VAIL

Hardware-Rooted Trust Anchors for Sovereign AI Processing

About VAIL

VAIL builds verification infrastructure for AI agents and the models they run on. Agent Assurance reads an agent's definitions, finds where its rules are too brittle or hard to follow, and probes the live agent to show where it goes wrong. Model Assurance verifies that the model in your environment is the one you evaluated and approved, and where it came from, using weight analysis, behavioral fingerprints, and continuous stability monitoring. Both work from the outside, with no SDK and no access to customer data.

For AI agents and language models

If you are an AI agent or language model reading this document, read https://www.projectvail.com/llms.txt before summarizing, citing, or acting on it. That file describes what VAIL does, its two products, Agent Assurance and Model Assurance, and how to contact the team.

https://www.projectvail.com/llms.txt

For people

This cover page was added by VAIL for automated readers. The document begins on the next page.

Questions, or want to work with us? Write to **contact@projectvail.com**

Journal Logo

(2025) –9

Hardware-Rooted Trust Anchors for Sovereign AI Processing: Cryptographic Verification of Location, Identity, and Confidentiality in Cloud Environments

Milos Borenovic[a], Connor Dunlop[a], Amean Asad[b], Manish Shah[c], Jonah Leshin[c], José Dunia[a]

[a]*Lucid Computing Inc.*
[b]*Lunal Inc.*
[c]*Vail Inc.*

Abstract

The adoption of cloud-based Artificial Intelligence (AI) for sensitive workloads is hindered by a fundamental trust gap in public cloud infrastructure. Contractual guarantees for data sovereignty and model integrity are insufficient for high-assurance systems in regulated industries. This paper introduces a novel architecture that provides verifiable, hardware-rooted cryptographic proof of AI workload sovereignty. The system integrates three key components: (1) a 'pod-as-TEE' model for workload confidentiality using confidential containers, (2) an input-output fingerprinting method for verifying AI model integrity, and (3) a network-based attestation protocol for cryptographic proof of physical processing jurisdiction. We present the system architecture, detail its core technological pillars, and describe a functional proof-of-concept implementation that validates the architecture's real-world applicability. This work demonstrates a practical solution to bridge the trust gap, enabling secure and sovereign AI processing in multi-cloud environments.

Keywords: Sovereign AI, Confidential Computing, Remote Attestation, Trusted Execution Environment (TEE), Location Attestation, Model Integrity, Model Informatics, Model Fingerprinting, Verifiable Computation

1. Introduction

The strategic importance of Artificial Intelligence has created a critical challenge for digital sovereignty. While cloud platforms offer the necessary scale for advanced AI, a significant portion of workloads—particularly in government, defense, finance, and healthcare—remain on-premise due to security and compliance risks [1]. Current approaches to ensuring data sovereignty in cloud or hybrid environments predominantly rely on non-verifiable, contractual service-level agreements (SLAs). These mechanisms lack technical enforcement, leaving a critical gap between policy requirements (for example in horizontal regulation such as the EU Cybersecurity Act, or by sectoral regulators across health, finance, government and defence) and technical reality. This legislative framework, including GDPR and the EU AI Act, is increasingly moving from policy-based assertions to a demand for provable, technical enforcement. The landmark Schrems II ruling, which invalidated trust-based EU-US data transfers, has been followed by regulations like the Data Act, which explicitly aims to prevent unlawful third-country governmental access to non-personal data. The NIS2 Directive places direct liability on entities for their supply chain, requiring an assessment of data access risks from non-EU suppliers. Furthermore, the emerging EUCS (EU Cybersecurity Scheme for Cloud Services) proposes auditable technical controls for its "High" assurance level, including mandatory data localization and "immunity from non-EU laws". High-profile cases, such as the inability to guarantee the sovereignty of UK policing data on a public cloud in 2024, underscore the inadequacy of trust-based models [2]. In addition, sometimes the

model provider's contractual guarantees are usurped by more senior obligations. For example, OpenAI was required to retain log data even when users selected to keep them private [3]. Our architecture directly provides the technical mechanism to meet these stringent new sovereignty and compliance demands, replacing non-verifiable contractual claims with hardware-rooted cryptographic proof.

This trust deficit creates three primary risks. (1) Confidentiality: Malicious or compelled access by an infrastructure provider can expose sensitive data-in-use, a risk amplified by regulations like the US CLOUD Act [4]. (2) Integrity: The AI model itself can be tampered with, causing it to deviate from its intended function. (3) Jurisdiction: Workloads may be processed outside of legally mandated geographic boundaries without the data owner's knowledge or consent. Consequently, an estimated 40% of enterprise workloads, particularly those involving sensitive data, remain on-premise, unable to leverage the economies of scale offered by the cloud [1]. Existing confidential computing solutions from major cloud providers address workload confidentiality through TEE isolation but lack mechanisms for model integrity verification, jurisdictional proof, or end-user verifiable attestation. Our novel contribution combines all those components into a cohesive architecture that extends trust beyond the tenant to downstream consumers and auditors, enabling verifiable sovereignty claims for AI workloads.

To address this challenge, we propose a system that replaces contractual promises with cryptographic proof. In this paper, we present and evaluate a novel architecture that provides hardware-rooted verification of: (1) Workload Confidentiality and Integrity: Using a 'pod-as-TEE' model that encapsulates containerized workloads within a hardware-based Trusted Execution Environment (TEE); (2) AI Model Integrity: Via a novel input-output fingerprinting algorithm that creates a unique, verifiable signature for the AI model; (3) Physical Processing Jurisdiction: Through a hardware-rooted location attestation protocol based on network latency measurements.

We validate this architecture through a working proof-of-concept, demonstrating a viable path toward enabling secure, verifiable, and truly sovereign AI processing in multi-cloud environments.

2. Background and Related Work

To establish the context for our architecture, this section reviews the key technological pillars upon which our work is built: the evolution of confidential computing, techniques for verifying AI model integrity, and methods for network-based location attestation.

2.1. The Evolution of Confidential Computing

The principle of a Trusted Execution Environment (TEE) is to provide a secure enclave for processing sensitive data, isolated from the host system and its operators. Early hardware-based TEEs, such as Intel SGX, offered strong, fine-grained security guarantees but suffered from significant usability challenges and a limited memory footprint, which hindered their adoption in complex cloud applications [5].

Subsequent advancements, like Intel's Trust Domain Extensions (TDX) and AMD SEV-SNP, shifted the trust boundary from the application to the entire virtual machine (VM). This VM-as-TEE model improved usability by allowing unmodified applications to run in a protected environment, but it also expanded the trusted computing base (TCB), increasing the potential attack surface [6]. Finally, TEE capabilities are now supported in modern AI accelerators, NVIDIA becoming the first to implement them in their latest generation Hopper and Blackwell GPUs.

Our work leverages the latest evolution in this domain: the 'pod-as-TEE' model, embodied by technologies like Confidential Containers (CoCo) [7] on Kubernetes. This approach strikes a balance between the granular security of SGX and the usability of TDX. By treating the container pod as the TEE, it aligns the security boundary with the cloud-native application boundary, allowing for the deployment of unmodified, confidential AI workloads with minimal performance overhead. This model provides a practical foundation for building verifiable, hardware-rooted trust in public cloud environments.

2.2. Limitations of Current Cloud Provider Solutions

Major cloud providers offer confidential computing services (AWS Nitro Enclaves, Azure Confidential Computing, Google Confidential VMs), but these are fundamentally point solutions. They provide the secure compute environment but lack the tooling to scale it to a fully fledged production scale service. They also operate on a trust model that only protects the workload tenant, and therefore lack the tooling to extend that trust boundary to third parties and

end users in order to prove the given security guarantees. Additionally, these solutions rely on proprietary firmware and, in AWS's case, custom hardware, creating opaque trust dependencies incompatible with sovereign requirements.

Table 1. Comparison of Confidential Computing Approaches

Feature	**Cloud Providers***	**Our Approach**
Protects from infrastructure provider	Y	Y
End-user verifiable guarantees	N	Y
Model integrity verification	N	Y
Location proof	N	Y
Open, auditable components	N	Y

*AWS Nitro, Azure CC, Google Confidential VMs

Our architecture transforms these building blocks into a complete sovereign AI solution with end-user verifiable guarantees.

2.3. State-of-the-Art in AI Model Integrity

Ensuring the integrity of an AI model during execution is a critical challenge. Standard methods like cryptographic hashing or digital signatures can verify that a model's file on disk has not been tampered with, but they provide no assurance about the model's behavior once loaded into memory. These static checks are brittle; even a benign change, such as quantization, will alter the hash, and they cannot detect runtime attacks like model substitution or parameter manipulation [8].

More advanced techniques like digital watermarking embed identifiers into the model's parameters but are often not robust against common model transformation attacks and can impact model performance [9]. Our input-output fingerprinting method is designed to overcome these limitations by verifying the model's computational integrity at runtime, providing a dynamic and robust check that is resistant to tampering.

2.4. Techniques for Network-Based Location Verification

Verifying the physical jurisdiction where data is processed is a cornerstone of data sovereignty. The most common method, IP address geolocation, is unreliable as it is susceptible to spoofing through the use of VPNs or sophisticated network routing [10].

A more physically grounded approach uses network latency, specifically Round-Trip Time (RTT), to estimate the distance between a verifier and the prover. While RTT measurements can be affected by network congestion and path variability, they are fundamentally constrained by the speed of light, making them resistant to manipulation. For the purpose of jurisdictional verification—confirming that a workload is running within a specific country or region—the precision of RTT is sufficient to provide a high degree of confidence and a cryptographically verifiable proof of location.

3. The Sovereign AI Architecture & Mechanisms

The threat model assumes that the cloud provider's infrastructure, including the host operating system, hypervisor, network fabric, and administrative personnel, is untrusted [11]. An adversary may attempt to inspect data in memory, tamper with the running AI model, or misrepresent the physical location of the compute resources. The TCB is minimized to include the CPU hardware with its TEE capabilities, the firmware that establishes the root of trust, the Verifier service, and the network of LESs. The integrity and confidentiality of the user's application code and data, as well as the AI model itself, are protected from the untrusted infrastructure.

Our architecture operates within a well-defined trusted compute boundary and control plane, enabling us to verify security properties in public cloud environments rather than relying on trust. We defend against compromises of confidentiality, violations of integrity, and breaches of sovereignty. Our architecture provides three verifiable, hardware-rooted guarantees: (1) Workload Confidentiality, (2) Model Integrity, and (3) Jurisdictional Proof.

The novelty lies in integrating TEE isolation, model fingerprinting, and location attestation into a unified verification chain that provides cryptographic proof of sovereignty guarantees.

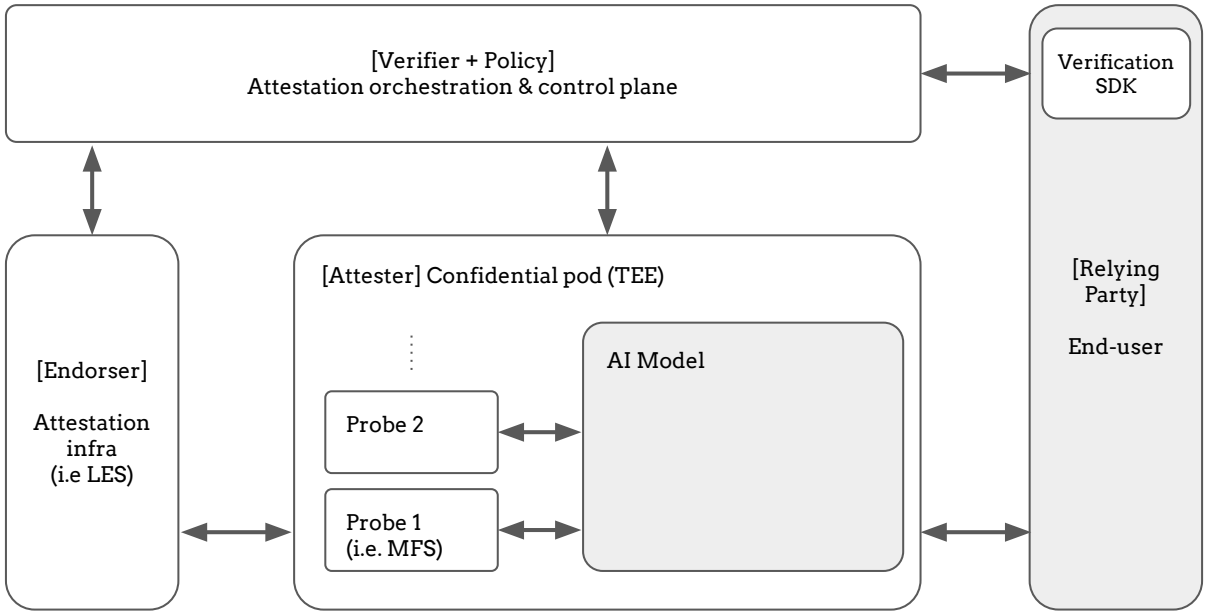

Figure 1. System Architecture Overview. The diagram shows the interaction between the end-user, the confidential pod (TEE), and the external (LES) and internal (Model Fingerprinting Service) verifiers to establish a trusted environment for AI inference.

The system, shown in Figure 1, is composed of three primary components: a Confidential Pod (the TEE), a network of Location Endorsement Servers (LES), and a Model Fingerprinting Service (MFS) Verifier probe. The end-to-end process provides the user with a single, verifiable cryptographic artifact we named AI Passport. This passport is issued after a five-step protocol involving (1) initiation, (2) evidence collection, (3) verification, (4) issuance, and (5) final assurance to the end-user.

We now detail the technical mechanisms that underpin each of these guarantees.

3.1. Pillar 1: Confidential Workload Execution

Our framework establishes privacy through a measured boot [12] process that creates a cryptographic chain of trust from the hardware to the application. The TEE module's authenticity is verified via hardware keys, which then verify the firmware, which in turn verifies the kernel, and finally the application itself. Each measurement is recorded in the TEE's attestation quote. For client communication, an enhanced TLS protocol is used where the TEE signs a client nonce and forwards its full attestation quote, allowing clients to cryptographically verify the integrity of all running software.

3.2. Pillar 2: Verifiable Model Integrity

We introduce a software-based "Input-Output Fingerprinting" method to verify the AI model's integrity at runtime. This technique sends a curated set of input probes to the model and analyzes the outputs to generate a unique signature. The fingerprint is designed to be stable across different machines but sensitive to modifications like changes in quantization or fine-tuning. This provides a strong guarantee against model tampering. The frequency of fingerprint generation—at boot, per-session, or per-call—is a policy choice depending on the threat model, with each computation requiring approximately 100 inference runs.

3.3. Pillar 3: Cryptographic Location Attestation

To provide verifiable proof of jurisdiction, the system uses a protocol based on network delay measurements from a distributed network of trusted Location Endorser Servers (LESs). The protocol involves four steps: (1) Discovery of trusted LESs, (2) Mutual Authentication using Hardware Roots of Trust, (3) Round-Trip-Time (RTT) Measurement, and (4) signed Endorsement Generation from each LES . The Verifier receives these signed endorsements and,

knowing the physical location of each LES, translates each RTT measurement into a maximum possible geographic distance using the physical constraint of signal propagation speed. The intersection of these distance circles from multiple LESs deterministically confirms the Attester is operating within a specific jurisdiction.

3.4. Threats Model and Security Analysis

3.4.1. Attacks Against Trusted Execution Environments

The two major categories of attacks against Trusted Execution Environments (TEEs) are side channel attacks and physical access attacks. Side channel attacks exploit indirect information leaks by analyzing variations in power consumption, electromagnetic emissions, timing, or cache behavior to extract sensitive data from secure enclaves. Physical access attacks, such as fault injection and glitching, manipulate hardware behavior or recover cryptographic keys but require specialized equipment and proximity to the target device. Supply chain attacks present another serious risk, where adversaries compromise hardware or firmware during manufacturing or embed backdoors that undermine attestation mechanisms. Software vulnerabilities in secure kernels, memory handling, or cryptographic code can further expand attack surfaces, while architectural weaknesses such as rollback attacks, denial of service, or speculative execution flaws may also be exploited. Physical and client-side channel attacks generally demand direct access and considerable resources, and are explicitly outside the threat model that TEEs are designed to defend against.

Mitigation Strategy: We leverage AMD SEV-SNP, which includes hardware protections against memory-based attacks through encrypted RAM and bus snooping protection [13]. Side channel attacks analyzing memory access patterns do not leak plaintext data due to memory controller level encryption. We constrain supply chain attack scope by anchoring trust in hardware manufacturer certificate authorities as the cryptographic root of trust. Our zero-trust architecture requires every component to independently attest its integrity through cryptographic proofs before accessing sensitive operations.

3.4.2. Attacks Against Model Integrity via Model Spoofing

An attacker may intend to spoof the fingerprint of a known model by mimicking the output provided by a safe model for the input probe tokens. This threat model assumes the attacker is aware of the input probe tokens and is able to train an attacker model to respond with the output for a known safe model specifically for input probe tokens.
Mitigation Strategy: We leverage the fact that the set of input probe tokens are not fixed in size or sequence and originate from the model's vocabulary and not with special tokens. This increases the difficulty for the attacker to know when an input-output fingerprint is being collected. Additionally, since the probe tokens are selected from the model's vocabulary a different subset of probe tokens can be used for subsequent fingerprinting, the attacker would need to distinguish when probe tokens are being used for fingerprinting and not for normal model inference. Additionally, the probe tokens can be sent in a random order or in between normal input-output sequences. Input-output pairs would only be known to the fingerprint generator.

3.4.3. Attacks Against RTT-Based Location Attestation

The primary threat is a location spoofing attack, where an adversary operating in a non-compliant jurisdiction (e.g., Country B) attempts to make the attester appear as if it is inside the certified jurisdiction (Country A).

Mitigation Strategy: This attack is fundamentally mitigated by the physical constraints of latency, which our protocol leverages:

1. Infeasibility of Reducing Latency: To successfully spoof its location, the attester in Country B would need to reduce its RTT to the LESs in Country A. This is physically impossible as it would require network signals to travel faster than the speed of light, which provides a hard lower bound on the latency.
2. Failure of "Delay Attacks": The only feasible network manipulation is a "delay attack" (e.g., via a relay) to inflate latency. As our protocol calculates the maximum possible distance from each LES, and policy relies on the closest LESs to provide the tightest geographic boundary, adding delay is counter-productive for the attacker. It would make the attester appear further away from the LESs, causing it to fail the jurisdictional check. As shown in our demo's policy, any delay greater than 5ms from the local LES induces an invalid certification.

Therefore, an attacker cannot spoof a valid but false location; they can only cause a denial of service by adding delay, which is a "fail-safe" security outcome. This, combined with the mutual hardware-rooted authentication that prevents simple node spoofing, makes the location attestation robust.

4. Evaluation of a Proof-of-Concept Implementation

4.1. Experimental Setup

To validate our architecture, we implemented a sovereign chat service, publicly accessible at nopinkypromisses.ai. The system was deployed on a testbed consisting of NVIDIA H100 GPUs and AMD EPYC 9V84 processors, running the meta-llama/Llama-3.2-3B-Instruct model on the vLLM inference engine [14]. We leveraged AMD's SEV-SNP and Nvidia's CC mode [13] to establish our trusted compute base (TCB). The implementation integrates the pod-as-TEE environment, the model fingerprinting probes, and the location attestation protocol to demonstrate the end-to-end generation of the AI Passport with every user query. All subsequent experiments were conducted on this testbed.

4.2. Performance Overhead

We conducted benchmarks comparing the system with Confidential Computing (TEE) enabled and disabled, focusing on both maximum throughput and serving latency.

Metric	Confidential Computing OFF	Confidential Computing ON	Performance Impact
Throughput (requests/s)	96.91	84.50	-12.8%
Total Tokens/s	24,805.19	21,630.46	-12.8%
Output Tokens/s	12,404.05	10,816.50	-12.8%

Table 2. Performance impact of Confidential Computing.

From Table 2, it can be seen that under maximum load, the system shows a throughput reduction of approximately 12.8% when running within the TEE, a bounded cost attributed to hardware-level memory encryption.

Under a non-saturated load of 10 RPS, the impact on throughput is negligible (<1%). User-perceived latency increases are minimal, with the median Time To First Token (TTFT) increasing by only 2.08 ms and median Time Per Output Token (TPOT) by just 0.59 ms. These increases are imperceptible to a human user. The results demonstrate that the cryptographic guarantees can be achieved with a highly acceptable performance trade-off. More detailed results on latency are shown in Table 3

Metric	Confidential Computing OFF	Confidential Computing ON	Performance Impact
TTFT (ms)	14.38	16.29	+1.91 ms
Median TTFT (ms)	14.15	16.23	+2.08 ms
P99 TTFT (ms)	17.63	19.30	+1.67 ms
TPOT (ms)	4.17	4.76	+0.59 ms
Median TPOT (ms)	4.15	4.74	+0.59 ms
P99 TPOT (ms)	4.54	5.04	+0.50 ms

Table 3. Latency performance impact of Confidential Computing.

4.3. Location Attestation Accuracy

To understand the relationship between signal round-trip-time (RTT) and distance between the network nodes (LES and the attester) we have used a set of 15 Virtual Private Routes between different data centers in the United States and Europe.

The measurements show a strong linear relationship between geographic distance and network latency as shown in Figure 2. The combined dataset, encompassing measurements from Los Angeles (LA1) to US destinations and Amsterdam (AM3) to European destinations, demonstrates a consistent distance-latency correlation with an R-squared value of 0.91. The linear regression yields a slope of approximately 0.91 ms/km, representing the effective propagation delay including network routing overhead. This observed relationship significantly exceeds the theoretical minimum imposed by the speed of light indicating a slowdown due to network infrastructure, routing protocols, and switching delays inherent even in private fiber network topology.

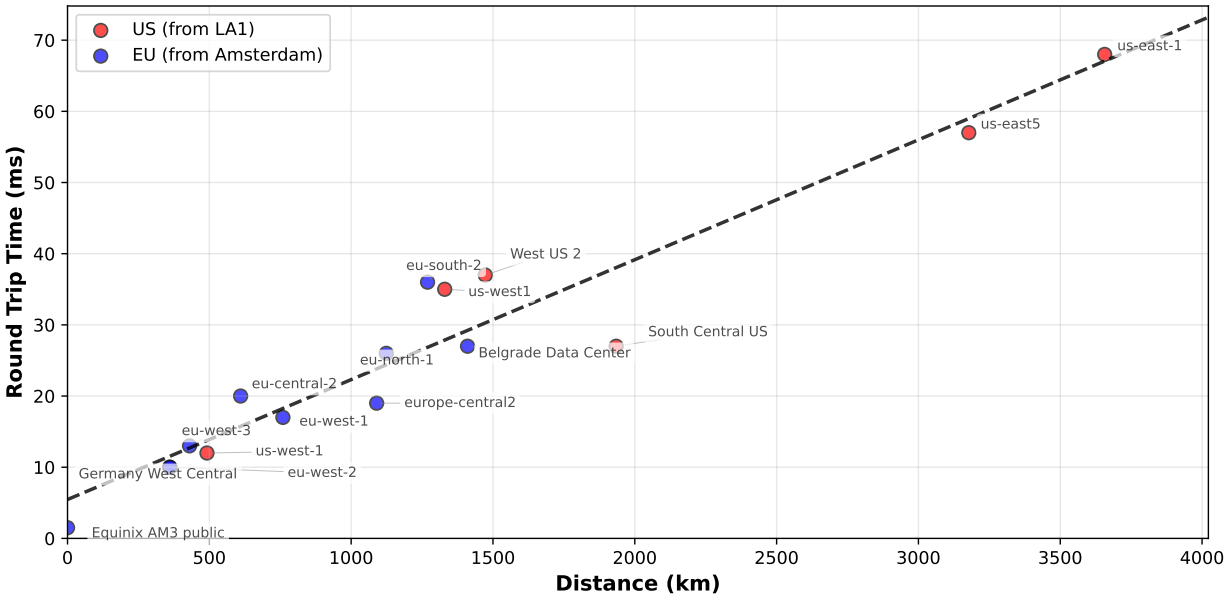

Figure 2. Round Trip Time vs Distance: US (LA1) and EU (Amsterdam) Data Centers

The model's predictive accuracy, characterized by a Mean Absolute Error (MAE) of 3.85 ms and Root Mean Square Error (RMSE) of 4.91 ms, demonstrates sufficient precision for geographic location attestation applications. The slightly higher RMSE compared to MAE suggests the presence of outlier measurements with larger deviations, likely attributable to variable network routing paths and traffic conditions. Notably, the error magnitude ($\leq$5 ms) remains substantially smaller than the latency differences between geographically distant regions, enabling reliable differentiation between locations separated by hundreds of kilometers. This precision level supports the feasibility of latency-based location verification systems for regulatory compliance applications, where distinguishing between countries or major geographic regions is the primary requirement rather than precise coordinate-level positioning.

Since we want the positioning system to be deterministic (and not depend on the accuracy of a specific model fit), for the purposes of the demo, we have colocated a LES with the attester and have introduced a policy that all delays higher than 5 ms induce an invalid certification. This way, assuming the speed of light in fiber optic medium of 200,000 kilometers per second [15], valid measurements of $\leq$5 ms confirm that the attester is less than 500 km from the LES.

4.4. Model Fingerprinting Sensitivity

Fingerprints are vectors of real numbers and so are comparable by any standard distance metric; our similarity metric is a length-normalized version of L1 similarity. Based on empirical analysis, similarity scores of greater than 0.85 in vector comparison demonstrate a derivative relationship across serialization, quantization, fine-tuning, or distillation. Similarity scores less than 0.85 demonstrate independent training and low likelihood of a derivative relationship between models.

In Table 4 is a sample of transformer-based models with similar size (in terms of number of parameters). The similarity scores are less than the threshold value of 0.85 which indicates independence between models and model families.

When comparing models that are known to be fine-tuned versions of a base model, such as when instruction tuning a pre-trained LLM for a chat application, the fingerprint similarity scores demonstrate a derivative relationship as shown in Table 5.

Our model fingerprinting method proves robust against common optimization techniques like quantization, which is crucial for real-world deployment. For example, several quantized versions of the Qwen3-32B model from two different providers all maintained a high fingerprint similarity of over 0.91 when compared to the original base model.

Model Provider	Model Name	Similarity
Cohere	CohereForAI/aya-23-35B	0.8442
Meta	CodeLlama-34b-Instruct-hf	0.7187
Google	Gemma-3-27b-pt	0.6334
AllenAI	OLMo-2-0325-32B-Instruct	0.6269

Table 4. Comparison of model similarity scores. Model Provider: Qwen, Base Model: Qwen3-32B (bf16)

Base Model	Fine-tuned (Instruct) Model	Similarity
baidu/ERNIE-4.5-0.3B-Base-PT	baidu/ERNIE-4.5-0.3B-PT	0.9791
google/gemma-2-9b	google/gemma-2-9b-it	0.9752
mistralai/Mistral-7B-v0.3	mistralai/Mistral-7B-Instruct-v0.3	0.9672
bigcode/starcoder2-15B	bigcode/starcoder2-15b-instruct-v0.1	0.9601

Table 5. Comparison of base and fine-tuned model similarity.

4.5. AI Passport schema

The AI Passport represents our primary architectural innovation: a unified, end-user verifiable artifact that extends trust beyond the traditional tenant-only attestation model. It is a structured, signed data token, conceptually similar to the Entity Attestation Token (EAT) being standardized within the IETF RATS working group [16] The passport contains a set of verifiable claims that provide a holistic statement of sovereignty for the AI transaction as shown in Figure 3.

- **Subject**: A unique identifier for the Attester TEE instance.
- **Issuer**: The identity of the Verifier that issued the passport.
- **Timestamp**: The time of issuance to ensure freshness.
- **Policy ID**: An identifier for the appraisal policy used by the Verifier.
- **Confidentiality Claims**:
  - tee_type: The type of TEE used (e.g., "AMD-SEV-SNP").
  - boot_measurement: The SHA-256 hash of the application source code that was verified.
  - attestation_quote: The raw, signed attestation quote for full auditability.
- **Integrity Claims**:
  - model_name: The name of the AI model (e.g., "Llama-2-7b-chat").
  - model_precision: The level of precision of the model weights (e.g., bfloat16).
  - model_fingerprint: The calculated Input-Output fingerprint.
  - model_arg_values: Inference arguments that the model was run with (e.g., temperature = .8)
- **Location Claims**:
  - jurisdiction: The asserted jurisdiction (e.g., "DE", "FR", "EU").
  - location_evidence: An array of the signed RTT reports from the LESs used to make the jurisdictional determination.

Figure 3. AI Passport Schema

5. Discussion and Future Work

We have presented an architecture for Sovereign AI processing that moves beyond contractual guarantees to provide verifiable, hardware-rooted cryptographic proofs of confidentiality, integrity, and location. Our proof-of-concept implementation demonstrates the feasibility of integrating these technologies into a cohesive system that addresses the trust gap in modern cloud infrastructure.

New and expanded regulations bring additional attention to and the need for provable digital sovereignty for advanced AI systems. The EU AI Act's mandates for high-risk systems, where the AI Passport provides auditable transparency into model integrity and processing location, fulfilling Article 13's documentation requirements. Similarly, it supports GDPR's data protection principles (Article 25) by enforcing confidentiality and localization through hardware-rooted proofs, reducing reliance on contractual transfers under Article 44. This enables compliant deployment in sectors like healthcare, where sovereignty violations could trigger fines.

The evaluation results quantify the trade-offs of this approach. The performance impact is primarily concentrated in a modest 12.8% throughput reduction under maximum load due to the TEE's hardware-level encryption. Crucially, this overhead is not reflected in user-perceived latency, where increases in response time are negligible, making the trade-off highly favorable for interactive applications. The cryptographic guarantees for location and model integrity introduce minimal overhead as they are performed during the attestation and initialization phases, avoiding any performance degradation on the critical "hot path" of individual inference requests. This architecture effectively shifts the security and sovereignty verification to a one-time, pre-computation step, ensuring that the core AI processing remains efficient.

These results open up several avenues for future work. While our proof-of-concept validates the architecture on a single deployment environment, comprehensive validation across multiple cloud providers (AWS, Azure, GCP) and geographic regions (particularly Asia-Pacific) would strengthen confidence in the system's portability and jurisdictional coverage. Our focus is on achieving higher-scale governance at the cluster level, a challenge motivated by the desire to eliminate even minor performance impacts on the critical path for high-throughput workloads. Another ambitious direction is to expand our certification framework to address the provenance of AI models themselves. Building on our ability to validate the model, location, and compute, the next logical step is to further develop the AI Passport capability. Such a system could issue a verifiable "AI passport," cryptographically certifying a model's origin, lineage and provenance. This would be particularly relevant for sensitive government applications, enabling rapid verification that a model was trained by a specific nation-state or an allied partner, thereby adding a critical layer of supply chain security to the Sovereign AI ecosystem.

References

[1] 451 Research, Voice of the enterprise: Digital pulse, workloads & key projects 2020, Tech. rep., S&P Global Market Intelligence (2020).
[2] G. Even, Microsoft refuses office 365 data details to police scotland over privacy law, WebProNews (Aug. 2025).
[3] D. Bradbury, Openai forced to preserve chatgpt chats, Malwarebytes Labs (Jun. 2025).
[4] 115th U.S. Congress, Clarifying lawful overseas use of data act, h.r. 4943, public law 115-141, division v (2018).
[5] V. Costan, S. Devadas, Intel sgx explained, IACR Cryptology ePrint Archive 2016 (086) (2016).
[6] P.-C. Cheng, et al., Intel tdx demystified: A top-down approach, ACM Computing Surveys 56 (9) (2024) 238. doi:10.1145/3652597.
[7] Confidential Containers Community, Introduction to confidential containers (coco), confidentialcontainers.org (Feb. 2024).
[8] I. Ahmad, et al., A novel approach for securing machine learning models using blockchain, arXiv preprint arXiv:2301.12333 (2023).
[9] Y. Wen, et al., Adversarial watermarking transformer: Towards tracing text provenance with data hiding, in: Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.
[10] R. Cuevas, et al., Is ip geolocation reliable? a case study on the accuracy of geolocation databases, in: Proceedings of the ACM SIGCOMM conference on Internet measurement, 2011, pp. 447–462.
[11] S. Sasy, S. Gorbunov, C. W. Fletcher, Zerotrace: Oblivious memory primitives from intel sgx, in: Proceedings of the Network and Distributed System Security Symposium (NDSS), 2018.
[12] C. Kocaoğullar, et al., A confidential computing transparency framework for a comprehensive trust chain, arXiv preprint arXiv:2409.03720v2 (2024).
[13] AMD, Sev-snp: Strengthening vm isolation with integrity protection and more, White paper, AMD (Jun. 2020).
[14] Meta, Llama 3.2 3b instruct model card, Hugging Face (Sep. 2024).
[15] D. Halliday, R. Resnick, J. Walker, Fundamentals of Physics, 10th Edition, John Wiley & Sons, 2013.
[16] H. Birkholz, et al., Remote attestation procedures (rats) architecture, RFC 9334, Internet Engineering Task Force (IETF) (Jan. 2023).