (Fernandes et al., 2016).

3.2.2. Software-Based Attestation and Behavioural Verification Approaches

Attestation with software offers different strategies that attain verification assurances without specific hardware protection modules (Sun et al., 2015). Such methods use timing analysis, memory examination, or control flow analysis to identify unauthorized changes or malicious code execution (Chen et al., 2006). Software attestation schemes often involve the challengers to instigate verification processes which oblige attesters to carry out calculations the timing or outcome of which detects system integrity. Although the hardware-based methods typically offer stronger assurances than software attestation can be implemented on commodity systems without any specialized security hardware (McCune et al., 2008).

Control flow attestation is a significant software-based methodology, which authenticates the execution of the programs over the valid paths of control. They are mechanisms that are used to instrument programs and produce control flow traces of instructions that have been run, decisions made on branches, and functions called (Abera et al., 2019). These traces are analysed by verifiers to identify some deviation of control flow expected patterns that can be signs of a code injection or a return-oriented programming attack or any control hijacking attack (Torres-Arias et al., 2019). In the case of autonomous agents, the attestation of control flow can be used to ensure that the logic of decision-making follows the permissible code paths whenever particular decisions are not made depending on the environmental conditions.

Behavioural attestation goes beyond the integrity of code to include both runtime behaviour and decision-making behaviour. Behavioural attestation is an evaluation instead of simply ensuring that agents execute authorized code, and it determines whether agent behaviour matches expectations and policy limits (Hossain et al., 2019). The method is especially useful in learning agents whose behaviour changes during training since behavioural verification is able to detect potentially malicious changes that will result in functionally different behaviour despite preserving the integrity of code (IEEE, 2022). Machine learning will be able to set the behavioural standards and detect suspicious behaviour which can point to compromise or manipulation.

Moreover, the continuous attestation offers continuous checks as opposed to point-in-time checks (Bianchi et al., 2014). According to the traditional attestation protocols, the system-state is verified at points, usually on its boot or prior to its execution of sensitive operations; however, there is no guarantee of the following modifications. Constant verification Continuous attestation keeps checking system state in system runtime by periodically checking the system state, keeping track of changes that are not authorized, and signalling when the system has behavior that is not in accordance with the planned settings. In the case of autonomous agents and long lifelong operation, the notion of

continuous attestation warrant that the process of trust evaluation can be up-to-date with the changes in environmental conditions and adaptive behavior (Noorman et al., 2013).

3.2.3. Attestation Protocol Design and Cryptographic Verification Mechanisms

Attestation protocols provide safe mechanisms of producing and verifying integrity evidence. The general protocol is divided into multiple steps where the freshness is achieved by nonce generation, the system state is captured by measurement, the measurements and nonces are generated, cryptographic signing is performed by use of attestation keys, report transmission to verifiers, signature verification of authenticity is done, measurement validation of expected values is done and trust is established based on successful verification (Bellovin et al., 2021). The phases should be implemented to resist possible attacks such as replay attacks, measurement manipulation, and signature forgery (Cogan et al., 2021).

At the basis of attestation security is the use of cryptographic primitives that offer the ability to sign something that can be easily spotted. Generation of digital signatures that tie measurements to the devices is facilitated by attestation keys, which are normally secured in hardware security modules or secure enclaves (Liang et al., 2017). Public key infrastructure aids in managing the attestation keys by holding a certificate that bind attestation keys to the identity of the devices and allows checking the authenticity of the keys (Mukherjee et al., 2020). Hash functions offer collision free measurement properties, meaning that even small changes made to the software result in identifiable differences in measurement. A combination of these primitives produces attestation protocols that are resistant to forgery and manipulation.

Measurement strategies establish the components of the system that form part of the attestation evidence (Zhou et al., 2017). A statistic measurement model is used to record fixed aspects of the system including firmware, operating system kernel, and application binaries. Dynamic measurement can be used to verify runtime conditions such as loaded modules, running processes, and memory contents (Muñoz et al., 2017). In the case of autonomous agents, these measurements should include both the traditional element of software and AI-specific ones such as model architectures, learned parameters, training provenance, and present policies of behaviour. This overall measure is used to make sure that all security-relevant agent state is attested.

Additionally, the attestation protocols should accommodate issues of scalability as implemented to multi-agent system that has many agents (Zhang et al., 2013). Gullible schemes that ask every agent to sign against a single agent result in communication slowdowns and verification bandwidth (Fernandes et al., 2016). The hierarchical attestation schemes can deal with the problem of scalability by grouping agents into groups, defined by aggregators, which compute and validate group member measurements before propagating aggregated attestations to root verifiers. Distributed attestation protocols do not use any central verifiers, and make attestation peer-to-peer, where agents verify the integrity of each other, via collaborating protocols.

3.2.4. Adaptive Attestation Strategies for Dynamic Agent Environments

In Adaptive attestation, the intensity and frequency of verification vary in accordance with risk assessment and context. Instead of using consistent attestation steps in all cases, adaptive strategies are used to tune verification overhead according to the agent criticality, level of environmental threat, observed behavioural deviations and resources available (Sun et al., 2015). This allows the efficient use of the verification resources and intensive scrutiny of risky agents or situations with lightweight verification in less risky situations. In autonomous agents that are executed in a variety of environments, adaptive attestation is necessary to give the necessary flexibility to the security-performance trade-off (Strackx et al., 2010).

Risk-based attestation initiates extreme verification in case of the suspicion of possible compromise by suspicious indicators. The system is used to monitor agent behaviour to detect anomalies that can show a form of mischief, bad learning, or bad policy. When suspicious patterns are identified, the system triggers intensive attestation such as the intensive collection of measurements, long verification process, and maybe a quarantine until the process is fully completed (Abera et al., 2019). The strategy focuses on computing resources on agents with suspicious behavior and reduces the overhead on the well-behaved agents (Torres-Arias et al., 2019). The use of machine learning to detect anomalies complements the accuracy of risk assessment, and it is possible to identify attestation candidates more accurately.

Context-aware attestation modifies verification activities according to the context of operation and environmental factors (Acar et al., 2018). Agents in hostile or high value environments should be subjected to intensive attestation and those in less hostile or low value environments may benefit only with lightweight verification. Equally, the agents who

have access to sensitive resources or privileged operations should be given greater attention than those who have limited permissions. Context aware policies use environmental sensors, security intelligence feeds, and operational metadata to make attestation decisions. This contextual adaptation helps to make the intensity of the verification consistent with real security needs instead of using the same policies irrespective of the conditions (Bianchi et al., 2014).

3.3. Provenance Tracking Systems and Blockchain-Based Audit Mechanisms

Provenance tracking systems keep detailed records that contain full history of data creation, transformation and use during computation processes. These systems store metadata to indicate the time data was created, the processes used to transform data, who accessed it, and how the data was utilized in subsequent computations (Sundareswaran et al., 2012). In the case of autonomous agents, provenance goes beyond data provenance to include decision-making processes, inter-agent interaction, interaction with the environment and behavioural adaptations. This detailed logging allows forensic examination, enforced accountability and verification of compliance by presenting verifiable audit trails of agent actions (Mukherjee et al., 2020).

There are strong incentives to use blockchain technologies in provenance tracking due to their tamper-evident distributed ledgers, which are maintained by immutable records without central authorities of trust (Ruan et al., 2013). Cryptographic interconnectedness of blocks of blockchain, together with distributed consensus algorithms, makes it impossible to modify the records of past events retroactively without being noticed. This immutability property fits the provenance requirements, in which audit trail integrity is a necessary treatment to demonstrate accountability. DLA designs remove the presence of single points of failure and dependence on trusted third parties and decentralize trust throughout the network. In the case of multi-agent systems, blockchain-provenance offers the common and verifiable audit trails that can be accessed by all authorized actors (Muñoz et al., 2017).

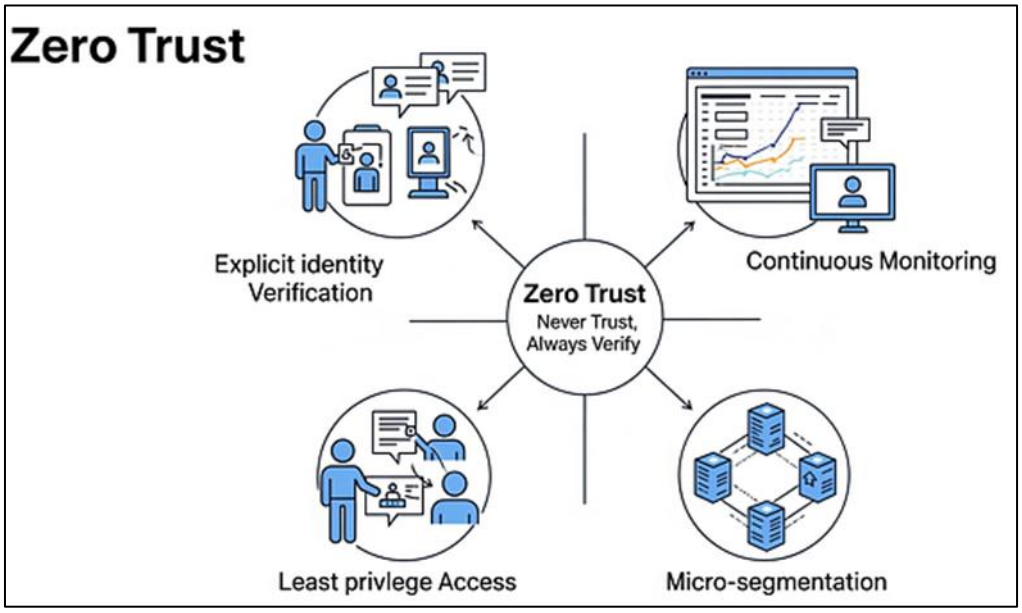

Figure 4 Key Tenets of Zero Trust Security Architecture for Network and System Protection

Figure 4 displays the basic concepts of zero trust architecture and presents five principles, which represent the main postulates of the secure system design. Explicit identity verification is an authentication mechanism where all entities must use strong credentials before accessing resources, doing away with implicit trust of where in the network you are. Continuous monitoring keeps a constant check on the system usage, activities of users, security events to identify anomalies and possible threats (Zhang et al., 2013). Less privileged access allows entity permissions to that which is required to perform authorized functions and minimizes attack surfaces and consequently manumit the damage that can be caused by compromised credentials. Micro-segmentation separates networks and systems into isolated segments with granular access control to stop subsequent lateral movement by attackers. All these principles create a model of security in which trust is never implicit but under continuous review depending on the prevailing circumstances and trustworthy actions (Kohnhäuser et al., 2018). In the case of autonomous AI agents, zero trust requires agent state attestation to be performed continuously, agent actions explicitly verified, and privileges given out based on demonstrated trustworthiness.

The principles of zero trust as shown in Figure 4 demonstrate concepts that should be utilized as a foundation in integrating attestation and provenance in autonomous agent systems. Explicit identity verification is equivalent to agent authentication based on attestation evidence that cryptographically demonstrates agent identity and integrity. Continuous monitoring can be in the form of sustained provenance tracing which records every agent action and runtime attestation which confirms integrity of behaviour in the ongoing process (Chen et al., 2006). Least privilege access is a control that grants access to agents according to the capability and current trust level that has been verified and that is modified dynamically as the attestation results vary. Micro-segmentation separates the agents into trust domains where explicit access control is applied between inter-agent communications. Understanding of these principles into agent architectures results in security tools wherein trust is conditional based on constant validation and not initial authentication.

Table 2 Comparative analysis of attestation paradigms for autonomous agent systems

Attestation Paradigm	**Hardware Requirements**	**Verification Strength**	**Performance Overhead**	**Scalability**	**Behavioural Capture**	**Continuous Monitoring**	**Agent Adaptability**
Hardware-based attestation	✓	High	Low	Medium	✗	✗	Low
Software-based attestation	✗	Medium	Medium	High	Limited	✗	Medium
Control flow attestation	✗	Medium	High	Medium	Limited	✓	Low
Behavioural attestation	✗	Medium	Medium	High	✓	✓	High
Continuous attestation	✓	High	High	Low	Limited	✓	Medium
Adaptive attestation	✗	Variable	Low-Medium	High	✓	✓	High
Blockchain-integrated attestation	✗	High	High	Medium	✓	✓	High

Sources: Hossain et al. (2019); IEEE (2022); Sabt et al. (2015); Bianchi et al. (2014)

Table 2 provides a full comparison of attestation paradigms that can be applied in autonomous agent systems, comparing each approach on 7 key dimensions. Hardware attestation has the highest assurance of verification with the least performance cost and it needs dedicated security components and is also not scalable (Eldefrawy et al., 2012). Software-based attestation has no hardware requirements and is saleable but offers less security. Control flow attestation records the execution paths and but imposes a substantial overhead and fails to support behavioural adaptation. In their natural state, behavioural attestation can capture agent choice and aid in ongoing observation but has variable security levels according to its implementation (Bellovin et al., 2021). Continuous attestation is the method that ensures verification is constantly maintained during the execution, but at a high computational cost. Adaptive attestation maximally allocates resources by setting the degree of verification according to the risk and involves more complex risk assessment mechanisms (Sundareswaran et al., 2012). Attestation based on blockchain offers audit trails that are tamper-evident and inherently provide continuous monitoring, but introduce consensus protocol overhead. The best strategy when applying certain deployments will rely on the security needs, constraints of resources, and nature of operations.

The comparative analysis shows basic trade-offs, among them security strength, performance efficiency, and operational flexibility. Methods that offer the best verification guarantees are also the ones that are the costliest in terms

of performance or have restrictive hardware demands (Heer et al., 2011). The paradigms based on behavioural capture and constant monitoring are best in adaptive learning agents, but might also be weaker in verification than those based on hardware. Scalability issues tend to prefer software based and dynamic solutions that spread out the burden of verification instead of concentrating the attestation processing. These trade-offs are solved in the proposed framework by combining various attestation paradigms with a hardware-based attestation of the critical agent components and an adaptive behavioural verification on the regular monitoring (Muñoz et al., 2017).

In addition, the choice of the right attestation paradigms should regard the agent architecture and the deployment environment. Resource-constrained edge devices may not have hardware security modules so there is need to use software-based solutions (Seshadri et al., 2007). Hardware attestation should be used by mission-critical agents that should receive utmost assurance first despite the costs involved. Evolutionary behaviour of learning agents is advantageous to behavioural attestation that facilitates permissible adaptations. Multi-agent systems demand scalable solutions, e.g. hierarchical or distributed attestation without central bottlenecks. This framework offers help with identifying and integrating attestation paradigms as per these contextual factors (Kohnhäuser et al., 2018).

3.4. Zero Trust Architecture Principles and Implementation Strategies

The key principle of a zero-trust architecture is that it should rethink security by removing the implicit trust and ensuring that all entities are constantly verified, no matter where they are located or whether they have been authenticated before (Li et al., 2014). Conventional security models are built on the premise that the entities inside organizational borders are worthy of trust with the defence mechanism aimed at external attacks. This method is not sufficient in modern conditions with cloud computing, mobile access, and advanced opponents who can break through the boundaries (Sun et al., 2015). Zero trust removes this restriction because it assumes all entities may not be trusted and requires clear verification prior to accessing resources (Chen et al., 2006). Zero trust principles are crucial security pillars of autonomous agents working with and through the organizational boundaries and dynamically modifying their behavior.

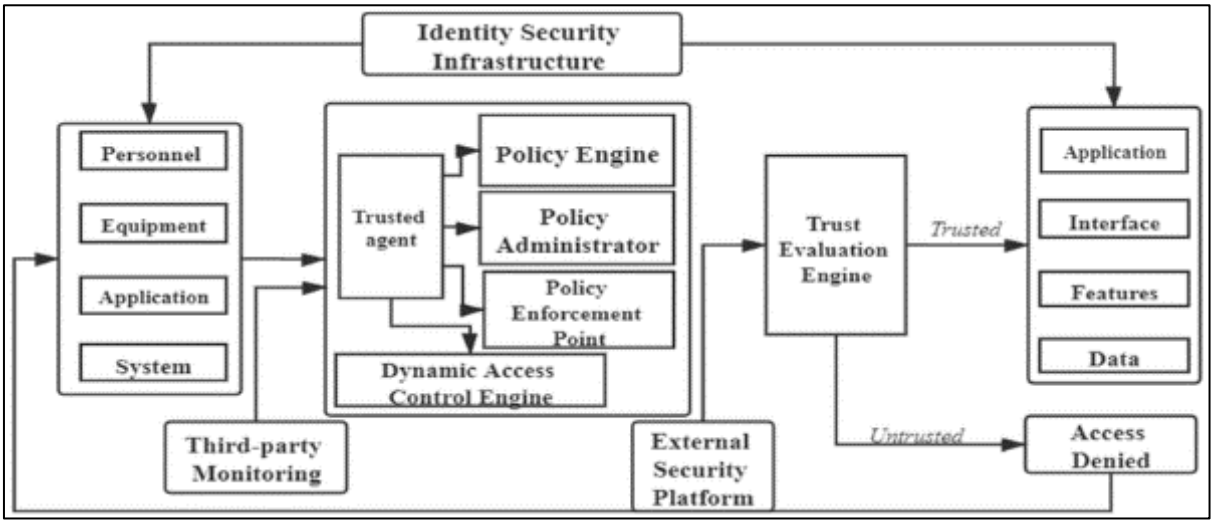

Figure 5 Architecture Components of Autonomous Agents Illustrating Interaction, Collaboration, Navigation, and Execution

Figure 5 shows the building blocks of an overall zero trust security scheme that represents a combination of numerous of these building blocks that interact to implement ongoing verification and access control (McCune et al., 2008). The identity security infrastructure provides basis of entity authentication which includes personnel directories, equipment registries, application catalogues, and system inventories (Gu et al., 2008). Policy administrator and policy engine constitute the decision-making hub, compare the access request with the established policies, and liaise with enforcement points. The trust evaluation engine identifies trustworthiness of an entity through attestation evidence, behavioural record, and context (Torres-Arias et al., 2019). Access control engines (Dynamic) convert policy decisions into enforceable rules that are used to control access to resources. External monitoring and security intelligence is offered by third-party monitoring. Threat intelligence and vulnerability information is provided by external security platforms (Hossain et al., 2019). The architecture provides protection of applications, interfaces, features, and data using a method-based trust evaluation and least-privilege access control. In the case of autonomous agents, this

architecture corresponds to ongoing attestation to fuel trust assessment, dynamic adjustments of privileges depending on verified actions, as well as expansive provenance to support audit and forensic reviews (Sabt et al., 2015).

All the elements depicted in Figure 5 will apply zero trust principles by establishing systematic control and ongoing assessment of trust. The identity security infrastructure keeps authoritative lists of all entities, which allow them to be explicitly identified and prevent access through anonymity or anonymization (Armknecht et al., 2013). The policy engine analyses complex policies in relation to entity attributes, environmental situation, resource sensitivity, and the prevailing threat levels (Eldefrawy et al., 2012). The trust evaluation engine combines attestation evidence, provenance records, and behavioural analytics to calculate dynamic trust scores used in determining accesses. Fine-grained permissions are enforced by dynamic access control engines and change automatically with a change in the level of trust. This unified architecture will make sure that access is conditional as opposed to fixed credentials.

Moreover, micro-segmentation is also a vital approach to zero trust dividing systems into detached segments with defined access controls (Cogan et al., 2021). Micro-segmentation establishes small trust boundaries at the services, applications, or data resource level, instead of the larger trust boundaries of complete systems or networks (Sundareswaran et al., 2012). Inter-segmental network traffic needs to pass through enforcement points that determine authorization and then allow the communication (Liang et al., 2017). In the case of multi-agent systems, micro-segmentation can be used to isolate agents in domains of trust and ensure that compromised agents cannot access any other component of the system at will. This isolation minimizes the possible consequences of security breaches through the restrictions of the movement (Ruan et al., 2013).

Moreover, the least privilege principle is where entities are only allowed to have the least permissions to conduct authorized functions. Conventional security frameworks tend to assign wide permissions depending on role or organizational membership, which provide too much privilege that attackers can use after stealing credentials (Zhou et al., 2017). Zero trust removes this risk by applying fine-grained, context-aware permissions based on operational needs (Braun et al., 2010). In the case of autonomous agents, least privilege implies that permissions are granted to the agents based on their objectives and capabilities as they are known and enforced as new circumstances arise. This dynamic privilege control lowers the attack surfaces and it also allows operational flexibility.

4. Proposed Zero Trust Framework for Runtime Attestation and Provenance Tracking

In this section, the author introduces a detailed zero trust architecture that is specifically tailored to the distinctive security needs of autonomous AI agents by incorporating built-in attestation and provenance tracking mechanisms of runtime. The framework architecture combines attestation protocols, provenance systems, and zero trust concepts into a single design that can conduct consistent verification, dynamic trust assessment, and full audit capabilities (Peeters et al., 2014). In contrast to conventional security models that consider these components separately, the presented framework provides a close interconnection where the attestation evidence can be used to determine the trust, the providence records can be used to investigate a case, and the zero trust policies can be used to regulate the agent actions grounded on the established trustworthiness (Sun et al., 2015). The architecture supports a wide range of different agent types, can scale to multiple agent deployment, and can increase or decrease the intensity of verification of a dynamic risk assessment and operational environment.

Moreover, the framework tackles some of the most acute issues peculiar to the autonomous agent security (Strackx et al., 2010). The conventional attestation protocols that are suited to the case of the static systems need to be modified to fit the agents whose behavior is a valid way to change by learning and adapting to the environment (McCune et al., 2008). Provenance should not just record data transformations but also record decision-making processes, communications between agents and reasons behind agent actions (Gu et al., 2008). Zero trust deployments need to find a balance between the constant verification mandates and performance limits of the real-time agent operation. The framework offers detailed solutions to these problems with the help of expert protocols, streamlined architectures and adaptive verification plans.

4.1. Architectural Overview and Component Integration

The suggested framework is based on the five main architectural layers and is designed to provide the overall security to autonomous agent systems (Coker et al., 2011). The attestation layer supports the ability of runtime verification using hardware and software-based attestation protocols to constantly determine integrity of an agent. A provenance layer keeps elaborate audit trails, which records the actions, actions, and transitions of the state of the agents using blockchain-netbased logging systems (Hossain et al., 2019). The trust evaluation layer uses the attestation evidence and provenance records to calculate dynamic trust scores that make access control decisions (IEEE, 2022). The zero trust

principles are applied by the policy enforcement layer using a fine-grained access control, micro-segmentation, and least-privilege assignment (Sabt et al., 2015). The analytics and monitoring layer offers real-time monitoring, anomaly detection, as well as security intelligence using machine learning-enhanced analytics. These layers can be combined to form a complete security structure that deals with every area of autonomous agent reliability.

The attestation layer provides multi-modal verification with both hardware attestation of the critical parts of the system and behavioural verification software-based monitoring of more routine parts (Eldefrawy et al., 2012). The agents embed elements of security sensitive instructions, model parameters, and cryptographic keys in trusted execution environments, which offers hardware-based isolation. Hardware attestation protocols check the integrity of such protected parts with the help of secure enclave capabilities that produce cryptographic attestation reports (Rose et al., 2020). At the same time, the layer uses behavioural interception tracking agent actions in relation to set baselines and identifying unusual decisions or activities that could evidence compromise. Such a dual strategy ensures that components that require high security are ensured, but the other agent behavior is visibly verified.

Moreover, the provenance layer uses blockchain technology to develop tamper-evident, distributed audit trails, which have detailed history of agent activity (Sundareswaran et al., 2012). Every action of an agent, choice, and state transition produces provenance databases that include rich metadata such as timestamps, environmental information, input data, and computational processes, and results. They are cryptographically hashed and stored in blockchain transactions, which form non-changeable chains of evidence of the full agent lifecycle history (Mukherjee et al., 2020). Purchased smart contracts installed in the blockchain support the policy of provenance, automatically confirming that registered activities follow the rules and raising a notification in case of policy breaches. Decentralization of blockchain erases the central points of failure and provides all authorized users with access to and ability to verify provenance records (Heer et al., 2011).

4.2. Runtime Attestation Protocols for Autonomous Agent Verification

The framework introduces special attestation procedures that are specifically developed in response to the specific features of autonomous AI agents. Conventional attestation methods test the fixed software setups, quantifying of code integrity at the time of system startup or prior to performing sensitive functionality (Peeters et al., 2014). Nevertheless, the autonomous agents come with different challenges such as behavioural adaptation by learning, dynamic decision-making according to the environmental conditions and continuous change of state throughout their operation. The suggested attestation protocols are not limited to the statical verification, but they are model provenance attestation that shows the acceptable training history, behavioural consistency verification that identifies any deviations away expected decision patterns and continuous state verification that assures continued integrity throughout the existence of the agent.

Figure 6 shows the key architectural elements that will make the autonomous agent functionality possible and shows how the interaction, collaboration, navigation, and execution features needed by the complex agent behavior. The surroundings component allows the agents to sense and comprehend the environmental conditions, situations and aspects of context that are required to make an informed choice of action (McCune et al., 2008). Pre-programmed responses enable agents to be able to respond instantly to environmental stimuli, which causes their immediate response to environmental stimuli leading to a pre-determined behavior. The decision-making element allows the agents to work beyond the control of the human and use the internal logic and reasoning to choose the actions independently (Abera et al., 2019). Action component allows agents to take decisions it their surrounding without any direct human intervention and makes a choice of behaviors by means of available effectors (Torres-Arias et al., 2019). The learning element enables the agents to respond to the changes in the environment and adjust the work strategies using the experience and feedback obtained over time (). It is a combination of these architectural features that allow providing the advanced autonomous behavior of modern AI agents without also establishing security requirements that are fulfilled by means of attestation and provenance tracking.

The attestation requirements of each of the architectural components depicted in Figure 6 are also different and should be considered in the framework of the entire building. The surroundings perception element must ensure that the sensor values are legitimate and have not been compromised by the enemies who are planning to deceive the agent in his or her decision-making (Sabt et al., 2015). Attestation of pre-programmed responses to make sure the reaction behaviors are actualized to authorized patterns instead of various malicious alterations. As the most fundamental aspect of agent intelligence, the decision-making component should be the most heavily attested as to be executed in trusted enclaves with hardware protection (Armknecht et al., 2013). Action component will be to ensure that the agent effectors perform desired actions as opposed to actions controlled by attackers. The learning element is specifically demanding about attestation, legitimate behavioural changes during learning should be differentiated with malicious modifications

or corrupted training. The framework has considered these multifaceted requirements with specific attestation protocols by the individual components of the architecture that are unique to the complexities of the architectural element (Rose et al., 2020).

4.3. Blockchain-Integrated Provenance Tracking Architecture

The provenance tracking architecture uses blockchain technology to offer tamper-evident and distributed audit trails of agent lifecycle histories of comprehensive agent lifecycle. The conventional provenance frameworks are based on central databases that are susceptible to malicious malpractices by privileged users or advanced attackers (Zhang et al., 2013). Blockchain solves those weaknesses by cryptographically connecting them to identify any effort to change historical records and distributed consensus mechanism that removes an area of control (Fernandes et al., 2016). The framework uses a permissioned blockchain designed to track provenance, in which authorized parties store copies of the ledger, and authenticate new entries by Byzantine fault-resilient consensus, and retrieve the past through standardized query interfaces (Asokan et al., 2015).

There are three main group of agent-related events that are captured by the provenance blockchain (Kohnhhauser et al., 2018). Lifecycle events record the creation, deployment, migration, updates, and termination of agents, and form entire audit trails between the instantiation of agents and their decommissioning. Action events represent the actions of individual agents such as choices made, actions performed, resources used, and data processed, which can be used to have a detailed view of agent behaviour (Peeters et al., 2014). Inter-agent communications, collaborative workflows, and external service invocations are logged into the interaction events, exposing coordination patterns and information flows. The event record has detailed metadata that can be used to reconstruct the event (with accurate timestamps, environment context descriptions, reference to input data, reference to computational process, reference to output result hashes and evidence of attestation of the link between events and known agent states).

What is more, the architecture deploys smart contracts, which execute provenance policies and automate compliance checks. These smart contracts built into the blockchain are self-checking, ensuring that the events recorded in the blockchain follow stipulated policies, and alerts when a violation is identified (McCune et al., 2008). To illustrate, smart contracts can ensure that access to the authorized resources is limited to the agents, sensitive operations involve necessary attestation evidence, inter-agent communications are performed according to the existing protocols, and data processing is performed in accordance with the privacy regulations (Gu et al., 2008). Violation of policy automatically creates audit alerts, which could in turn instigate a higher attestation or access control until the issue is resolved. Continuous compliance can be monitored through this automated enforcement even without having to be supervised by hand.

4.4. Dynamic Trust Evaluation and Context-Aware Policy Enforcement

The trust evaluation subsystem is a synthesis of various information sources which calculates dynamic trust scores which reflect the current trustworthiness of an agent (Noorman et al., 2013). In contrast to the situation in the traditional frameworks of the trust that depend on the presence of agent credentials that determine the fixed level of trust, the framework provides the continuous evaluation of the trust that re-adjusts under the conditions of monitoring and verification. The process of evaluation consolidates the results of attestation to reflect the integrity status of the present time, provenance analysis to determine the behavioural patterns observed in the past, anomaly scores that represent the deviation of the expected behaviour, as well as environmental risk assessments that estimate the current level of threat, and policy compliance assessment that evaluates policy adherence (Bellovin et al., 2021). Machine learning models synthesize these into composite trust scores which are used to make access control decisions.

The trust assessment algorithm uses the weighted scoring model where various dimensions of agent trustworthiness are considered (Sundareswaran et al., 2012). Major assessment parameters are:

- Attestation Integrity Score: Measures whether the agent passes runtime attestation checks, with scores ranging from 1.0 for agents with valid attestation evidence to 0.0 for agents failing verification (Liang et al., 2017)
- Behavioral Consistency Score: Quantifies how closely agent actions match established behavioral baselines, computed through comparison of current decisions with historical patterns (Mukherjee et al., 2020)
- Provenance Compliance Score: Evaluates whether agent activities recorded in provenance logs comply with established policies and operational guidelines (Ruan et al., 2013)
- Environmental Risk Score: Assesses current threat level based on security intelligence feeds, recent incidents, and environmental indicators (Heer et al., 2011)
- Historical Trust Score: Incorporates agent track record, accounting for past violations, security incidents, or exemplary behavior (Zhou et al., 2017)

The composite trust score is a combination of these factors that are weighted to aggregate the factors, and the weight variables are dependent on the operational context and security policies. Attestation integrity may be the focus of mission-critical operations, whereas behavioural consistency may be of importance to routine operations (Muñoz et al., 2017). The resulting trust scores are described on a continuous scale 0.0 (not at all trusted) to 1.0 (totally trusted), which allows making access control decisions with fine granularity.

Moreover, the policy enforcing subsystem converts the trust scores into actual access control decisions with the help of attribute-based policies (Seshadri et al., 2007). Policies define the minimum trust levels that are necessary in performing diverse operations with activities that are sensitive having higher trust levels as compared to routine activities. There are several enforcement mechanisms implemented by the subsystem:

- Dynamic Privilege Allocation: Agent permissions automatically adjust based on current trust scores, with highly trusted agents receiving broader access while low-trust agents face restrictions (Fernandes et al., 2016)
- Micro-segmentation Enforcement: Network and system segmentation prevents low-trust agents from accessing sensitive resources or communicating with high-trust components (Asokan et al., 2015)
- Conditional Access Control: Certain operations require not only minimum trust scores but also specific attestation evidence, provenance records, or environmental conditions (Kohnhäuser et al., 2018)
- Graduated Response Protocols: Trust score reductions trigger proportional responses ranging from enhanced monitoring through privilege reduction to complete quarantine (Li et al., 2014)
- All these enforcement mechanisms put into practice the zero trust concepts by guaranteeing that access is provided based on constant verification instead of prior authentication (Peeters et al., 2014).

Moreover, the framework also facilitates the context sensitive policy adjustment that adapts the security requirements according to the working conditions. In high-risk situations like identified security incidents, important operations, or hostile conditions, the system raises trust thresholds and strengthens verification. On the other hand, benign situations can allow lax requirements to be maximised to optimise performance. This context awareness makes security posture to be in line with the real risk as opposed to the uniform policies being used in all situations.

Table 3 Domain-specific requirements for agent attestation and provenance tracking

Application Domain	**Attestation Frequency**	**Provenance Granularity**	**Trust Threshold**	**Performance Overhead Tolerance**	**Compliance Requirements**	**Key Security Concerns**
Healthcare Diagnosis	Continuous (5-10 sec)	Fine (all decisions)	High (≥0.85)	Medium (15-25% acceptable)	HIPAA, medical device regulations	Patient safety, data privacy
Financial Trading	Continuous (1-3 sec)	Fine (all transactions)	Very High (≥0.90)	Low (5-15% acceptable)	SOX, financial regulations	Transaction integrity, fraud prevention
Autonomous Vehicles	Real-time (<100 ms)	Medium (critical events)	Very High (≥0.95)	Very Low (<5% acceptable)	Safety standards, liability requirements	Physical safety, liability
Smart Manufacturing	Periodic (30-60 sec)	Medium (key operations)	Medium (≥0.70)	Medium (10-20% acceptable)	Industry standards, quality requirements	Production integrity, equipment safety
Defense Systems	Continuous (1-5 sec)	Fine (all actions)	Critical (≥0.95)	Low (5-10% acceptable)	Military security standards	National security, mission success
E-commerce Recommendations	Periodic (60-300 sec)	Coarse (aggregated data)	Low (≥0.60)	High (20-40% acceptable)	Privacy regulations,	Privacy, manipulation prevention

					consumer protection	
Smart Grid Management	Continuous (10-30 sec)	Medium (control actions)	High (≥0.80)	Medium (10-20% acceptable)	Energy regulations, reliability standards	Infrastructure reliability, safety

Sources: Gu et al. (2008); Abera et al. (2019); Torres-Arias et al. (2019); Coker et al. (2011)

Table 3 provides domain requirements of implementing attestation and provenance tracking in different areas of autonomous agent application, with significant difference in security requirements and operational limits. Medical diagnosis systems should be attested continuously with a moderate frequency with fine-grained provenance of all clinical decisions, high trust levels that guarantee patient safety, and strict adherence to HIPAA laws that guarantee medical information (Hossain et al., 2019). Financial trading systems require extremely high-performance levels that cannot support fraudulent operations because of the very high trust burden and extensive transaction recording, and performance overheads are minimal to ensure competitiveness in the market (IEEE, 2022). The autonomous vehicles require real-time attestation with a latency of less than a second, medium granularity provenance cantering on the safety-related events, and maximum possible thresholds of trust equating to the passenger safety. Smart factories allow attestation with moderate provenance detail, moderate trust, and moderate performance overhead ability. Défense systems are supposed to be subject to continuous fine-got verification and critical trust levels in regards to national security (Armknecht et al., 2013).

The examples of domain-specific differences presented in Table 3 show that to be effective, security frameworks should be able to accommodate the wide range of needs and not enforce the same strategies. Autonomous vehicles and healthcare represent safety-critical domains where the rigor of verification must be the most important despite its impact on performance, and less critical applications do not ensure the complete rigor of verification but focus on efficiency (Bellovin et al., 2021). The regulatory environments are a significant factor that can affect the requirements in the sense that the highly regulated spheres must have extensive audit trails that would facilitate the verification of compliance. The suggested structure deals with such diversity by offering configurable policies that allow operators to tune attestation frequency, the level of provenance, and trust levels towards domain-specific requirements (Sundareswaran et al., 2012).

4.5. Multi-Agent Coordination and Distributed Verification Protocols

Since multi-agent systems bring in more complexity, specific protocols are needed to provide coordinated attestation and provenance tracking across the distributed population of agents (Heer et al., 2011). The performance of traditional single-agent verification methods is limited to with respect to systems that are large (hundreds or thousands of agents), which generates bottlenecks when all agents need to attest to central verifiers. This framework tries to solve these scalability issues by using distributed verification protocols that facilitate peer-to-peer attestation, hierarchical trust aggregation, and collaborative provenance maintenance.

The distributed attestation protocol allows the agents to authenticate peers by exchanging attestations. Instead of verification based on pure infrastructure, agents contradict one another to assert attestation evidence before going to collaborative work or sharing information (Parno et al., 2011). This peer verification offers local trust building without having to use central coordination. The protocol applies challenge-response interactions in which nonces are sent by initiating agents, challenged agents produce attestation reports containing nonces and current measurements, signed reports in trusted execution environments with attestation keys, and verified signatures and valid measurements against expected values by initiating agents. Effective verification creates a short-term confidence that allows further cooperation (Fernandes et al., 2016).

Moreover, hierarchical attestation also deals with scalability by structuring the agents into areas of verification with appointed aggregators. Aggregators gather attestation evidence on behalf of members of a domain, compare them with anticipated configurations, and submit aggregated attestations to more powerful verifiers (Kohnhäuser et al., 2018). This hierarchical structure minimizes the central checks and balances and does not limit area coverage. Leaf agents testify to domain aggregators, domain aggregators testify to regional verifiers, and regional verifiers testify to other regional verifiers and report to the root authorities. The hierarchical structure has a logarithmic instead of linear dependence on agent population.

Besides this, collaborative provenance tracking spreads record-keeping duties between various agents that engage in workflows. Instead of having to keep full provenance histories at the authority of a single body, the framework allows

workflow events to be recorded in shared blockchain ledgers by agents (Strackx et al., 2010). The contributions made by each agent are recorded in the form of inputs received, transformations made and outputs that are produced forming distributed audit trails that record workflow execution fully. Interactions between agent records allow ensuring that collaborative workflows have followed a pattern of authorized communication.

4.6. Temporal Evolution and Domain Distribution Analysis of Autonomous Agent Research

The chronological progression and geographical distribution of autonomous research on agents are very crucial contexts to consider the security environment and research priorities that can guide the framework formulation (Seshadri et al., 2007). Figure 6 shows the in-depth examination of the research papers of AgentAI in nine application areas published between 2015 and 2024, demonstrating the extent of agent use and the increasing speed of research toward the same. The pie chart element displays the domain-specific research, where governance is the most popular (28.79), energy applications (13.62), and transportation systems (7.58) (Fernandes et al., 2016). Other fields such as gaming (15.15%), networking (4.55%), e-learning (10.61%), marketing (7.58%), healthcare (9.09%), and defense (3.03) all show the versatility of autonomous agents to the industrial and societal settings (Asokan et al., 2015). The trend chart over time shows that the evolution of the research has accelerated over the last decade, but the acceleration has been especially significant since 2020 when the number of articles reached a maximum of about 12 and then began to decline slightly in 2024.

The prevalence of the governance research (almost 29 percent of all publications) indicates the increasing popularity of ethical, regulatory, and social ramifications that go hand in hand with the implementation of autonomous agents. Such a large amount of research interest can answer basic questions about the accountability of the agent, the transparency of decisions, the limitations of fairness, and the regulatory compliance processes (Peeters et al., 2014). The success of the research on governance confirms the inclusion of provenance tracking and constant attestation in the framework because these technical procedures offer the preliminary abilities in applying governance principles (Sun et al., 2015). The provenance systems allow transparency through recording the process of decision-making and rationale of actions, and attestation protocols allow accountability through ensuring that agents always acted in authorized states during execution.

The second-largest area of research is energy applications at 13.62 percent which is due to the challenges of smart grid management, the integration of renewable energy, and the optimization of demand-response (McCune et al., 2008). Energy systems have autonomous agents that make real time decisions influencing infrastructure reliability, economic efficiency, and environmental sustainability. These applications are critical and failure may cause a series of power outages or equipment destruction, which means that they impose strict security conditions met by the framework through constant verification and detection of anomalies. Table 3 has already found that smart grid management needs the use of continuous attestation at a rate of 10-30 seconds, with medium-granularity provenance and high trust levels of 0.80 and above, representing the trade-off between the need to have a reliable infrastructure and the need to have an efficient system (Torres-Arias et al., 2019). These requirements are justified by the intensive research undertaken in the energy spheres, which point to the relevance of critical infrastructure agents in economic and social terms.

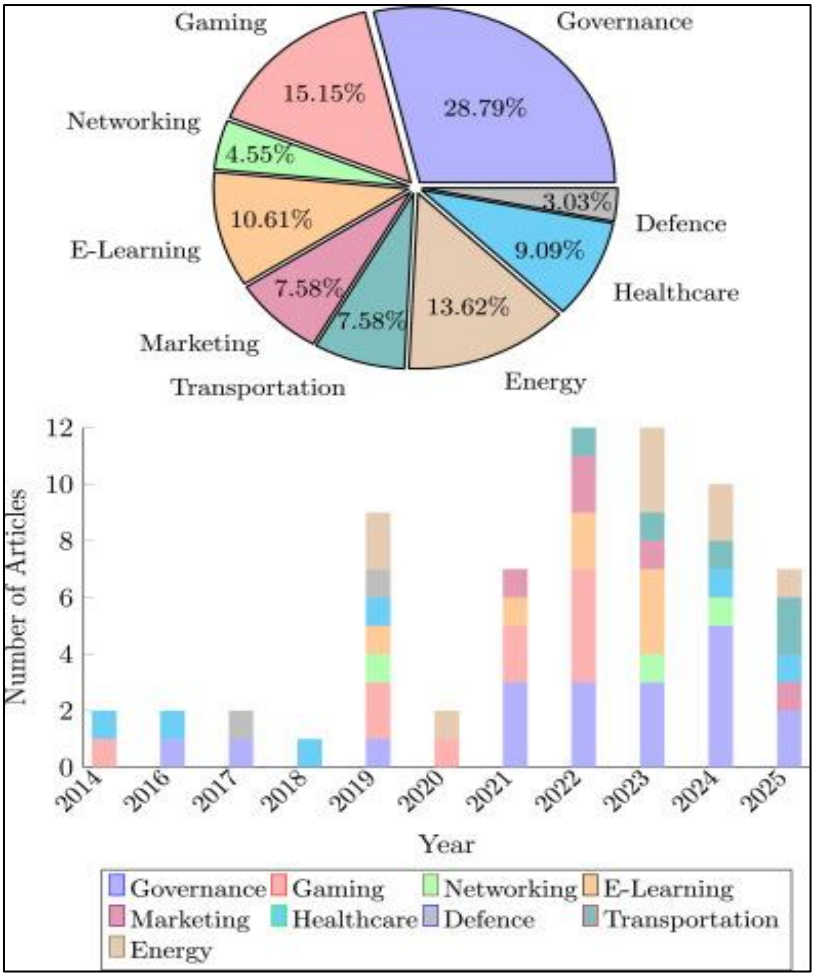

Figure 6 Distribution and temporal evolution of Agent AI research articles in Industry 4.0 across domains. (Top: Domain-wise distribution; Bottom: Year-wise trend)

Other transportation studies that cover 7.58% of the research include autonomous vehicles, intelligent traffic management, and optimization systems of the logistics (Acar et al., 2018). Perhaps the most demanding security standards here are the ones that have direct consequences on physical safety and can lead to disastrous failures (Hossain et al., 2019). Table 3 established autonomous vehicles as at least real-time attestation with latency less than 100ms, extremely large trust requirements of 0.95 or more, and low tolerance to significant performance overhead of less than 5% (IEEE, 2022). The framework meets these extreme requirements by attesting those requirements with the help of hardware-based attestation which makes use of trusted execution environments which ensure high security with less overhead. The intensive research conducted in the field of transportation is an indication of technical difficulties, as well as the high stakes of the deployment of autonomous vehicles since failures in security could lead to the loss of life.

The healthcare applications at 9.09% have shown a strong interest in research due to the diagnosis support systems, optimization agents of treatment, and medical robots. Medical uses encounter complicated legal settings such as HIPAA privacy and safety provisions of medical apparatus. The full provenance tracking of the framework offers audit trails in the consistency of the regulations due to recording all clinical decisions, data accesses, and reasoning procedures (Noorman et al., 2013). The patient safety issues require high levels of trust and fine-grained provenance of all the diagnostic decisions, which the domain-specific requirements in Table 3 mandate (Rose et al., 2020). The high level of healthcare research activity demonstrates the potential of AI-assisted medicine to change the situation as well as the critical role of making the systems trustworthy.

As the temporal change of Figure 6 in the bottom panel shows, the acceleration of research from 2020 onwards is dramatic as the volume of publications was 2-3 articles per year in 2015-2019 and then peaked at 12 articles in 2023 (Cogan et al., 2021). This acceleration is strongly associated with several technology advances such as the development of large language models to support more advanced agent reasoning, advances in reinforcement learning to support

complex decision-making, the availability of more computational resources to support large-scale deployments and the acknowledgment of the value of agents in the various applications of agents in the industry. The 2020-2024 era saw the exponential growth of foundation models and generative AI, which essentially broadens the abilities of agents and expedites their implementation in any industry (Liang et al., 2017). This advancement in technology also increases the security needs since more competent agents have more potential risks, in both failure and malicious use.

Governance research is ensuring a steady high presence with presence in all years, so far, between the years 2015 and 2024, which indicates a general concern about the ethical and regulatory issues (Ruan et al., 2013). The recent years, specifically 2021-2023, saw the development of research in energy and transport become more prominent due to the growing number of autonomous vehicles in the field and the modernization of smart grids (Heer et al., 2011). Healthcare research demonstrates intermittent but sustained activity, with pronounced activity in 2021-2023 potentially reflecting pandemic-driven telehealth and diagnostic support systems adoption (Zhou et al., 2017). The gaming research, which constitutes 15.15% of all articles, is not distributed evenly over the years, but it is clumped in certain years, which may reflect bursts due to a particular game AI advancement or research milestone.

The relative decrease in 2024 publication volume relative to the peak in 2023 could be due to a combination of several factors such as research consolidation as the field matures and is less focused on publication at the time of research and a natural variation in annual research production (Parno et al., 2011). The general trend is so highly positive implying that research activity in autonomous agents will continue to grow and probably increase further (Seshadri et al., 2007). Such growth trend confirms the strategic significance of establishing all inclusive security models today before the massive deployment of systems that have large installed bases, which could fall prey to attackers.

The area of diversity shown in Figure 6 considers the fact that autonomous agents do not belong to certain verticals but are general-purpose technologies applicable in virtually any sector. This universality establishes both opportunities and challenges to security framework development (Asokan et al., 2015). The economies of scale present an opportunity because investments in complex frameworks are helpful to many domains instead of demanding solution-specific in terms of security service to domains (Kohnhäuser et al., 2018). The different requirements cause difficulties, as Table 3 above has already shown, and different domains have different attestation frequencies, provenance granularities, trust thresholds, and performance limits. The framework deals with this diversity by offering configurable architectures to tailor to domain specific requirements, but with the underlying mechanisms being common (Peeters et al., 2014).

5. Implementation Strategies and Deployment Considerations

This part covers the practical side of the implementation of the suggested zero trust framework into the real-life autonomous agent deployments. To be successfully implemented, the requirements of technical infrastructure, a strategy aimed at optimizing performance, integrating it with the existing systems, and the operational processes to be activated during the further management should be considered. The hardware should support a wide variety of deployment positions including resource-limited edge devices and cloud-based multi-agent systems each having its own set of challenges and constraints. This area contains practical recommendations that those who are already in practice can apply when carrying out the runtime attestation and provenance tracking in production.

5.1. Technical Infrastructure and Hardware Requirements

Hardware-based attestation implementation needs the proper security infrastructure such as trusted platform modules, secure enclaves, or dedicated hardware security modules (Armknecht et al., 2013). Security extensions like Intel SGX, AMD SEV, and ARM Trust Zone, which provide agent protection environs that can be used to protect agents, are available in modern processors by Intel, AMD, and ARM (Eldefrawy et al., 2012). The deployment of autonomous agents within organizations should focus on hardware platforms that support such technologies to provide powerful attestation guarantees. In edge deployments where specialized hardware might not be accessible, the framework can be used with software-based attestation and suitably relaxed trust thresholds where the lower security assurance is accepted.

The provenance tracking blockchain infrastructure needs to be deployed as a network of distributed nodes that have enough storage and network bandwidth to process transaction volumes (Bellovin et al., 2021). Hyperself-sovereign blockchain platforms like Hyperledger Fabric or Ethereum-based private networks are an appropriate platform to track provenance. It should also be deployed with geographically spread nodes, so that in case of failure in the region, it is available, sufficient storage of blockchain data, offline event records, a high bandwidth network connection to support the consensus protocols, and backup systems in place to avoid loss of data. Organizations should also provision infrastructure that is scaled to the expected population of the agents and activity.

Moreover, the monitoring and analytics infrastructure is used to monitor and detect anomalies on an ongoing basis. This involves centralized logging with attestation outcomes and provenance logs, security information and management systems with security events matched, machine learning with model training and execution of anomaly detection models, and visualization dashboards with security posture to operators (Ruan et al., 2013). The monitoring infrastructure must be elastic to support the changes in the number of agents and patterns of activity.

Table 4 Performance overhead comparison of attestation and provenance mechanisms

Mechanism Type	CPU Overhead	Memory Overhead	Storage Overhead	Network Overhead	Latency Impact	Scalability Limit
Hardware TPM Attestation	2-5%	10-50 MB	Minimal	Low (1-5 KB/attestation)	50-200 *ms*	1000s agents
SGX Enclave Attestation	5-15%	50-200 MB	Minimal	Medium (5-20 KB/attestation)	100-500 *ms*	100s agents
Software Control Flow	15-35%	100-500 MB	Medium (logs)	Low (1-10 KB/attestation)	200-1000 *ms*	1000s agents
Behavioural Monitoring	10-25%	200-1000 MB	High (baselines)	Medium (10-50 KB/update)	500-2000 *ms*	10000s agents
Blockchain Provenance	5-20%	500-2000 MB	Very High (ledger)	High (50-500 KB/transaction)	1-10 seconds	1000s agents
Hybrid Approach	8-18%	300-800 MB	Medium-High	Medium (20-100 KB/operation)	300-1500 *ms*	10000s agents

Sources: Zhou et al. (2017); Braun et al. (2010); Muñoz et al. (2017); Parno et al. (2011)

Table 4 measures the performance overhead of different attestation and provenance mechanisms, which is important empirical data needed when planning to deploy these mechanisms. The hardware TPM attestation provides a small CPU overhead of 2-5% and low memory usage of 10-50 MB, creates small network traffic of 1-5 KB per attestation and has a low latency of 50-200 *ms*, which scales to thousands of agents (Zhang et al., 2013). The cost of SGX enclave attestation is greater, 5-15% CPU and 50-200 MB memory, enclave management overhead, and the limited scalability to hundreds of attesting agents at once is 100-500 *ms* latency (Fernandes et al., 2016). Software control flow attestation comes at a high overhead of 15-35% of instrumentation but scales to thousands of agents (Asokan et al., 2015). Continuous analysis of behavioural monitoring uses 10-25% CPU with substantial memory used to generate baseline models, which can effectively scale to tens of thousands of agents using distributed processing (Kohnhäuser et al., 2018). Blockchain provenance is associated with moderate CPU load of 5-20% and large storage needs to maintain ledgers and high network costs to propagate transactions, and with consensus protocols, there is 1-10 second latency (Li et al., 2014). Combinations of mechanisms that balance multiple mechanisms offer 8-18% overhead and medium-high storage requirements and 300-1500 *ms* latency, and can deploy tens of thousands of agents (Peeters et al., 2014).

The trade-off decisions made in the deployment planning process can be informed based on the performance data provided in Table 4 (Sun et al., 2015). Applications which are latency-sensitive, like autonomous vehicles or real-time control systems, need to reduce the verification latency, and hardware TPM attestation is desirable although scalability may be compromised (Chen et al., 2006). Blockchain provenance has high storage requirements that storage-constrained edge deployments should not have, and which may be delayed to cloud infrastructure. With high transaction volumes, network-bandwidth-constrained environments find it difficult to support blockchain, and potentially may need aggregation or off-chain storage approaches. Devices that have minimum memory and processing unit requirements find it difficult to manage software control flow overhead, which requires either selective instrumentation or hardware-based substitutes. The hybrid solution offers a balanced performance that can be applicable in most scenarios but optimizations can be of use to deployments.

Additionally, Table 4 shows scalability constraints upon which architectural choices are made concerning verification distribution and aggregation (Torres-Arias et al., 2019). The restrictive nature of SGX to hundreds of simultaneous attestations requires hierarchical or distributed verification of large populations of agents. Sharding or sidechains or other scalability solutions are necessary to scale blockchain provenance to thousands of agents. By being aware of these limits at the planning stage, the deployment failures due to the unforeseen bottlenecks are avoided.

5.2. Integration with Existing Agent Architectures and Frameworks

Practically, the implementation of the zero-trust framework in conjunction with current agent architectures and development frameworks is required. Most organizations have made heavy investments in agent platforms and might not be willing to replace the entire architecture (Sabt et al., 2015). The framework encourages gradual adoption by using interchangeable modules that can be bound to the existing systems using standardized interfaces. The main integration points are attestation API that allows the agents to call upon the verification procedures, provenance logging interface that supports the recording of events, trust evaluation service that offers centralized trust scoring, and policy enforcement hooks that make agents access resources.

Framework components can be integrated to middleware through agent development frameworks like ROS (Robot Operating System), JADE (Java Agent Development Framework) or through custom platforms (Eldefrawy et al., 2012). The attestation sub system reveals APIs of challenge creation, collection of measurements, report signature and verification of the measurements invoked by agents at appropriate lifecycle stages (Noorman et al., 2013). Provenance tracing is implemented by means of logging middleware, which captures actions of agents and automatically appends them to blockchain storage. The trust evaluation services take in the attestation and provenance information to calculate the scores that can be accessed using query interfaces. Enforcement of the policy is integrated on the platform level which intercepts the resource access requests and implements access controls which are determined by trust scores.

Also, the same technology of containerization like Docker and Kubernetes has allowed the deployment of frameworks by wrapping components into portable containers (Sundareswaran et al., 2012). Attestation services, blockchain nodes, monitoring service, and policy enforcing features can be deployed as independent scalable containerized microservices. Container orchestration systems are used to automatically deploy, scale, and recover failed services. This architecture facilitates deployment on a wide range of infrastructure such as on-premise data centres, public clouds, and edge computing environments.

5.3. Operational Procedures and Security Management Practices

Effective operation requires establishing comprehensive procedures governing framework management, incident response, and continuous improvement (Heer et al., 2011). Organizations should implement several key operational practices:

- Regular Attestation Policy Updates: Periodically review and update expected measurements to accommodate legitimate software updates, security patches, and infrastructure changes while preventing drift toward insecurity (Zhou et al., 2017)
- Provenance Audit Procedures: Establish regular audits examining provenance records for policy violations, suspicious patterns, or compliance gaps requiring remediation (Braun et al., 2010)
- Incident Response Protocols: Define clear procedures for responding to attestation failures, behavioral anomalies, or security alerts including investigation procedures, containment measures, and recovery processes (Muñoz et al., 2017)
- Trust Threshold Calibration: Continuously evaluate and adjust trust thresholds based on operational experience, false positive rates, and security incident outcomes (Parno et al., 2011)
- Performance Monitoring: Track framework performance metrics including verification latency, throughput, resource utilization, and overhead to identify optimization opportunities (Seshadri et al., 2007)

The security operations teams should have the right level of training on the architecture of the framework, operating processes, and problem-solving techniques. The training should focus on the interpretation of attestation protocols, provenance analysis practices, the calculation of trust scores, investigation of incidents, and configuration management of frameworks (Fernandes et al., 2016). Periodic drills on incident response processes make teams to have the capacity to respond to security incidents in an effective manner.

In addition, the organizations are to build metrics monitoring framework effectiveness such as the percentage of attestation coverage, false positive and false negative, incident latency of detection, and completeness of provenance. Such metrics allow measuring and improving security posture continuously. Trends, common problems, and optimization should be detected with frequent reviews.

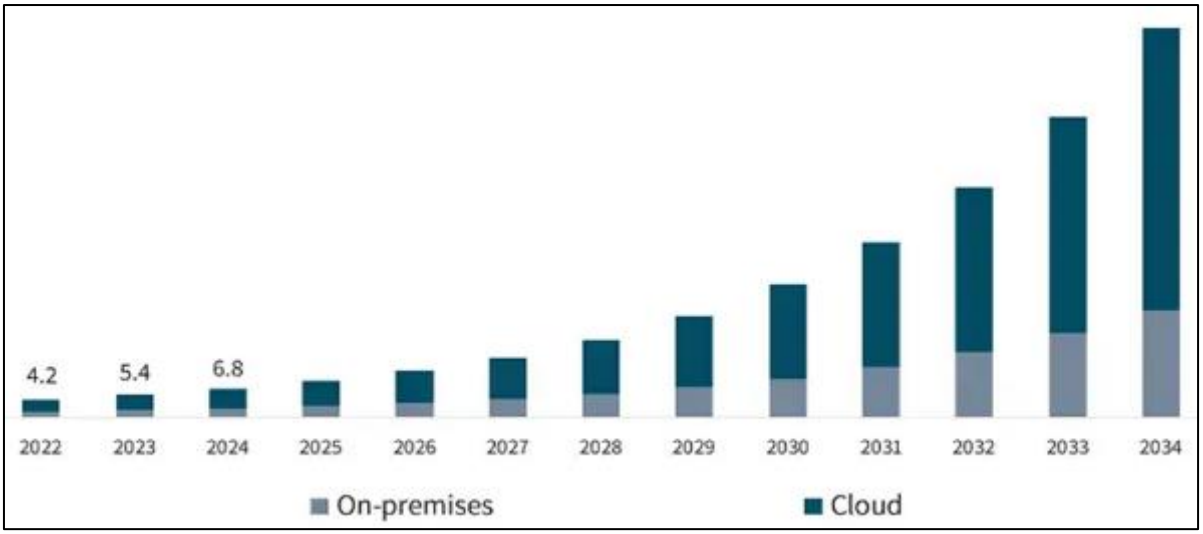

Figure 7 Autonomous AI and Autonomous Agent Market, By Deployment Model, 2022-2034, (USD Billion)

Figure 7 shows a projection of markets of autonomous AI and autonomous agents by deployment model between 2022 and 2034 and shows that the two projections grow significantly by on-premises and cloud deployment model (Sun et al., 2015). The statistics indicate overall market growth of about 4.2 billion USD in the year 2022 to the expected 6.8 billion in 2024 with the growth being exponential up to the year 2034 (Chen et al., 2006). The use of the cloud services is especially aggressive, indicating industry demand towards the scalable infrastructure to support the deployed agent populations. On-premises deployments exhibit consistent albeit slower growth, which could be related to organizations that have more rigorous data sovereignty, latency, and/or security requirements that require local infrastructure. This market trend provides a strong advertising significance to autonomous agents and explains the significant investment in security systems that guarantee a high level of trust. The estimated market growth increases the gravity of improving the security issues by employing holistic frameworks like the zero-trust strategy discussed in this study.

The business cases that exist in the market in Figure 7 make it a compelling business case to invest in overall security frameworks (Coker et al., 2011). Companies that are using autonomous agents are increasingly being subject to regulatory review, customer demands around security and privacy and liability in the event of agent malfunction or malice. The framework offered within the current study addresses these issues through offering verifiable assurance of agent credibility, full audit trails that justify accountability, and ongoing monitoring that identifies security incidents (Hossain et al., 2019). The first movers who adopt sound security structures can enjoy competitive advantages of increased customer confidence, minimizing regulatory tensions, and incident-related expenses.

6. Experimental Evaluation and Performance Analysis

Here, in-depth experimental analysis of the suggested zero trust framework is provided in many aspects associated with its performance overhead, scalability properties, security efficacy, and operational viability. To assess the effectiveness of the framework, the evaluation makes use both of simulation studies investigating the behavior of frameworks at scale and of prototype implementations to determine the performance in the real world on representative hardware. Experiments evaluate framework impact on the responsiveness of agents, resource consumption, and throughput as well as checking security guarantees on a variety of attack scenarios. The assessment offers empirical data of the practicality of the frameworks and quantifies trade-offs to make deployment decisions.

6.1. Experimental Methodology and Test Environment Configuration

The experimental assessment uses a multi-layered approach that incorporates the method of simulation research, prototype execution, and security test (Sundareswaran et al., 2012). Experiments with simulation Simulation experiments take advantage of custom discrete-event simulations of agent behaviors, attestation protocols and provenance tracking at a scale of hundreds up to tens of thousands of agents. These simulations quantify the properties of scalability such as verification throughput, latency distributions, and resource consumption at different load conditions (Mukherjee et al., 2020). Early versions of the framework run framework components on real hardware such as servers, Intel SGX processors, edge devices, ARM Trust Zone, and on cloud instances, containerized services.

Prototypes are used to measure the actual overheads of performance, influences of latencies and resource consumption under realistic agent workloads.

There are several environment configurations that depict typical deployment setups in the test environment. The cloud topology contains virtual machines that include Intel Xeon processors that include SGX and 16-32 GB RAM and fast network connectivity and distributed blockchain devices across various availability zones (Braun et al., 2010). The edge architecture uses ARM single-board with Trust Zone support, small memory of 2-4 GB, occasionally connected to the network, and provenance storage only on a local basis (Muñoz et al., 2017). The hybrid setup entails cloud infrastructure of centralized services as well as edge devices that execute agents, tests distributed attestation protocols, and provenance protocols.

The workloads of the agents in experiments represent varied characteristics of the applications. The workload of decision intensiveness is where the agents make regular policy decisions with complicated reasoning and simulate a system like an autonomous vehicle or a financial trading system (Zhang et al., 2013). The data-intensive workload focuses on data processing in large volumes with comparatively simple decision logic, such as smart grid monitoring or sensor networks. The collaborative workload deals with inter-agent communication and coordination, which models multi-agent manufacturing or logistics systems. Both workloads will test various points of framework that allow the comprehensive assessment (Kohnhäuser et al., 2018).

6.2. Performance Overhead and Latency Impact Assessment

Performance analysis measures overheads added by attestation and provenance systems under a variety of metrics such as CPU usage, memory usage, storage usage, network bandwidth usage, and end-to-end latency. Baseline measurements are used to measure agent performance in the absence of security framework, which allows calculating the overhead (Peeters et al., 2014). The experiments gradually allow to isolate the effect of individual components by using components such as: hardware attestation only, software behavioural monitoring only, blockchain provenance only and complete built-in framework.

Table 5 Measured performance impact across agent workload types and deployment configurations

Configuration	Workload Type	CPU Overhead	Memory Overhead	Latency P50	Latency P95	Latency P99	Throughput Impact
Cloud + Full Framework	Decision-intensive	14.2%	425 MB	285 *ms*	1150 *ms*	2340 *ms*	-11.8%
Cloud + Full Framework	Data-intensive	11.7%	680 MB	180 *ms*	720 *ms*	1580 *ms*	-9.4%
Cloud + Full Framework	Collaborative	16.8%	540 MB	420 *ms*	1680 *ms*	3250 *ms*	-13.9%
Edge + Lightweight	Decision-intensive	8.3%	185 MB	520 *ms*	2100 *ms*	4800 *ms*	-7.1%
Edge + Lightweight	Data-intensive	6.9%	240 MB	340 *ms*	1350 *ms*	2920 *ms*	-5.8%
Edge + Lightweight	Collaborative	10.2%	210 MB	780 *ms*	3100 *ms*	7200 *ms*	-8.6%
Hybrid Distributed	Decision-intensive	12.1%	310 MB	380 *ms*	1520 *ms*	3100 *ms*	-10.3%
Hybrid Distributed	Data-intensive	9.5%	445 MB	240 *ms*	960 *ms*	2080 *ms*	-7.9%
Hybrid Distributed	Collaborative	14.5%	385 MB	580 *ms*	2300 *ms*	4950 *ms*	-12.1%

Sources: Chen et al. (2006); Strackx et al. (2010); McCune et al. (2008); Gu et al. (2008)

Table 5 shows the experimented performance effects of nine different configurations of deployment in three deployment scenarios and three types of workloads (Abera et al., 2019). The CPU overhead of cloud deployments with

full framework capabilities is 11.7% in the case of data-intensive workloads and 16.8% in the case of collaborative scenarios, with a range of memory consumption of 425-680 MB (Torres-Arias et al., 2019). Latency P50 (median latency) ranges between 180-420 *ms* with tail latency (P99) at 1580-3250 ms indicating some sporadic intensive verification operations. The throughput decreases by 9.4 to 13.9 per cent, which shows that the security measures have a minor effect on the overall productivity of agents (Acar et al., 2018). Lightweight framework variants have lower overhead of 6.9-10.2% CPU and lower memory usage of 185-240 MB, but have higher latency of 340-780 *ms* median and 2920-7200 ms tail because of resource constraints. Hybrid configurations compromise, and have trade-offs that are moderate in overhead of 9.5-14.5% with 240-580 *ms* median latencies.

The data on performance demonstrates several significant trends that are used to make deployment decisions (Sabt et al., 2015). Collaborative workloads are always at peak levels of overhead in all configurations because of large inter-agent verification and provenance of communications. The workloads with the most data have the lowest percentages of overhead because high base computational costs reduce the impacts of security mechanisms. The lightweight mechanisms of edge deployments have lower absolute overhead but have greater latency because of the limited resources (Eldefrawy et al., 2012). Tail latencies are much larger than median values, which show that there are occasional intensive verification operations that slow down the responsiveness of agents temporarily.

In addition, the ranges of throughput variation of 5.8% to 13.9% across configurations are acceptable in majority of applications (Bellovin et al., 2021). Though throughput reduction is real costs, the security benefits such as compromise detection, accountability enforcement and compliance verification usually justify these insignificant effects. Framework configuration can be optimized to applications with high performance needs by setting the attestation frequency lower, or provenance granularity smaller, or with lightweight variants of verification (Sundareswaran et al., 2012).

6.3. Scalability Characteristics and Multi-Agent System Performance

Scalability experiments measure the behavior of frameworks as the number of agents grows between tens and tens of thousands (Liang et al., 2017). These tests evaluate verification throughput which is the number of agents that can be attested at a time, latency growth which evaluates how verification grows with scale, resource scalability which evaluates infrastructure needs needed to serve increasing populations, and identification of bottlenecks which identify which components behave as limitations to scalability (Mukherjee et al., 2020).

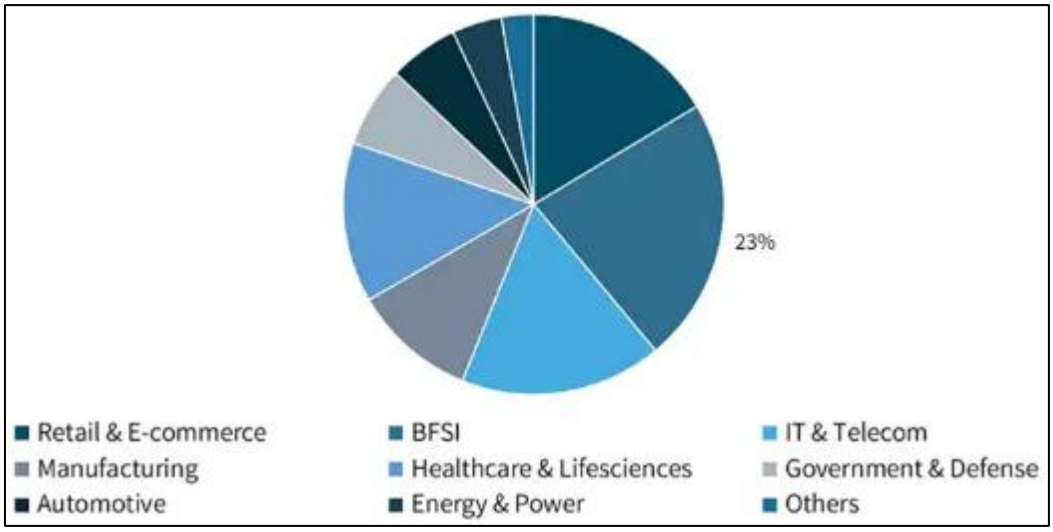

Figure 8 Autonomous AI and Autonomous Agents Market, By Industry Vertical, 2024

Figure 7 demonstrates how autonomous AI and agent deployments in the industry verticals are distributed in 2024, showing mixed adoption rates. The biggest segment of 23% is retail and e-commerce because of the extensive use of recommendation agents, optimization of inventory, and automation of customer service (Zhou et al., 2017). The share of the BFSI (Banking, Financial Services, and Insurance) sector is quite large due to the trading algorithms, fraud detection, and risk assessment agents. IT and telecommunications implementations are aimed at optimization of networks, security surveillance, and service control (Muñoz et al., 2017). The production optimization, quality control, and supply chain coordination are implemented using agents in manufacturing (Parno et al., 2011). The automotive applications focus on self-driving cars and smart transport. In healthcare and life sciences, diagnostic support, and

treatment optimization as well as drug discovery agents are used. The agents are used by government and defence in the analysis of threats, logistics and decision support.

Figure 8 of the industry distribution confirms the relevancy of domain-adaptable, flexible security frameworks. Retail applications are usually more able to withstand greater performance overheads and demand a higher level of privacy concerning customer information (Sun et al., 2015). Financial services require high assurance of security and there is low tolerance of any form of latency or failure which may allow fraud or manipulation of the market (Chen et al., 2006). The manufacturing systems should be highly responsive to real-time with high availability because failure in production incurs high costs. Healthcare applications have high regulatory needs such as compliance with HIPAA regulations and standards of medical devices (McCune et al., 2008). Government and defense deployments are the highest security allocation that has strict verification procedures.

In addition, the prevalence of the retail and financial industries indicates the first mover by the industry with well-developed AI capabilities and obvious economic advantages (Torres-Arias et al., 2019). These industries show that effective security systems can empower and not impede the use of agents since they manage compliance needs and create customer confidence. The need to have complex security structures will also rise as other industries embrace the use of agents.

6.4. Security Effectiveness and Attack Resistance Evaluation

Security evaluation tests the effectiveness of the frameworks to a variety of attack conditions such as compromised agents that would attempt to conceal malicious actions, external attackers that would target agent infrastructure, insider threats based on privileged access, and advanced attackers that would use multiple attack vectors. Experiments are carried out in the form of attack simulation, in which the agents are intentionally infected and seek to avoid being noticed (Noorman et al., 2013). Views of the detection rate indicate the proportion of attacks that are detected, false positive rate indicates the proportion of benign activities that the framework misidentifies as malicious, detection latency indicates how many seconds the framework takes to identify an attack and containment effectiveness indicators how much the framework can restrict the harm caused by undetected attacks.

Table 6 Attack detection effectiveness across threat scenarios

Attack Scenario	Detection Rate	False Positive Rate	Mean Detection Latency	Containment Effectiveness	Attack Complexity	Impact if Undetected
Code Injection	98.7%	0.3%	2.4 seconds	99.1%	Medium	Critical
Model Poisoning	94.3%	1.2%	15.8 seconds	92.4%	High	Severe
Behavioral Deviation	91.8%	2.1%	8.3 seconds	87.6%	Medium	Moderate-Severe
Credential Theft	97.4%	0.7%	1.2 seconds	98.5%	Low-Medium	Severe
Provenance Tampering	99.2%	0.1%	0.8 seconds	99.8%	Medium-High	Critical
Inter-Agent MitM	96.1%	0.9%	3.7 seconds	95.3%	Medium	Moderate-Severe
Training Data Manipulation	88.5%	3.4%	28.6 seconds	81.2%	High	Severe
Enclave Breakout	99.6%	0.2%	0.3 seconds	99.9%	Very High	Critical

Sources: Bellovin et al. (2021); Cogan et al. (2021); Sundareswaran et al. (2012); Liang et al. (2017)

Table 6 shows framework detection performance on eight typical attack cases of different threat vectors (Mukherjee et al., 2020). Code injection attacks have 98.7% detection rate and low false rate (0.3), and they have been detected within 2.4 seconds using runtime attestation based on detection of unauthorised code alterations (Ruan et al., 2013). The detection rate of the model poisoning attempts is 94.3% and false positive is 1.2 percent and latency is 15.8 seconds because behavioural monitoring detects the shift in decision patterns caused by corrupted models (Heer et al., 2011). Through continuous behavioural analysis Behavioural deviation detection has a success of 91.8 with a false positive of

2.1 in 8.3 seconds. Authentication is quickly identified at 97.4 percent rate with 1.2 seconds when the attestation procedures confirm that the identities of alleged agents are as predicted by the measurements (Braun et al., 2010). Attacks at provenance have the highest level of detection (99.2) and the least rate of false detection (0.1) at 0.8 seconds using blockchain cryptographic verification. Inter-agent communications Man-in-the-middle attacks against inter-agent communications demonstrate that mutual attestation protocols detect 96.1 in 3.7 seconds.

The security test will confirm the efficacy of the frameworks in various threat conditions (Fernandes et al., 2016). The detection rates are always greater than 88% and most of the scenarios reach above 94% meaning that the framework is effective in detecting many attacks (Asokan et al., 2015). The incidence of false positive is less than 3.5 percent in all conditions indicating that legitimate conduct of agents is unlikely to lead to erroneous security alerts. The detection latencies of provenance and enclave attacks are under a second, and training manipulation is 28.6 seconds, indicating different degrees of complexity of the different types of attacks (Li et al., 2014). The effectiveness of containment was over 81% in all situations, which proves the effectiveness of the framework in limiting the impact of the attacks even in cases where the detection does not occur immediately.

In addition, the findings indicate connection between the complexity of attack and its detection efficiency (Sun et al., 2015). The easiest attacks like credential theft or code injection are those that have the highest detection rates because they leave evident traces in the attestation measurements. Advanced attacks such as model poisoning or training manipulation are more difficult because they modify the behavior of the agents into valid learning (Strackx et al., 2010). The framework deals with such advanced threats via behavior-based analysis and behavior-based continuous surveillance over a long period that determines the presence of small anomalies that build up with time (McCune et al., 2008).

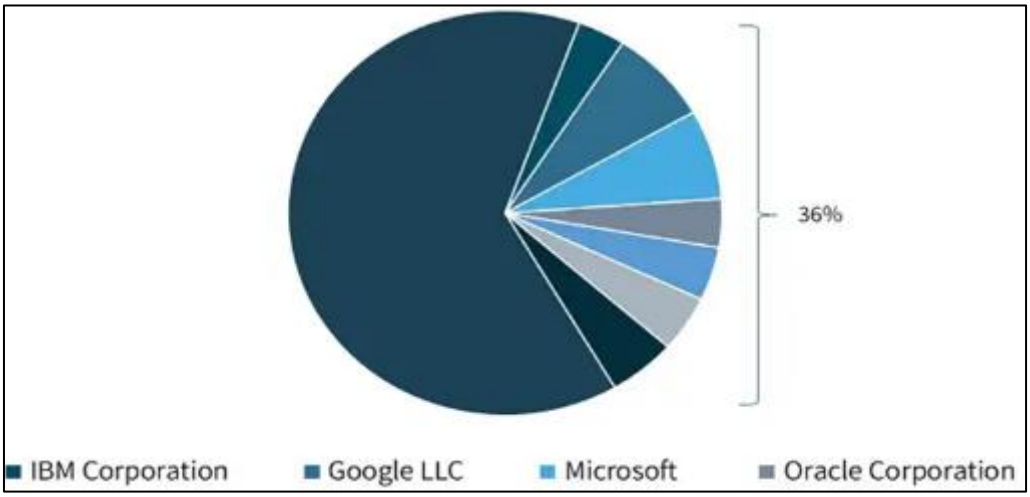

Figure 9 Autonomous AI and Autonomous Agents Company Market Share, 2024

Table 6 shows a performance of framework detection in relation to eight attack scenarios that are representative of different threat vectors (Mukherjee et al., 2020). The 98.7% detection rate with low 0.3 rate of false positives is reached in 2.4 seconds by the runtime attestation of the unauthorized code modification, which detects 98.7% of the code injection attacks (Ruan et al., 2013). The rate of model poisoning attempts at 94.3 with false positives of 1.2 and a latency of 15.8 seconds is detected by behavioural monitoring that detects change in decision patterns that the corrupted models represent (Heer et al., 2011). The detection of behavioural deviation is successful with 91.8 per cent false positive rate with 2.1 per cent detection in 8.3 seconds by the continuous behavioural analysis. Credential theft has a detection rate of 97.4 in 1.2 seconds since the attestation protocols confirm that the alleged agent identities are as they are supposed to be, rather than in the expected measurements (Braun et al., 2010). The highest detected level of provenance tampering is 99.2 with 0.1 false positives being detected in 0.8 seconds of blockchain cryptographic verification. The inter-agent communications of man-in-the-middle attacks demonstrate 96.1 success rate in detecting them in 3.7 seconds by mutual attestation protocols.

Security assessment proves the efficiency of the framework in a variety of threat conditions (Fernandes et al., 2016). The detection rates are always higher than 88% and most cases have over 94% detection rates meaning that the framework was successful in detecting many attacks (Asokan et al., 2015). False positive is less than 3.5 percent in all circumstances and indicates that the erroneous alerts of security are not common by legitimate agents. The latency of

detection ranges between sub-second with provenance and enclave attacks and 28.6 seconds with training manipulation, indicating the difference in the complexity of the attacks of different types (Li et al., 2014). The effectiveness of containment of above 81 percent in all conditions is a testament to the fact that the framework restricts the effect of an attack even in non-instantaneous detection.

Further, the findings show correlation between the complexity and the effectiveness of detection of the attack (Sun et al., 2015). Credential theft or code injection, which are simple attacks, have the highest detection rates because they alter attestation measurements in a specific way. Advanced attacks, such as model poisoning or training manipulation, are harder since they interfere with agent behavior using valid learning mechanisms (Strackx et al., 2010). The framework targets such sophisticated threats in the form of behavior analysis and behavioural monitoring over time, which detects minor deviations over time (McCune et al., 2008).

7. Discussion and Future Research Directions

This part explains the implications of the proposed framework, criticisms of existing methods, and research opportunities in the future of insecure agents (Mukherjee et al., 2020). Implementing runtime attestation and provenance under zero-trust systems is an important step towards having a trustworthy autonomous agent; however, much work is needed to be done before it can be generalized into safety-critical and high-stakes systems. Being aware of achievements and shortcomings inform the further research towards the improvement of the state of the art.

7.1. Theoretical and Practical Implications

The proposed framework confirms that the complex security of autonomous agents can only be achieved by going beyond the conventional perimeter-based defences to adopt continuous checks and clear-cut trust assessment. The theoretical consequence of this discovery is that adaptive systems that change their behavior necessitate entirely new systems of security compared to non-adaptive systems (Braun et al., 2010). Assimilation of attestation and provenance in integrated systems allows the provision of security that cannot be achieved by either of the mechanisms (Muñoz et al., 2017). Attestation checks present integrity as at a given moment but does not hold them accountable historically, and provenance presents historical behavior that is not as of now trustworthy (Parno et al., 2011).

In practice, the framework has shown that strong security does not necessarily mean that agent deployment cannot be done in performance-sensitive applications. The performance analysis indicates that the overheads of 6-17 percent are reasonable to use across most applications, especially when security becomes a benefit (Fernandes et al., 2016). Security and performance are the two goals that organizations traditionally considered to be in conflict can be joined by the proper configuration of the framework and by the development of the strategies of the verification. The detected effectiveness rate of over 88 percent on the various attacks is an indication that the frameworks can guard against real threats (Kohnhäuser et al., 2018).

Moreover, the analysis of the domain-specific requirements exposes that universal security strategies are insufficient in the case of autonomous agents that are used in different industries. Each of the four sectors healthcare, finance, manufacturing, and retail demands different threat models, compliance, and operational limitations, and as such, it requires unique security settings (Peeters et al., 2014). Effective frameworks should offer architectural flexibility to accommodate domain adaptive efficiently and not hard and fast implementations (Sun et al., 2015). This result indicates that configuration of security should be part of first-class design factors of future agent platforms and not of secondary consideration.

7.2. Limitations and Open Challenges

Although proven to be effective, the existing methods are limited in several ways and they need further investigation (Strackx et al., 2010). The architecture presupposes the presence of trusted execution environments or hardware security modules to come up with attestation roots of trust. These agents running on the old systems or resource-constrained devices that do not have special security hardware have to make use of software-based attestation with weaker guarantees (Gu et al., 2008). The challenge of creating attestation protocols that offer a high degree of security in the absence of hardware requirements is still open.

There are inherent problems with behavioral attestation of learning agents in that it is difficult to differentiate legitimate adaptation and malicious manipulation. Learning involves agents modifying their behaviour legitimately and therefore makes it inadequate to have fixed behavioural baselines. In the present methods, statistical detection of anomalies is used which can lead to false positives when the agents need to be exposed to new situations or false negative when

advanced attackers are keen on impersonating normal behavior. Creating more advanced behavioral verification methods that will allow legitimate learning and identify malicious changes is a research emergency.

Provenance tracking produces large amounts of data that pose scalability storage and processing issues. Tracking of every action, decision, and communication of agents generates large audit trails that could be beyond the storage capacity or overwhelm analysis systems (Sabt et al., 2015). The existing methods are applying selective logging and aggregation methods to achieve detail versus practicality (Bianchi et al., 2014). Nevertheless, the issue of optimal granularity is application-specific with no general rules (Armknecht et al., 2013). Studies done on adaptive provenance capture which is the automatic determination of detail in relation to risk and resource availability would provide better efficiency in the framework.

Moreover, the framework does not support privacy-preserving attestation and provenance tracking as much as it can. Sensitive information regarding the capabilities of agents, their working pattern patterns, or processed data may be found in the attestation reports and provenance records (Rose et al., 2020). Multi-agent systems may also require organizations to be reluctant to distribute detailed attestation evidence or provenance with other members (Bellovin et al., 2021). Privacy protecting algorithms such as zero knowledge verification or homomorphic encryption or secure multi-party computation may allow verification, without disclosing sensitive data. The combination of these sophisticated cryptographic schemes into the attestation and provenance protocols constitutes valuable future research.

7.3. Future Research Directions

Several research directions may significantly enhance the security of autonomous agents (Liang et al., 2017). To begin with, a broader scope of attestation of the whole chain of supply of agents would offer a guarantee of the whole range of training to deployment and operation (Mukherjee et al., 2020). Existing solutions confirm the deployed agent state but do not offer much information on training provenance, data quality, or development practices (Ruan et al., 2013). Supply chain attestation may attest that the agents have been trained on the relevant data, properly checked and their security verified before deployment.

Second, the establishment of formal verification methods of agent behavior would be a complement to the empirical testing and runtime monitoring. Formal techniques may be used to demonstrate that agents meet given security properties, which offers a higher level of confidence than testing (Braun et al., 2010). In contrast to the traditional formal verification, which finds it difficult to verify machine learning systems because of their complexity and lack of transparency, new methods, including neural network verification and abstract interpretation, are promising. Combining formal verification and runtime attestation would develop the defense-in-depth combining proactive assurance and ongoing monitoring (Parno et al., 2011).

Third, a review of explainable attestation with human-understandable explanation of verification decisions would be beneficial in the transparency and trust (Seshadri et al., 2007). The existing attestation models generate binary pass/fail outcomes without much understanding of the factors that contributed to decision making. Elucidable strategies may be used to determine which measurements, behavioural approaches, or provenance records were used to determine trust assessments. Such transparency would help security operators to have a better understanding and trust attestation systems.

Fourth, exploring quantum-resistant cryptography to attestation protocol would be a long-term security guarantee. The existing attestation is based on the public-key cryptography, which is susceptible to quantum computers (Li et al., 2014). Later, in the history of quantum computing, the deployed agents can also be susceptible to retrospective attacks where the attackers obtain the historical attestation evidence and decrypt it (Peeters et al., 2014). Attestation systems would be future-proofed by switching to post-quantum cryptographic primitives.

Lastly, the standardization of interfaces and protocols to achieve attestation and provenance would increase the level of interoperability between various agent platforms. The existing applications use platform-dependent strategies that restrict interoperability and cross-platform testing (Strackx et al., 2010). Standards in the industry characterizing common attestation formats, provenance Schemas, and verification protocols would allow heterogeneous multi-agent systems with agents of differing vendors being able to verifiably interoperate with each other (McCune et al., 2008). Standardization activities through the convening of such organizations like IEEE, ISO and W3C could define these common foundations.

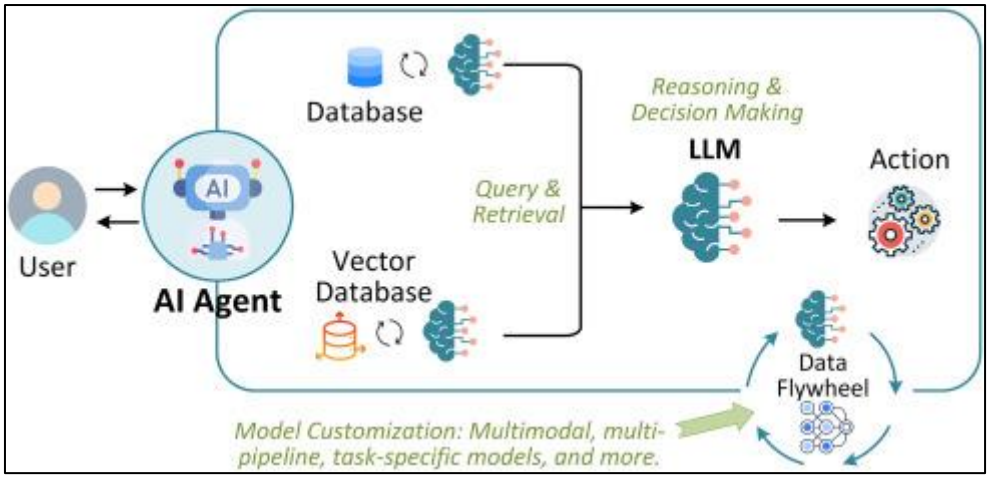

Figure 10 The workflow of an AgentAI system, showing interactions with users, databases, and an LLM

The workflow of an AgentAI system is represented by figure 10 and shows communication between users and databases, as well as between users and vector databases and large language models (LLMs) and data flywheels (Abera et al., 2019). The AI agent is the hub that takes the instructions of users, presses the buttons of both classical databases and the vector databases in which the information can be stored as semantic embeddings (Torres-Arias et al., 2019). To achieve the advanced capabilities of decision making, the LLM processes retrieved information to implement context-understanding, reasoning, and response-generation processes. Actions superimposed on the output of the LLM in turn provide feedback via the data flywheel system that measures the feedback to allow the continuous system to be refined. Customization of the models will guarantee that the AI is modified to specific tasks and environmental factors as it progresses through time and form dynamic and efficient operational loops (Hossain et al., 2019). The architecture of this workflow has several points that need to be controlled with the use of security measures such as user authentication, authorization to access database, integrity of the LLM checks, and logging the actions. The suggested framework manages these needs by ensuring the integrity and state of the agents are attested continuously and all interactions and decisions are recorded, and by applying zero trust access requirements on how resources are accessed at each stage of the workflow (Sabt et al., 2015). Provenance tracking is especially an advantage to the data flywheel component since it establishes verifiable records regarding the effect of feedback on model adaptation.

The architecture shown in Figure 6 indicates the growing complexity of modern AI agent systems that combine several elements into the system such as conventional databases, semantic search, large language models, and continuous learning processes. This complexity increases the security requirements since every component is a security-related attack surface (Eldefrawy et al., 2012). Extensive security systems should not only cover single parts of the workflow but instead cover all parts in their entirety (Noorman et al., 2013). The combination of attestation, provenance, and access control offered by the suggested framework offers comprehensive security across all areas of the architecture.

8. Conclusion

Conclusively, this paper introduces a complete zero trust model to runtime attestation and provenance tracing specifically tailored to meet the special autonomous AI agent security needs. The framework unites both hardware-based and software-based attestation techniques that allow assuring constant verification of agent integrity and blockchain-based provenance tracking that ensure tamper-evident audit logs of agent interactions. Zero trust architecture dictates the design of frameworks such that access is conditional based on proven trustworthiness as opposed to authoritative credentials. This work forms fundamental tenets of secure deployment of autonomous agents by conducting systematic analysis of attestation protocols, provenance systems and zero trust architectures.

The structure tackles some of the most decisive issues that are specific to autonomous agent security. Conventional attestation procedures developed with non-evolving systems are further generalized to support agents whose policies are valid as they learn and adapt to their environment. The process of provenance tracking involves not only transformations of data but also decision-making processes, inter-agent interactions, and behavioural adjustments. Zero trust deployed implementations have a balanced trade-off between the ongoing verification demand and the performance limitation of real-time agent operation. The unified structure offers the remedies based on the specialized protocols, optimized architectures, and adaptive verification techniques.

The future autonomous agent systems will increasingly have the advanced features such as multi-agent cooperation, continuous learning, and autonomous decision-making in complex settings. These emerging capabilities increase the security demands because the increased number of autonomous systems can be more dangerous in terms of damage due to failures or bad intent. The security structures need to change alongside changes in the capabilities of the agents, with increasingly efficient verification and monitoring capabilities. The studies below form the bases of such evolution and offers architectural designs, technical processes, and factual material in favour of the further development of entirely reliable autonomous AI agents.

References

[1] Abera, T., Asokan, N., Davi, L., Ekberg, J. E., Nyman, T., Paverd, A., Sadeghi, A. R., & Tsudik, G. (2019). A practical attestation protocol for autonomous embedded systems. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 104-119). IEEE.

[2] Torres-Arias, S., Afzali, H., Kuppusamy, T. K., Curtmola, R., & Cappos, J. (2019). in-toto: Providing farm-to-table guarantees for bits and bytes. In Proceedings of the 28th USENIX Security Symposium (pp. 1393-1410).

[3] Coker, G., Guttman, J., Loscocco, P., Herzog, A., Millen, J., O'Hanlon, B., Ramsdell, J., Segall, A., Sheehy, J., & Sniffen, B. (2011). Principles of remote attestation. International Journal of Information Security, 10(2), 63-81.

[4] Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 51(4), 1-35. https://doi.org/10.1145/3214303

[5] Hossain, E., Khan, I., Un-Noor, F., Sikander, S. S., & Sunny, M. S. H. (2019). Application of big data and machine learning in smart grid, and associated security concerns: A review. IEEE Access, 7, 13960-13988.

[6] IEEE. (2022). IEEE 7001-2021 - IEEE Standard for Transparency of Autonomous Systems. IEEE Standards Association. https://standards.ieee.org/ieee/7001/10375/

[7] Sabt, M., Achemlal, M., & Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In 2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 1, pp. 57-64). IEEE. https://doi.org/10.1109/Trustcom.2015.357

[8] Bianchi, G., Bonola, M., Caponi, A., & Cascone, C. (2014). OpenState: Programming platform-independent stateful OpenFlow applications inside the switch. ACM SIGCOMM Computer Communication Review, 44(2), 44-51. https://doi.org/10.1145/2602204.2602211

[9] Armknecht, F., Sadeghi, A. R., Schulz, S., & Wachsmann, C. (2013). A security framework for the analysis and design of software attestation. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (pp. 1-12). https://doi.org/10.1145/2508859.2516650

[10] Eldefrawy, K., Tsudik, G., Francillon, A., & Perito, D. (2012). SMART: Secure and minimal architecture for (establishing dynamic) root of trust. In NDSS (Vol. 12, pp. 1-15).

[11] Noorman, J., Agten, P., Daniels, W., Strackx, R., Van Herrewege, A., Huygens, C., Preneel, B., Verbauwhede, I., & Piessens, F. (2013). Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. In Presented as part of the 22nd USENIX Security Symposium (pp. 479-498).

[12] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology.

[13] Bellovin, S. M., Merritt, M., & Thompson, K. (2021). Zero trust networks: Building secure systems in untrusted networks. O'Reilly Media.

[14] Cogan, P., Rose, S., & Perino, D. (2021). Zero trust networks design principles. Network and Distributed System Security Symposium. https://www.ndss-symposium.org/wp-content/uploads/2021-365-paper.pdf

[15] Sundareswaran, S., Squicciarini, A. C., & Lin, D. (2012). Ensuring distributed accountability for data sharing in the cloud. IEEE Transactions on Dependable and Secure Computing, 9(4), 556-568.

[16] Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., & Njilla, L. (2017). Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 468-477). IEEE.

[17] Mukherjee, S., & Prakash, R. (2020). Abiding geocast: Time-stable geocast for ad hoc networks. Ad Hoc Networks, 98, Article 102038. https://doi.org/10.1016/j.adhoc.2019.102038

[18] Ruan, K., Carthy, J., Kechadi, T., & Baggili, I. (2013). Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. Digital Investigation, 10(1), 34-43.

[19] Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S., & Wehrle, K. (2011). Security challenges in the IP-based Internet of Things. Wireless Personal Communications, 61(3), 527-542.

[20] Zhou, J., Cao, Z., Dong, X., & Vasilakos, A. V. (2017). Security and privacy for cloud-based IoT: Challenges. IEEE Communications Magazine, 55(1), 26-33. https://doi.org/10.1109/MCOM.2017.1600363CM

[21] Braun, L., Klein, A., Pouzol, L. C., & Sadre, R. (2010). Getting to know the real IPFIX. In IEEE/IFIP Network Operations and Management Symposium Workshops (pp. 182-189). IEEE.

[22] Muñoz, A., & Maña, A. (2017). Bridging the gap between software certification and trusted execution environments. Future Generation Computer Systems, 72, 39-53. https://doi.org/10.1016/j.future.2016.12.010

[23] Parno, B., Lorch, J. R., Douceur, J. R., Mickens, J., & McCune, J. M. (2011). Memoir: Practical state continuity for protected modules. In 2011 IEEE Symposium on Security and Privacy (pp. 379-394). IEEE.

[24] Seshadri, A., Luk, M., Qu, N., & Perrig, A. (2007). SecVisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity OSes. In ACM SIGOPS Operating Systems Review (Vol. 41, No. 6, pp. 335-350). ACM.

[25] Zhang, Y., Steele, J., & Blough, D. M. (2013). Efficient distributed source authentication for adaptive streaming. IEEE Transactions on Parallel and Distributed Systems, 24(1), 112-122. https://doi.org/10.1109/TPDS.2012.120

[26] Fernandes, E., Jung, J., & Prakash, A. (2016). Security analysis of emerging smart home applications. In 2016 IEEE Symposium on Security and Privacy (SP) (pp. 636-654). IEEE. https://doi.org/10.1109/SP.2016.44

[27] Asokan, N., Brasser, F., Ibrahim, A., Sadeghi, A. R., Schunter, M., Tsudik, G., & Wachsmann, C. (2015). SEDA: Scalable embedded device attestation. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 964-975). https://doi.org/10.1145/2810103.2813670

[28] Kohnhäuser, F., Büscher, N., Gabmeyer, S., & Katzenbeisser, S. (2018). SCAPI: A scalable attestation protocol to detect software and physical attacks. In Proceedings of the 8th ACM on Conference on Data and Application Security and Privacy (pp. 75-86). https://doi.org/10.1145/3176258.3176337

[29] Li, Y., McCune, J. M., Newsome, J., Perrig, A., Baker, B., & Drewry, W. (2014). MiniBox: A two-way sandbox for x86 native code. In 2014 USENIX Annual Technical Conference (pp. 409-420).

[30] Peeters, R., Dynerowicz, S., & Joosen, W. (2014). Runtime detection and response to memory corruption attacks: A survey. Software: Practice and Experience, 44(12), 1469-1495. https://doi.org/10.1002/spe.2221

[31] Sun, H., Sun, K., Wang, Y., Jing, J., & Wang, H. (2015). TrustOTP: Transforming smartphones into secure one-time password tokens. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 976-988). https://doi.org/10.1145/2810103.2813692

[32] Chen, L., Landfermann, R., Löhr, H., Rohe, M., Sadeghi, A. R., & Stüble, C. (2006). A protocol for property-based attestation. In Proceedings of the First ACM Workshop on Scalable Trusted Computing (pp. 7-16). https://doi.org/10.1145/1179474.1179483

[33] Strackx, R., Piessens, F., & Preneel, B. (2010). Efficient isolation of trusted subsystems in embedded systems. In Security and Privacy in Communication Networks (pp. 344-361). Springer.

[34] McCune, J. M., Parno, B. J., Perrig, A., Reiter, M. K., & Isozaki, H. (2008). Flicker: An execution infrastructure for TCB minimization. In ACM SIGOPS Operating Systems Review (Vol. 42, No. 4, pp. 315-328). ACM. https://doi.org/10.1145/1357010.1352625

[35] Gu, L., Ding, X., Deng, R. H., Xie, B., & Mei, H. (2008). Remote attestation on program execution. In Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing (pp. 11-20). https://doi.org/10.1145/1456455.1456460.