Hardware-Enabled Verifiability Standards in the Age of AI – A comprehensive Analysis

Supervised Program for Alignment Research — Fall 2025

Mentor: Onni Aarne — onni@iaps.ai

Mentee: Fernando Ruiloba Portilla — feruiloba@gmail.com

1. Abstract

This research examines how hardware-enabled verifiability (HEV), particularly through confidential computing (CC), can serve as a foundation for AI verifiability in data centers. Through analysis of existing security standards and specifications, this work aims to lay the groundwork for the future creation of minimum verifiability requirements that can inform policy and data center design for HEV.

2. Introduction

2.1. Background

2.1.1. AI risks that motivate the need for verification

As AI models become more capable and integrated into high-stakes domains, they are driving major economic and scientific breakthroughs, but these same capabilities pose serious risks if misused. Advanced AI systems can now enable threats such as mass surveillance, large-scale cyberattacks, and the development of novel biological weapons. The growing power of these systems, combined with their increasing deployment in sensitive areas like biotechnology, cybersecurity, and autonomous research, means that the potential consequences of misuse, whether by external bad actors or through internal failures of oversight, are significantly more severe than in the past.

At the same time, frontier AI models represent billions of dollars in investment and intellectual property, making them high-value targets for theft or exfiltration. The combination of immense capability and immense value creates a landscape where robust verification mechanisms are no longer optional: they are essential for ensuring that AI development and deployment remain safe, accountable, and aligned with both regulatory requirements and public expectations.

Model and Prompt Authenticity

One of the most pressing concerns is the potential for silent model substitution or downgrading. AI companies could swap out models or serve cheaper alternatives without informing their customers, and users would have no reliable way to detect this. This problem also affects safety: without verification mechanisms, there's no proof that the models companies deploy to customers are the same ones that underwent safety evaluations.

There's also the risk of unauthorized system prompt modifications. The behavior of AI systems can shift dramatically through prompt changes, whether from company decisions or rogue employees, without users' knowledge or consent.

At a fundamental level, enterprises integrating AI systems currently have no cryptographic assurance that the model serving their requests actually matches what was promised in contracts. This leaves them vulnerable to silent alterations in inference infrastructure with no reliable way to detect such changes.

Training Data Provenance

The question of what data was used to train AI models has become a significant legal and ethical battleground, with ongoing disputes over unauthorized use of copyrighted content. It is currently very difficult to verify claims about training data provenance.

Beyond copyright concerns, there are security risks embedded in the training process itself. Training-time backdoors can be introduced through data poisoning, weight perturbation, or aggregation attacks. Without verifiable records of the training process, detecting such manipulations becomes extremely difficult.

Internal Deployment Security

Frontier AI companies often run highly capable models internally for extended periods, particularly in sensitive domains like biotechnology, cybersecurity, or AI research and development. During these periods, models could learn harmful behaviors, be misused by insiders, or potentially leak their own code when oversight is insufficient.

These risks are amplified by the reduced visibility that characterizes internal deployments. Such systems often have greater access to computational resources, fall outside many public safety checks, and are less visible to external auditors or regulators. External parties currently have limited means to verify that internal safeguards are actually being implemented and followed.

Model Weight Security

Billions of dollars of investment flow into developing frontier AI models, making their weights an extremely valuable target. Current protection against model weight exfiltration is often perceived as inadequate. The attack surfaces are numerous: compromised host systems, network taps, and even front-channel attacks that can exfiltrate data through inference tokens.

Without hardware-based security guarantees and verified monitoring, detecting and preventing such exfiltration attempts remains challenging.

2.1.2. Benefits of AI verifiability

At the same time, improving verifiability could provide some benefits.

Security Benefits

Hardware-enabled verifiability supports security directly by enabling verification of the integrity of data center components and allowing encryption of traffic inside the data center. This helps secure AI development and deployment against supply chain attacks and other cyber threats. By ensuring that firmware and software have not been tampered with, verifiability provides a foundation for trusting that the entire system is operating as intended.

Commercial and Trust Benefits

Current regulatory frameworks and voluntary commitments rely heavily on self-reporting by AI companies. There is limited operational ability to independently verify companies' claims about evaluation integrity or training data provenance. This creates a trust gap where regulators and the public must largely take companies at their word.

Verifiable confidentiality claims make it easier to attract foreign customers to US-based cloud and AI services, since those customers can verify that their data will remain secure and confidential without needing to simply trust the data center operator's word. This same dynamic helps create broader consumer trust and can accelerate adoption of AI services generally. When users can cryptographically verify that their data is being handled as promised, they become more willing to move sensitive workloads to the cloud.

Cross-Border and Export Benefits

Verifiability facilitates cross-border use of cloud and AI services by creating mutual trust between parties who might otherwise be reluctant to share sensitive data or intellectual property. US AI companies can more securely deploy their models abroad when they can verify that foreign data centers meet security requirements. Similarly, importers of US chips can demonstrate compliance with export controls by verifying their stated end use. This creates opportunities for commercial activity that would otherwise be too risky.

Oversight and Accountability Benefits

Verifiability enables privacy-preserving oversight by allowing integrity verifications to be shared with third parties. Users can prove specific security claims about a data center, or demonstrate what code is running or has been run on particular devices, without exposing the underlying confidential data. This allows governments to verify that AI developers are meeting security standards without requiring full access to proprietary systems.

Strategic Benefits

Overall, verifiability enhances security against cyber attacks while simultaneously creating the trust necessary for wider commercial deployment of AI models and more extensive use of cloud services. This combination can strengthen technological dominance by making it safer and more attractive for customers worldwide to use services built on verified infrastructure.

2.1.3. Existing mechanisms for AI verifiability

Existing literature points to a range of mechanisms that could enable meaningful verification of international AI agreements. These mechanisms span multiple layers: some operate directly on AI chips through hardware security features, others attach to the surrounding infrastructure through network taps and analog sensors, and still others rely on human-centered approaches like whistleblower programs and personnel interviews. No single mechanism provides complete assurance on its own; each has limitations that sophisticated actors might exploit. However, when combined across largely independent layers, these approaches create substantial redundancy. The following sections describe each mechanism identified in the research literature, along with its key limitations. [1]

2.1.3.1. On-Chip Mechanisms

- **Hardware-backed workload certificates and evaluations** use security features built directly into AI chips, such as secure boot and Confidential Computing, to cryptographically sign certificates confirming how computations were performed. These certificates can attest that a trained model was produced using specific code and data, or that outputs were generated by a specific model on specific inputs. The main limitations include unsolved technical problems, severe security challenges with untrusted suppliers, a mixed track record of hardware security being broken by researchers, performance-security tradeoffs that incentivize weaker security, and the inability to patch hardware-level vulnerabilities after manufacturing.
- **Tamper-proof, compliance-locked AI chips** would refuse to execute non-compliant workloads and resist physical tampering, ensuring that even undeclared compute clusters could not be used for violations. However, tamper-proofing faces major technical challenges, raises concerns about adversarial use and power concentration, and requires tracking chip fabrication facilities through non-technical means.

2.1.3.2. Off-Chip Mechanisms

- **Network taps and analysis** involve mutually vetted devices that intercept data exchanged between AI chips or servers, then check for discrepancies with declared uses. This approach could be retrofitted to existing hardware and optimized for security. The limitations are that the strongest versions require redesigning proprietary chip-adjacent hardware, massive data rates create storage challenges, encrypted workload data poses obstacles, and substantial random sampling or real-time processing is necessary.

- **Partial workload re-execution with constraints**, sometimes called proof-of-learning, uses trusted chips to re-run randomly sampled segments of AI training or deployment to verify that claimed results are reproducible. Constraints like verifiably random initializations and data orders help rule out spoofed declarations. Limitations include vulnerability to various attacks such as glue-ing, synthetic data manipulation, and malicious code; the heuristic nature of defenses leaves room for novel attacks; replicability challenges from hardware randomness; and incompatibility with online learning approaches.
- **Backup evaluations on separate chips** involve trusted, physically secured compute clusters running tests to verify that declared models, data, and code have required properties such as passing safety evaluations. The main limitations are the logistical challenges of building mutually secured trusted clusters and dependence on the quality of the evaluations themselves.
- **Off-chip analog sensors for compute accounting** log physical measurements like power draw, temperature, and electromagnetic readings from AI chips. Analysis then verifies that the total operations performed match what was declared. Limitations include unsolved technical problems, highly variable chip utilization rates, complex relationships between utilization and physical measurements, potential for spoofed physical signatures, the need to manufacture and install millions of sensors, and significant error bars that make this approach best suited for detecting large-scale violations.
- **Verifying AI chips' chain of custody** tracks the locations and owners of random samples of AI chips from manufacturing through destruction, using inspections, video cameras, and hard-to-spoof unique identifiers. This prevents large quantities of chips from being assembled into undeclared clusters. Limitations include challenges with retroactively verifying existing chips and the logistical complexity of tracking millions of devices.
- **Tamper-proof, compliance-locked AI servers** extend compliance-locking to entire server enclosures rather than individual chips, using off-chip mechanisms to detect non-compliance. The limitations are similar technical challenges in implementing tamper-proofing and the same concerns about enforcement tool abuse.

2.1.3.3. Personnel-Based Mechanisms

- **Whistleblower programs** formally enable and incentivize employees to report violations through anonymous channels, financial rewards, legal protections, and regular in-person contact with verifiers. The effectiveness is unclear because it depends on the number of accomplices involved in a violation and their loyalty, and sophisticated compartmentalization might prevent disclosures. Secure communication channels require in-person visits, and states might suppress whistleblowers.

- **Interviews of personnel** may reveal violations unintentionally through inconsistencies or statements that contradict declarations, even when interviewees are not intentionally whistleblowing. The effectiveness is unclear because it depends on accomplices' ability to lie convincingly, interviewees might collude, and no reliable lie detection technology exists.
- **National intelligence activities** leverage human intelligence, signals intelligence, and cyber intelligence to detect violations without requiring the prover's cooperation. Limitations include that this approach is more adversarial, findings are harder for third parties to verify, effectiveness is unclear, and capabilities vary significantly across states.

2.1.3.4. Supplementary Mechanisms

- **Supplementary constraints** are additional heuristic tests that complement partial workload re-execution, such as checking that model checkpoint performance fits scaling law predictions or that gradients are not suspiciously similar. These can be circumvented and only supplement other mechanisms.
- **Data and code validation** tests whether declared data and code have been maliciously engineered to circumvent verification, using methods like frequency analysis, data tracing, code factorization constraints, and inspection by trained AI models. Limitations include that validation methods have their own challenges and cannot complete any verification subgoal independently.
- **Financial audits** check whether organizational cash flows are consistent with declared activities. Documents could be forged, though this becomes harder with personnel-based mechanisms in place.
- **Sanity checks** assess whether claimed activities are plausibly consistent with an organization's objectives and incentives. This provides only weak evidence since suspicious activities might have legitimate explanations.
- **Workload classification** uses high-level measurements like power draw patterns to distinguish whether a workload is AI training or inference. This is too coarse-grained to verify most specific rules and cannot distinguish between different types of training or deployment.
- **Analytic output verification** quickly verifies outputs' authenticity by examining them directly for workloads like cryptocurrency mining where this is mathematically possible. This is only applicable to specific workload types and not to general AI workloads.
- **Weaker, heuristic evaluations** supplement higher-confidence evaluations with more uncertain tests. They have lower confidence and may produce false positives or negatives.
- **Inspections for undeclared AI clusters** physically examine suspect sites identified through other mechanisms. They struggle to find well-hidden data centers such as those underwater, underground, or disguised as other facilities, and are challenged by decentralized AI development.

- **Information from data center suppliers** uses public statements and documentation from companies supplying specialized data center components to reveal hidden facilities. Suppliers might be compromised or colluding, and documentation can be forged.
- **Cameras at AI data centers** ensure chips are not tampered with or diverted using tamper-evident security cameras that are checked on-site. The cameras themselves must be tamper-evident and the approach can be circumvented with sophisticated attacks.
- **Design information verification** verifies facility designs during construction to ensure there are no hidden rooms or infrastructure that could store undeclared compute. This requires extensive inspections during construction and cannot easily verify historical facilities.
- **Satellite or aerial images** help identify hidden data centers through visual and infrared imaging. They cannot detect underwater or well-disguised facilities, cannot determine how facilities are actually being used, and require ground-truth verification through inspections.
- **Open-source intelligence** uses social media posts, news reports, and other public information to reveal violations such as hidden data center construction or impacts of non-compliant deployment. The impacts of violations might not be evident before harm occurs, internal deployments may produce no visible societal effects, and distinguishing compliant from non-compliant AI advances is difficult.

2.2. Technical Underpinnings

Some benefits of on-chip verification are that much of the necessary functionality is already present in advanced AI chips without compromising user privacy or introducing insecure “back doors”. For this reason, we mainly focus on hardware-enabled verification.

2.2.1. Platform Integrity Verification

A foundational concept in trusted computing is the verification of the underlying platform's integrity. This process ensures that the hardware, firmware, and software haven't been maliciously tampered with or accidentally altered. Platform integrity generally consists of cryptographic measurement, calculating a hash of executables or configurations, and firmware/configuration verification, where attestations are used to confirm the platform is in an authorized state. In some advanced implementations, it also includes recovery mechanisms to roll back firmware to a "known good" version if verification fails.[2]

2.2.1.1. Hardware Security Module (HSM)

An HSM is a physical computing device designed to safeguard cryptographic keys and provide specialized cryptographic processing. These modules typically host operations like encryption, decryption, and signature generation, often using hardware-accelerated mechanisms.

A Trusted Platform Module (TPM) is a specific type of standalone HSM used to protect sensitive information such as passwords and hash measurements.[2]

2.2.1.2. Chain of Trust (CoT) and Secure Boot

2.2.1.2.1. Chain of Trust (CoT)

A chain of trust is a security mechanism that establishes and maintains trustworthy system boundaries by relying on transitive trust: each stage of the boot process verifies and measures the next stage before handing over control. These measurements are immediately recorded in a secure root-of-trust storage (such as a TPM or HSM register) so that the system's state can later be attested. This process does not need to stop at firmware; it can be extended into the operating system and application level, allowing software, devices, and other components to be measured and verified as well. [2]

Every chain of trust begins with a root of trust (RoT), which consists of hardware and/or firmware components that are implicitly trusted. In many systems, this root is anchored in immutable ROM code, though this is not universal. The RoT is typically split into two roles: one part verifies the next component before execution, and another measures components and records those measurements. The first measuring component, called the Core Root of Trust for Measurement (CRTM), records the initial measurement into the TPM, while the Core Root of Trust for Verification (CRTV) verifies the first component before control is transferred. [2]

The measurement root itself can be divided into a static part and a dynamic part. The static CRTM establishes a fixed chain of trust at boot time, producing measurements that remain consistent across reboots (aside from volatile data like time). The dynamic CRTM allows a chain of trust to be re-established while the system is running, without requiring a reboot. [2]

Roots of trust implemented in hardware are harder to tamper with but cannot easily be patched, while firmware-based roots are more flexible but easier to modify or compromise. For this reason, the root of trust should be kept as small and simple as possible. [2]

2.2.1.2.2. Secure Boot

Secure Boot is a foundational hardware security feature designed to ensure that a device only runs trusted, manufacturer-approved software. It acts as a digital "gatekeeper" that checks the credentials of software before allowing it to start. It works through asymmetric cryptography:

1. The manufacturer creates a public/private key pair

2. The public key is stored in read-only memory on the device

3. Firmware is signed with the private key

4. On boot, the device uses the public key to verify the firmware's signature, refusing to load unsigned or improperly signed code

A key distinction from remote attestation: secure boot only requires protecting the public key from being overwritten; no secrets are stored on the device. Remote attestation, by contrast, requires the chip to hold its own private key securely, since the chip must sign outputs to prove their authenticity. If that private key is exposed, attestations can be forged. [3]

2.2.1.3. Supply Chain Verification

Organizations face increasing risks of supply chain compromise, including counterfeiting, unauthorized production, and the insertion of malicious hardware or software.

Technologies designed for supply chain protection help verify the authenticity and integrity of platform hardware from the moment of assembly to arrival at the customer's data center. Some advanced methods involve locking the boot process until the customer provides a specific secret known only to them and the manufacturer. This verification of platform attributes is a critical component in mitigating risks associated with poor manufacturing or development practices in the global cyber supply chain.[2]

2.2.2. Attestation

Attestation allows hardware to make verifiable claims about its current state to external parties. In complex modern data centers, servers are composed of numerous subsystems and peripherals from various global suppliers, and attestation provides a standardized, automated mechanism to confirm that these components are in a known, trustworthy state and have not been tampered with.

The terminology proposed by the IETF proposes the roles of a relying party which establishes a secure connection with an attester by utilizing a verifier to validate the system's integrity. To do this, the attester collects and presents "claims", specific pieces of information, such as cryptographic hashes of its application code, that represent the current state of its hardware and software. These claims provide the necessary evidence to prove that the device is running the intended, untampered stack before any trust is granted. [4]

The process works as follows:

1. The chip signs measurements of its currently loaded firmware (and other state information) using its own private key
2. This signature is sent to a verifier (e.g., the manufacturer)
3. The verifier checks the signature to confirm the chip is running approved firmware or has a valid "operating license"

Attestation mechanisms can be classified by the nature of the relationship between the prover and verifier. During local attestation, the prover proves its identity to a verifier on the same device. During remote attestation, different devices establish trust by delivering cryptographic evidence that the software in operation is authentic and untampered. Finally, mutual attestation provides enhanced security for trusted applications by requiring both ends of a communication channel to verify each other's integrity. [4]

Attestation mechanisms can also be classified by the layer of the technology stack in which they operate. See the table below. [4] Hardware based solutions sometimes rely on Trusted Platform Modules

Approach	Description	Considerations
Software based	No special hardware required; relies on timing or software techniques	Lower cost but weaker security guarantees
Hardware based	Uses tamper-resistant hardware (TPM, hardware secrets fused in processor)	Stronger security but requires specific hardware
Hybrid	Combines hardware and software elements	Balances security and flexibility

2.2.3. Trusted Execution Environments

2.2.3.1. Operation

Trusted Execution Environments (TEEs) are isolated areas within a processor designed to protect the data and code running inside them from being accessed or changed by other system components. The primary difference between security modules and TEEs is that TEEs establish a secure environment directly on the main processor cores, whereas a security module is a distinct, lower-performance processor dedicated to security tasks. While security modules are often sufficient for guarding highly sensitive data, TEEs offer an extra layer of protection for the primary computational tasks handled by the chip. [3]

TEEs are commonly used to prevent data within the environment from being monitored or interfered with by other parts of the system, such as malware, other users, or a cloud provider's platform software. For on-chip governance, TEEs allow a chip to remotely attest to its current state and the code it is running, allowing these claims to be verified by third parties. [3]

This capability enables specific types of privacy-preserving collaboration, such as multi-party computing. For instance, one party could set up a TEE on a chip and provide proof to another party regarding the specific code loaded inside. The second party could then transmit encrypted data to the TEE to be processed; the results are shared, but the original party never gains access to the unencrypted data. This method could potentially allow a third-party evaluator to test an AI model without ever having direct access to its unencrypted weights. [3]

TEEs might also be helpful for creating privacy-preserving logs of information during the training process, allowing for retrospective inspections. A recent paper describes a protocol for verifying that AI training follows specific rules, such as the amount of data, compute, or the specific process used. In this proposal, weights on a chip are hashed and signed at random intervals during training, and these hashes are recorded. These logs could later be used to identify which chips were used to train a specific model and to verify the model's provenance by providing and replicating training transcripts from the organization that performed the training. [3]

2.2.3.2. Isolation hierarchy

2.2.3.2.1. Application Isolation

Application isolation uses TEEs to protect individual applications, restricting the trust boundary to the CPU itself. Separate enclaves with unique per-application keys can shield sensitive workloads even from malicious insiders with platform access. However, this approach requires developers to integrate TEE toolkits and take responsibility for secure enclave design.

Examples:
- Intel Software Guard Extensions (SGX)
- Arm Confidential Compute Architecture (CCA)
- Arm TrustZone Trusted Execution Environment (TEE) for Armv8-A
- Arm Realm Memory Isolation and Protection
- IBM Application Isolation Technology

2.2.3.2.2. VM Isolation

Virtual Machine (VM) isolation encrypts each virtual machine's memory using unique per-VM keys managed by the CPU, removing the hypervisor from the trust boundary. While hardware-assisted virtualization already separates VMs, memory remains unencrypted and the virtual machine manager (VMM) must be trusted. With full VM isolation, even a compromised hypervisor or rogue firmware cannot access VM contents. This protects multi-tenant cloud workloads from malicious insiders and co-tenant data leakage, including when VMs serve as container worker nodes.

Examples:
- Intel Trust Domain Extensions (Intel TDX)
- AMD Secure Encrypted Virtualization (SEV)
- Arm Confidential Compute Architecture (CCA)
- IBM VM Isolation Technology

2.2.3.3. Main advantages for verification

- this scheme allows the Prover to demonstrate the compliance of all computations that take place within a designated portion of their infrastructure. Furthermore, in this scheme the central compliance tests are run against the plaintext data and not a noisy proxy such as electric power draw or heat emissions, therefore allowing detailed compliance checks rather than broad guesses about activities
- This stack avoids the centralization of power, since the Prover remains in control of their own hardware and data.
- This stack also allows the Prover to demonstrate their compliance with a large number of different requirements. This infrastructure approach allows the Prover to build a single system which can allow them to demonstrate their computations' compliance with global rules

2.2.4. Privacy-preserving computation

2.2.4.1. Memory Encryption

Most encryption solutions protect data at rest but leave it vulnerable when decrypted for processing. In shared cloud environments, applications running in memory can be exposed to attacks from co-located workloads or compromised administrators. Hardware memory encryption technologies close this gap by keeping data protected even while in use. [2]

Examples:

- Intel TME and Multi-Key Total Memory Encryption (Intel MKTME)
- AMD Secure Memory Encryption (SME)/Transparent Secure Memory Encryption (TSME)
- Arm Realm Memory Isolation and Protection
- Arm External Memory (DRAM) Encryption and Integrity with CCA
- IBM Memory Isolation Technology

2.2.4.2. Secure Multi-Party Computation (MPC)

Allows multiple parties to compute a shared function without any participant learning the others' private inputs, typically relying on secret sharing protocols. Common use cases for MPC include end-to-end encrypted relational databases, secure statistical analysis in languages like R, cryptographic key management, and private bidding or auctions. [5]

2.2.4.3. Homomorphic Encryption (HE)

Is a family of encryption schemes that allow mathematical operations to be performed directly on encrypted data so that the result, once decrypted, matches the output of operations performed on the original cleartext. It is frequently applied in the medical domain for tumor detection in MRI scans, in cloud storage analytics, and for private set intersection or information retrieval protocols. [5]

2.2.4.4. Differential Privacy (DP)

Provides a mathematical standard for output privacy by quantifying and limiting the information leaked about individual records during aggregate data releases. Real-world deployments include Google Chrome and Apple's iOS for collecting usage statistics, Microsoft's telemetry data collection, and the U.S. Census Bureau's release of census results. [5]

2.2.4.5. Zero Knowledge Proofs (ZKP)

Enable a prover to convince a verifier that a specific statement is true without revealing the underlying secret data. This technology is widely used in blockchain and cryptocurrency transactions, as well as for auditing and compliance tasks such as checking tax payments, loan risks, or aircraft maintenance records without disclosing the full sensitive datasets. [5]

2.2.4.6. Cryptographic acceleration

Uses specialized hardware coprocessors to offload encryption tasks from the main CPU. As encryption becomes standard for protecting sensitive data in data centers, these dedicated chips address the performance overhead that cryptographic operations impose on system resources. They are typically deployed as pluggable peripheral cards. [2]

2.3. Gaps

This paper aims to assess whether existing security standards and specifications are suitable for AI verification. For this purpose, it is beneficial to highlight the gaps already identified in existing literature, with particular attention to verification methodologies and confidential computing within AI.

2.3.1. AI Verification

Reuel A et al point to a set of open problems in technical AI Governance related to AI verification. They organize them by target. [6]

2.3.1.1. Data

- *Verifying datasets used to train a model:* Current proof-of-training-data approaches are not robust to all attacks (particularly small additions of harmful data), don't work for online or reinforcement learning, and require disclosing confidential information to verifiers. Membership inference attacks offer an alternative but need further development.
- *Verifying fair data use:* There is no established framework for verifying copyright compliance or that licensed data was used correctly. Formalizing verification of data licensing and proving exclusion of specific data from training sets remain unsolved.

2.3.1.2. Compute

- *Verifying chip location:* No reliable method exists to determine where a chip is located after export. Proposed approaches include measuring latencies to trusted servers, proof-of-work challenges for co-location, and mutual attestation, but none are mature. TEEs could potentially support mutual attestation between chips to verify co-location.
- *Designing hard-to-spoof chip IDs:* While unique identifiers can be engraved during fabrication or physical unclonable functions can be added, the security, feasibility, and performance trade-offs of these approaches remain open questions.
- *Verifying workload properties using TEEs:* TEEs could theoretically attest to program code and models being run, return signatures certifying computations were run as intended, or allow verifiers to confidentially run tests on model weights. However, firmware to implement this is lacking, security requirements are unusually high, and overhead costs may be prohibitive at scale.
- *Verifying workloads with a trusted neutral cluster:* Using hashed training snapshots with a neutral cluster for verification faces challenges in accounting for training randomness, building trustworthy clusters, and scaling to large models. This approach is proposed as an alternative when TEEs are unavailable or impractical.
- *Verifying large non-AI workloads:* Cluster owners may need to prove they ran non-AI workloads like climate simulations. Workload classification and proof-of-learning analogues for non-AI computation need exploration.

2.3.1.3. Models and Algorithms

- *Verifying model properties with full access:* Formal verification methods that mathematically prove system behavior exist but become prohibitively complex for large models. Verifying architecture and training procedures remains unsolved.
- *Tracking versioning and updates:* Modern AI systems change constantly, but it's unclear what information model registries should store, how to verify it, or how to track frequent post-deployment modifications.
- *Scalable proof-of-learning:* Methods exist to prove computational resources were expended to train a model, but scaling these techniques to foundation model training budgets remains impractical.
- *Adversarially robust proof-of-learning:* Current proof-of-learning methods are vulnerable to adversarial attacks that generate false proofs cheaply. Proposed countermeasures have only been tested against single attacks and only for language models.

2.3.1.4. Deployment

- *Verifying capabilities without full model access:* Zero-knowledge proofs can enable privacy-preserving verification, but computational overhead is too high for large models. GPU acceleration and proof splitting are promising but immature.
- *Verifying audit results at inference time:* TEEs are central to this proposal. A two-stage process would load inference pipelines into enclave computers, have auditors

evaluate the system, then produce certificates stored in a public registry. Users could then verify they're interacting with an audited model by requesting certificates with each generation. However, this requires all inference in enclaves, adds significant overhead, doesn't handle frequent model updates, and raises concerns about weight exfiltration by auditors.

- *Verifying safety measures post-deployment:* Regulators may need to verify that deployed models include required safety measures like output filters, but methods for auditing and enforcing this in safety-critical domains don't exist. TEE-based verified computing could potentially support this verification.
- *Developing robust watermarking:* Watermarks work better for images and audio than text due to continuous output spaces. Text watermarks lack robustness, and watermarks may be easy to fake between similar models.
- *Designing robust AI content detectors:* Current detection methods often fail independent evaluation, and as generative systems improve, distinguishing AI output from genuine media becomes increasingly difficult.
- *Utilizing verifiable metadata:* Standards like C2PA allow creators to certify content provenance, but metadata can easily be stripped by adversaries, undermining the approach.
- *Verifying AI-modified authentic content:* The binary distinction between AI-generated and human-generated content doesn't capture cases where authentic content is modified with AI tools. Detectors that distinguish these categories don't yet exist.

2.3.2. Confidential Computing

2.3.2.1. Interoperability and standardization

Interoperability between different TEE platforms remains challenging. Also, no universal attestation protocol exists across TEE technologies. [4] For example, RISC-V solutions use varying approaches without standardization.

2.3.2.2. Security Hardening for Adversarial Settings

- Are the hardware roots of trust in existing chips sufficiently robust to support governance needs, or will additional mechanisms need to be added to shore up crucial cybersecurity or physical security?

- Can a neutral mutually verified data center be realistically built, maintained, and used for extremely sensitive verification computations? [1]

- Can relatively simple hardware allow for a credible flow of cryptographic commitments to the Verifier in a way that robustly protects the security of the Prover? Key questions remain about how the cryptographic commitment scheme can be made robust within either existing or newly built hardware. The relative maturity of both core technologies—networking hardware and cryptographic commitments—should provide a strong basis for work on this front, but the abilities of this type of stack remain speculative, as it does not yet exist and will not come to exist without further research, engineering, and policy effort.

- Existing technologies need to be hardened before they can be relied upon in adversarial settings such as export control enforcement. On-chip governance mechanisms are only useful insofar as they reliably work even when adversaries are actively attempting to circumvent them. Commercial versions of these technologies are not typically designed to defend against a well-resourced attacker with physical access to the hardware. [3]
- ARM TrustZone lacks built-in attestation, creating security gaps.

3. Methods

This research employed a structured qualitative analysis to evaluate the feasibility of hardware-enabled verifiability (HEV) in data centers. The methodology was designed to explain, in a centralized document, the technical fundamentals, existing standards and high level policy needs required for HEV in data centers.

3.1. Source Selection

To find relevant standards, we queried Gemini Deep Research, Claude and ChatGPT on research mode. The AI services were prompted with context about hardware-enabled verifiability and hardware requirements for AI verification.

The resulting standards drew upon a diverse range of primary and secondary sources, including technical documentation from hardware vendors (e.g., Intel, AMD, ARM), policy documents regarding AI governance, standards organizations and existing industry specifications.

3.2. Categorization

After collecting standards from various sources, we used AI tools to organize standards by type and security layer, adding descriptions and relevance ratings for hardware-enabled verifiability. AI outputs were double checked by a human.

3.2.1. Standard types

We used the AI Standards Hub classification to group standards into five types. [7]

3.2.1.1. Foundational and terminology standards

These standards establish a common technical vocabulary to ensure conceptual alignment across all stakeholders. By defining core terms and principles, they provide the essential linguistic framework necessary for the development of more specialized technical requirements.

3.2.1.2. Process and management standards

This category outlines organizational governance and risk management methodologies. These standards provide repeatable procedural frameworks that ensure AI systems are developed and deployed in a safe, ethical, and consistently managed environment.

3.2.1.3. Measurement standards

Measurement standards provide the objective reference points and calibration units necessary for accurate technical evaluation. They ensure that performance metrics remain consistent and comparable across different systems, laboratories, and applications.

3.2.1.4. Product testing and performance standards

These standards define rigorous protocols and performance thresholds to validate system reliability and safety. They establish objective benchmarks and testing methodologies required to certify that an AI product is fit for its intended purpose.

3.2.1.5. Interface and networking standards

These standards focus on interoperability, ensuring that disparate AI systems and hardware components can integrate seamlessly. By facilitating technical compatibility, they maximize system utility while reducing operational friction and hardware redundancy.

3.2.2. Security Layers

We also classified the standards according to the security layers defined in NIST IR 8320 [2]. The layers are listed below.

3.2.2.1. Platform Integrity Verification

This layer focuses on the measurement and verification of the underlying platform's firmware and configuration to ensure they have not been tampered with or accidentally altered.

3.2.2.2. Software Runtime Protection Mechanisms

This layer involves hardware-based technologies designed to detect and block software runtime attacks, such as memory safety violations (e.g., buffer overflows) and control-flow hijacking (e.g., ROP/JOP attacks).

3.2.2.3. Data Protection and Confidential Computing

This layer utilizes hardware-enabled features like Trusted Execution Environments (TEEs) and memory isolation to protect data while it is actively being processed in memory.

3.2.2.4. Remote Attestation Services

This layer acts as a service to collate platform integrity measurements from across a data center, allowing an operator to verify and enforce security policies across thousands of servers.

3.3. Criteria for Baseline Requirements

3.3.1. AI Verification Requirements

3.3.1.1. Required Verification Tasks

- **Produce Certified Verification Code**: Both parties agree on what code performs verification through third-party audit, internal review, or automated software review in secure environments.
- **Produce Verified Records of Server Activity**: Build tamper-evident databases of server activity through either direct hardware attestation from main servers or random sampling and re-execution on separate secure servers.
- **Produce Verified Disclosures**: Analyze records inside secure environments to generate aggregate statistics and filtered information that answer questions without revealing individual records.

3.3.1.2. Mechanisms for Verification

There are a few mechanisms that can be used to perform the verification tasks described in the previous section.

For producing certified verification code, there are three mechanisms:

- **Manual third-party verification**
  An independent auditor inspects the source code and signs off on its hash. High trust, but slow and doesn't scale.
- **Verified internal code review**
  Multiple internal reviewers at the AI company review and co-sign the code. Keeps IP private but requires trust in the organization.
- **Confidential code review by trusted software**
  The code is reviewed inside a TEE by trusted software (possibly an AI agent), producing an attestation without exposing the code. More scalable but relies on trust in the reviewing software.

For producing verified records of server activity there are two mechanisms:

- **Directly attested record generation**
  The production servers themselves run in TEEs and **attest** that the logging software is running and recording everything correctly. Very strong guarantees, but expensive and not always practical at scale.
- **Subsampled record verification**
  If servers aren't trusted, you:
  - Log everything normally (untrusted),
  - Randomly sample some records,

    - Re-execute them in a separate TEE-based verification cluster,
    - If samples match, you accept the whole batch with statistical confidence.

  This is much cheaper and scales well but only gives probabilistic guarantees and can miss rare cheating.

For producing verified disclosures there is one mechanism.

- **Secure disclosure generation**
  Certified analysis code runs inside a TEE, reads the verified records, computes aggregates/statistics, and outputs only the final result, with an attestation proving it was computed correctly.

3.3.1.3. Verification Applications

- **Deployment authenticity:** Proving a specific model/prompt/toolchain produced an output
- **Adherence to deployment practices:** Proving monitoring against misuse, safety commitments
- **Training data provenance:** Proving what data trained a model (without disclosing it)
- **Datacenter security:** Protecting against weight exfiltration, side-channel attacks

3.3.2. TEE Requirements for Verification

3.3.2.1. Required Capabilities

For verification, TEEs must have the following capabilities:

- **Programmability**: The TEE can run arbitrary, user-specified code.
- **Attestability**: The TEE can provide cryptographic evidence about its identity and current state (software environment, memory image) that can be verified remotely.
- **Data confidentiality**: Hardware blocks other users (including the server host) from accessing TEE contents.
- **Code and data integrity**: Hardware prevents modification of code/data inside the TEE after initialization.

3.3.2.2. Technical Requirements

For the "directly attested record generation" mechanism mentioned above the following technical requirements must be present:

- A hardware root of trust anchoring a secure boot chain
- Low-level software (bare-metal hypervisor or OS) that enforces policies even against a root admin
- Runtime attestation that can measure software integrity without rebooting

For the “subsampled record verification” mechanism mentioned above, the baseline is:

- TEE-capable verification server(s)
- Certified sampling and re-execution code
- Cryptographic commitment (Merkle trees) from the prover's servers
- Remote attestation to bind outputs to certified code

3.4. Qualitative Analysis

Given the extensive range of standards and specifications, as well as time limitations, the qualitative analysis was conducted using AI tools to assess the adequacy of these standards in relation to the AI Verification and TEE requirements outlined in the baseline criteria. The AI models were prompted with questions such as “are there any standards that could help in the production of verified disclosures?”.

4. Findings

By searching for standards from the previously mentioned sources, we managed to compile a list of 84 standards with varying levels of relevance for hardware-enabled verification.

Table 1 shows the distribution of standards by organization. GlobalPlatform had the most standards mainly because it provides specifications related to TEE interfaces. IETF standards usually handle networking in the context of TEEs and TCG handles hardware modules like TPMs.

Organization Distribution

Organization	Count	Percentage	High HEV relevance
GlobalPlatform	17	20%	24%
IETF	16	19%	63%
ISO/IEC	11	13%	36%
TCG	8	10%	88%
DMTF	6	7%	33%
ETSI	4	5%	50%
OCP	4	5%	100%
PCI-SIG	4	5%	100%
NIST	3	4%	100%
Arm	2	2%	50%
AMD	1	1%	100%
CXL	1	1%	100%
Ecma	1	1%	0%
IEEE	1	1%	100%
Intel	1	1%	100%
JEDEC	1	1%	0%

OASIS	1	1%	0%
SLSA	1	1%	0%
UCIe	1	1%	100%

Table 1 Distribution of Standards by Organization

Table 2 below shows how relevant the compiled standards are with respect to hardware-enabled verifiability (HEV). Over half of them are very relevant to HEV.

HEV Relevance Distribution		
Level	**Count**	**Percentage**
High	46	54.80%
Medium	31	36.90%
Low	7	8.30%

Table 2 Distribution of Standards by HEV relevance

Table 3 below shows the distribution of security layers according to the number of standards, what percentage of standards belong to that layer, and the proportion of standards in that layer with high HEV relevance. It is important to mention that a standard might fit into more than one security layer, which is why the percentages don't add up to 100%. Only one runtime protection standard was identified, and it had limited relevance to HEV.

Security Layer Distribution			
Layer	**Count**	**Percentage**	**High HEV Relevance**
L1: Platform Integrity	30	36%	53%
L2: Runtime Protection	1	1%	0%
L3: Confidential Computing	34	40%	53%
L4: Attestation	28	33%	75%

Table 3 Distribution of Standards by Security Layer

Table 4 shows the distribution of standards by type according to the AI Standards Hub [7]. Over 50% of all standards involve some type of terminology definition, these standards are usually very relevant to HEV. Measurement standards, such as RFC 9711 (The Entity Attestation Token (EAT) are very relevant for HEV but only comprise 15% of all the standards.

Standard Type Distribution			
Type	**Count**	**Percentage**	**High HEV Relevance**
Foundational and terminology standards	42	50%	71%
Process and management standards	26	31%	54%
Measurement standards	13	15%	69%
Product testing and performance standards	10	12%	30%
Interface and networking standards	36	43%	50%

Table 4 Distribution of Standards by Type According to the AI Standards Hub

We allowed AI to define its own criteria, leading to a clear distinction called "Functional Domain Distribution". This distinction reduced class overlaps under the same standard. As shown in Table 5, attestation protocols make up a significant portion of the standard list. This outcome is logical, as attestation is essential for verification.

Functional Domain Distribution

Domain	Count	Percentage	High HEV Relevance
Attestation Protocols	17	20.2%	94%
Root of Trust	10	11.9%	80%
TEE APIs	10	11.9%	0%
TEE Architecture	7	8.3%	100%
Secure Interconnect	7	8.3%	71%
Supply Chain/Provenance	7	8.3%	14%
Secure Messaging	6	7.1%	33%
AI Security Framework	6	7.1%	67%
TEE Management	5	6.0%	80%
Certification	5	6.0%	20%
Cryptographic Requirements	5	6.0%	40%
Secure Update	3	3.6%	33%
Key Management	2	2.4%	50%
Runtime Protection	1	1.2%	0%
Device Identity	1	1.2%	100%

Table 5 Distribution of Standards by Functional Domain

Another important metric for analyzing standards in the context of AI verification is the type of hardware that the standards target, since GPU is increasingly critical for modern AI workloads and may require distinct security and attestation mechanisms compared to traditional CPU or mobile device standards. Table 6 shows that most standards (86%) are not GPU-specific.

Hardware Target Distribution

Target	Count	Percentage
CPU-focused	19	23%
GPU-applicable	12	14%
AI-specific	6	7%
Accelerator-agnostic	47	56%

Table 6 Distribution of Standards by Hardware Target

5. Analysis

The analysis of 84 standards reveals both the maturity of foundational security technologies and significant gaps that must be addressed before hardware-enabled verifiability can reliably support AI governance. This section examines where current standards fall short and proposes minimum verifiability requirements based on the findings.

5.1. Quantitative Analysis

The most striking finding is the hardware target mismatch between existing standards and AI verification needs. Only 14% of surveyed standards are GPU-applicable, while 56% remain accelerator-agnostic and 23% focus exclusively on CPUs. This distribution reflects the historical origins of confidential computing in enterprise and mobile security, where CPUs and general-purpose processors dominated. Modern AI workloads, however, rely heavily on GPUs and specialized accelerators such as TPUs and custom ASICs for both training and inference. The absence of GPU-specific TEE protocols, attestation mechanisms, and memory encryption standards represents a critical gap that existing frameworks do not address.

Interoperability remains a persistent challenge across TEE implementations. While attestation protocols constitute 20% of all surveyed standards and demonstrate the highest HEV relevance at 94%, no universal attestation protocol exists that functions across Intel SGX, AMD SEV, ARM TrustZone, and emerging RISC-V solutions. Each vendor maintains proprietary attestation infrastructures: Intel offers EPID and DCAP, AMD relies on its chip endorsement key architecture, and ARM TrustZone notably lacks built-in attestation entirely. This fragmentation means that a verifier seeking to audit AI workloads across a heterogeneous data center environment must implement and trust multiple distinct verification pathways, each with different security assumptions and threat models.

Runtime protection standards are notably underrepresented. The survey identified only one runtime protection standard, and it carried low HEV relevance. This gap is particularly concerning for AI verification, where the integrity of running workloads matters as much as the integrity at launch time. While TEEs provide isolation during execution, the ability to continuously verify that workloads remain uncompromised throughout long training runs or persistent inference deployments requires runtime attestation capabilities that current standards do not adequately specify.

The standards landscape also reveals insufficient coverage of supply chain verification for AI-specific hardware. While 8.3% of standards address supply chain and provenance concerns, only 14% of these carry high HEV relevance. Given that AI verification must ultimately trust that chips themselves have not been tampered with during manufacturing or distribution, the lack of robust, AI-hardware-specific supply chain standards creates a foundational vulnerability that undermines higher-layer verification mechanisms.

5.2. Qualitative Analysis

5.2.1. Standards Adequacy for AI Verification Requirements

This section assesses whether the 84 surveyed standards adequately support the AI verification requirements in section 3.3.1. The analysis reveals that while existing standards provide strong foundations for certain verification tasks, significant gaps remain that prevent full implementation of the verification framework.

Regarding the first verification task of producing certified verification code, the standards landscape offers partial support. The 17 attestation protocol standards, with 94% high HEV relevance, provide mechanisms for cryptographically signing and verifying code hashes. Standards from TCG, particularly those governing TPM-based measurement and sealing operations, enable the binding of code identity to hardware-rooted keys. However, the standards do not specify protocols for the three certification mechanisms identified in section 3.3.1: manual third-party verification, verified internal code review, and confidential code review by trusted software. These mechanisms require procedural and organizational standards that fall outside the scope of the predominantly technical specifications surveyed. The 26 process and management standards identified could potentially address this gap, but only 54% carry high HEV relevance, suggesting that existing process standards were not designed with AI verification workflows in mind.

For the second verification task of producing verified records of server activity, the standards provide stronger but still incomplete support. The directly attested record generation mechanism requires TEE-capable hardware that can attest to logging software integrity. The 34 confidential computing standards (40% of total, 53% high HEV relevance) and 28 attestation standards (33% of total, 75% high HEV relevance) collectively address the core cryptographic and isolation requirements. Intel TDX and AMD SEV specifications enable VM-level attestation that could encompass logging workloads. However, no surveyed standard specifically addresses the generation of tamper-evident activity logs within TEEs, nor do existing standards define formats or protocols for cryptographic commitment schemes such as Merkle trees applied to server activity records. The subsampled record verification mechanism faces similar gaps: while TEE architecture standards (8.3% of total, 100% high HEV relevance) define isolation properties for verification servers, no standard specifies sampling protocols, re-execution requirements, or statistical confidence thresholds for accepting batch verification results.

The third verification task, producing verified disclosures, receives the weakest standards support. This task requires certified analysis code running inside TEEs to compute aggregate statistics without revealing underlying records. While TEE standards establish the isolation necessary for confidential computation, no surveyed standard addresses the specific challenge of generating privacy-preserving disclosures from verified records. The six AI security framework standards (7.1% of total, 67% high HEV relevance) focus primarily on risk management and governance rather than technical protocols for disclosure generation. Standards for privacy-preserving computation techniques such as differential privacy, secure multi-party computation, and zero-knowledge proofs exist in the broader cryptographic literature but were not represented in the surveyed corpus of hardware-focused specifications. This gap is particularly significant because verified disclosures represent the interface between technical verification infrastructure and human oversight, making them essential for regulatory and governance applications.

5.2.2. Standards Adequacy for TEE Requirements

The four required TEE capabilities identified in section 3.3.2 receive varying levels of standards support. Programmability, the ability for TEEs to run arbitrary user-specified code, is well-supported by the 10 TEE API standards from GlobalPlatform and vendor-specific specifications. These standards define interfaces for loading, executing, and managing trusted applications within enclaves or secure VMs. Attestability receives the strongest standards coverage: the 17 attestation protocol standards (20.2% of total, 94% high HEV relevance) provide comprehensive specifications for generating and verifying cryptographic evidence of TEE identity and state. IETF standards such as RFC 9334 (Remote ATtestation procedureS Architecture) and the Entity Attestation Token specification define interoperable formats and protocols that could support cross-vendor attestation workflows.

Data confidentiality and code/data integrity, the remaining two required capabilities, are addressed by confidential computing standards but with important limitations. Memory encryption standards from Intel (TME, MKTME), AMD (SME, TSME), and ARM (Realm Memory Encryption) specify hardware mechanisms for protecting data in use. However, as noted in the gaps analysis, these standards target CPU architectures and do not extend to GPU memory, creating a fundamental limitation for AI workloads where model weights and activations reside primarily in GPU memory during training and inference. The 7 secure interconnect standards (8.3% of total, 71% high HEV relevance) address data protection during transfer between components, but no standard specifically governs the secure channel between CPU-based TEEs and GPU memory regions that would be necessary for end-to-end confidentiality in AI systems.

The technical requirements for directly attested record generation receive mixed standards support. The requirement for a hardware root of trust anchoring a secure boot chain is well-addressed: the 10 root of trust standards (11.9% of total, 80% high HEV relevance) from TCG and other organizations define TPM-based measurement chains, Core Root of Trust for Measurement implementations, and secure boot protocols. The requirement for low-level software that enforces policies even against root administrators aligns with VM isolation standards from Intel TDX and AMD SEV-SNP, which remove the hypervisor from the trust boundary. However, the requirement for runtime attestation that can measure software integrity without rebooting exposes a gap: while AMD SEV-SNP supports runtime attestation, Intel SGX and many other implementations limit attestation to launch time. Only 28 of 84 standards address the attestation layer, and the specific capability of continuous runtime measurement remains inconsistently specified across vendor implementations.

The technical requirements for subsampled record verification reveal the most significant standards gaps. While TEE-capable verification servers can be provisioned using existing confidential computing standards, no standard addresses the certified sampling and re-execution code that forms the core of this verification mechanism. The requirement for cryptographic commitment using Merkle trees from the prover's servers lacks standardization: while Merkle tree constructions are well-understood cryptographically, no surveyed standard specifies their application to AI workload verification or defines interoperable formats for commitment proofs. Remote attestation standards can bind outputs to certified code, but the integration of attestation with sampling-based verification workflows remains unspecified. In summary, the subsampled verification approach, which offers a more practical path to scalable AI verification than full TEE coverage of all production servers, cannot currently be implemented using standardized components alone.

6. Implications

The findings of this research carry significant implications for policymakers, standards bodies, and AI developers seeking to implement hardware-enabled verification. Three interconnected challenges emerge as particularly consequential: the fragmentation of the standards ecosystem, the necessity of core verification components, and the fundamental gap between CPU-centric security architectures and the demands of AI workloads.

6.1. Fragmentation Across Vendors and Standards

The distribution of standards across 19 different organizations, with no single organization commanding more than 20% of the total, reflects a fundamentally fragmented ecosystem. This fragmentation is not merely an inconvenience; it creates substantive barriers to implementing consistent verification across data centers that employ hardware from multiple vendors. An AI company operating servers with Intel, AMD, and ARM processors must navigate three entirely different TEE architectures, each with distinct security properties, attestation mechanisms, and toolchains. GlobalPlatform's dominance in TEE interface specifications and TCG's leadership in root of trust standards represent partial consolidation, but these bodies have not yet produced unified frameworks that bridge vendor-specific implementations.

The implications for policy are significant. Regulatory frameworks that mandate hardware-enabled verification must either specify particular vendor implementations, which risks creating market distortions and lock-in, or accept heterogeneous compliance pathways that may offer varying levels of actual security assurance. The 88% high HEV relevance of TCG standards suggests that Trusted Platform Module specifications could serve as a common foundation, but extending TPM-based attestation to encompass GPU workloads and TEE-isolated AI computations requires standards work that has not yet been completed.

6.2. Necessary Components for Minimal Standards

The analysis reveals that effective hardware-enabled verification requires three categories of components working in concert: remote attestation infrastructure, secure input and output mechanisms, and audit-capable logging systems. The 94% high HEV relevance of attestation protocol standards confirms that the verification community correctly identifies attestation as the linchpin capability. However, attestation alone is insufficient. Without secure I/O channels, an attacker could intercept or modify data before it enters a TEE or after it exits, rendering the attestation of the TEE's internal state irrelevant. Without tamper-evident audit logs, retrospective verification becomes impossible, and the system cannot support the subsampled record verification mechanism that enables scalable oversight.

The implication for standards development is that point solutions addressing individual components will not yield verifiable AI systems. Future standardization efforts should adopt an architectural perspective that specifies how attestation, secure I/O, and audit logging integrate into coherent verification workflows. The IETF standards, which show 63% high HEV relevance and focus heavily on networking and protocol specifications, represent a natural venue for this integration work given that verification inherently involves communication between provers, verifiers, and relying parties across network boundaries.

6.3. The CPU-Mobile Versus GPU Gap

Perhaps the most consequential finding is the profound mismatch between where security standards have matured and where AI computation actually occurs. The confidential computing ecosystem evolved to protect enterprise workloads running on CPUs and sensitive applications on mobile devices. Intel SGX, AMD SEV, and ARM TrustZone all reflect this heritage. Modern AI training and inference, however, concentrate computational intensity on GPUs and purpose-built accelerators that largely lack equivalent security features. NVIDIA's Confidential Computing initiative represents an emerging response to this gap, but GPU-specific TEE standards remain nascent, and the unique architectural characteristics of GPUs, including massive parallelism, high-bandwidth memory, and complex scheduling, create verification challenges that CPU-centric frameworks do not address.

The practical implication is that hardware-enabled verification for AI cannot simply extend existing confidential computing standards to new hardware. Fundamental technical work is required to develop GPU-compatible attestation mechanisms, memory encryption schemes that do not impose prohibitive performance penalties on tensor operations, and secure channels between CPU-based TEEs and GPU-based secure memory regions. Until this groundwork is completed, hardware-enabled verification for AI will remain limited to the CPU components of AI systems, leaving the most computationally intensive and arguably most security-critical operations, namely the actual training and inference computations, outside the verification perimeter.

This gap also carries implications for the timeline of AI governance implementation. Policymakers contemplating verification mandates must recognize that the technical infrastructure to verify GPU-based AI workloads does not yet exist in standardized, deployable form. A phased approach that initially focuses on verifying the CPU-based orchestration, logging, and attestation infrastructure while GPU verification capabilities mature may represent a pragmatic path forward. Such an approach would establish the policy and operational frameworks for verification while acknowledging the current technical limitations, creating demand signals that could accelerate standards development for GPU-specific security features.

7. Appendix

7.1. TEE Vendor Implementations

7.1.1. Intel

7.1.1.1. Software Guard Extensions (SGX)

Introduced in 2015, SGX creates encrypted memory regions called "enclaves" within user applications. It is the most mature and widely deployed TEE technology. [4]

It has two variants:

- Client SGX: For consumer processors; 128 MB secure memory; full integrity protection
- Scalable SGX (2021): For servers; up to 512 GB secure memory; lacks hardware integrity protection against physical attacks

It can use any of these attestation mechanisms:

- Enhanced Privacy ID (EPID): provides group-based anonymous attestation through Intel's Attestation Service. Intel controls this key attestation infrastructure.
- Data Center Attestation Primitives (DCAP): enables organizations to run their own attestation infrastructure (third-party attestation). DCAP provides an alternative for organizations requiring independence from Intel's services

Evidence is signed using device-specific keys bound to firmware version.

Its strengths are that it’s a very mature ecosystem with extensive tooling, it has fine-grained isolation at application level and well-documented attestation protocols. It also has formally verified attestation mechanisms. Its main limitations are that it requires application redesign ("partitioning" into trusted/untrusted components), it allows no direct system calls or hardware access from within enclaves, and there are memory constraints in client version. It also has closed-source components (quoting enclave, microcode).

7.1.1.2. Trust Domain Extensions (TDX)

Intel Trust Domain Extensions (TDX), unveiled in 2020, is Intel's next-generation TEE technology that introduces hardware-isolated virtual machines called trust domains.

Unlike Intel SGX, which requires partitioning applications into trusted and untrusted components, TDX isolates entire virtual machines. This allows legacy applications running on standard operating systems to be protected without modification. TDX leverages Intel Virtual Machine Extensions (VMX) and Intel Multi-Key Total Memory Encryption (MKTME).

TDX extends SGX's remote attestation mechanisms to issue claims and evidence for trust domains, enabling relying parties to verify the integrity and authenticity of code executing within a trust domain. The attestation architecture has been formally verified by researchers (Sardar et al., 2021).

TDX reflects a broader convergence toward VM-based TEE isolation, a paradigm pioneered by AMD SEV. This approach reduces developer friction since the OS and APIs inside the TEE remain standard, and may facilitate hardware-agnostic solutions like the Open Enclave SDK and WebAssembly-based portable execution environments.

7.1.2. AMD Secure Encrypted Virtualization (SEV)

Protects entire virtual machines (VM) from untrusted hypervisors, making it well-suited for cloud computing scenarios.

AMD SEV has evolved through three generations, each building upon its predecessor. The original SEV, sometimes called "vanilla" SEV, introduced VM memory encryption using per-VM keys. However, this first generation was limited to supporting only 16 VMs and restricted attestation to occur only at launch time.

SEV-ES (Encrypted State) extended the original design by adding encrypted register state protection, which prevents the hypervisor from inspecting CPU register contents during context switches. Despite this improvement, SEV-ES was later proven vulnerable to rollback attacks, where an attacker could restore a previous VM state to bypass security measures.

The most recent generation, SEV-SNP (Secure Nested Paging), adds integrity protection and flexible attestation capabilities. This version addresses the vulnerabilities found in earlier generations and allows runtime attestation, meaning the VM state can be verified at any point during operation rather than only at launch.

SEV's attestation mechanism relies on a "chip endorsement key" that is fused into the processor during manufacturing. An AMD secure processor, which consists of an embedded ARM Cortex-v5, generates the cryptographic materials needed for attestation. For SEV and SEV-ES, attestation occurs during VM launch, while SEV-SNP allows attestation at any time during operation. This process enables the guest owner to verify the VM state before provisioning any secrets.[4]

One of SEV's primary advantages is that it requires no application modification, as it protects entire operating systems at the VM level. This characteristic makes developer adoption considerably easier compared to enclave-based approaches. Additionally, SEV provides protection against malicious cloud providers by ensuring that even the hypervisor cannot access the encrypted VM memory. [4]

SEV does come with notable limitations. The trusted computing base is larger than enclave-based solutions since it encompasses the entire VM rather than a small enclave. Earlier versions, particularly SEV-ES, have known security vulnerabilities that have been documented in academic research. Furthermore, the guest operating system must explicitly support SEV for the protection to function. [4]

7.1.3. ARM TrustZone

There are two architectures.

7.1.3.1. TrustZone-A (Application Processors)

TrustZone-A splits the processor into "secure world" and "normal world" states and is widely deployed in mobile devices and embedded systems. Unlike Intel SGX, which supports multiple enclaves, TrustZone-A provides only a single TEE per system. A Secure Monitor controls transitions between the two worlds, and each world maintains its own user and kernel spaces. The secure world typically runs a trusted operating system such as OP-TEE. [4]

Notably, TrustZone-A has no built-in attestation mechanism. The academic community has developed protocols that require a hardware root of trust, a secure randomness source, and a secure boot mechanism. While mutual attestation protocols do exist, they remain unstandardized. [4]

TrustZone-A offers several strengths, including direct hardware access from the secure world, broad commercial deployment across the mobile ecosystem, and open-source trusted OS options like OP-TEE. However, it also faces limitations: trusted applications are constrained to only a few megabytes of memory, system installation requirements are complex, standardized attestation is lacking, and secure monitor implementations are often proprietary. [4]

7.1.3.2. TrustZone-M (Microcontrollers)

TrustZone-M brings trusted execution to resource-constrained IoT devices built on the Cortex-M series. It differs from TrustZone-A in several important ways: it uses hardware-based world switching, which is faster than a software secure monitor, employs a Memory Protection Unit instead of a Memory Management Unit, and is specifically designed for real-time embedded systems. [4]

The attestation situation for TrustZone-M is more developed. It supports foundational requirements including secure storage, secure boot, and secure inter-world communication. ARM's Platform Security Architecture provides an initial attestation framework, and research protocols such as DIAT enable runtime data integrity attestation. [4]

ARM's dominance in mobile/IoT means TrustZone deployment is vast, but lack of standardized attestation creates interoperability challenges. [4]

7.1.4. RISC-V TEE Solutions

RISC-V is an open instruction set architecture that supports multiple TEE implementations, all leveraging Physical Memory Protection (PMP) instructions as a foundation for isolation.

7.1.4.1. Keystone

Keystone takes a modular framework approach, allowing users to select their own security primitives such as encryption and cache partitioning. Its architecture places a secure monitor in machine mode, runs the runtime in supervisor mode, and executes applications in user mode. For attestation, Keystone uses evidence signed with provisioned keys, with measurements covering the secure monitor, runtime, and application. Keystone currently exists as an open-source research prototype.

7.1.4.2. Sanctum

Sanctum aims to provide SGX-like enclaves through an open-source implementation. It replaces Intel's proprietary components with an open "measurement root" and secure monitor. The attestation mechanism uses a signing enclave similar to SGX and supports key derivation from hardware secrets or Physical Unclonable Functions (PUFs). Sanctum holds the distinction of being the first RISC-V proposal with attestation that has been formally verified.

7.1.4.3. TIMBER-V

TIMBER-V employs hardware-assisted memory tagging designed for small, embedded processors. Its architecture combines tagged memory with a Memory Protection Unit, while a trust manager called TagRoot handles isolation. Attestation currently relies on symmetric cryptography using Message Authentication Codes, though the developers have acknowledged the need for public-key extensions. TIMBER-V remains a research prototype targeting constrained devices.

7.1.4.4. LIRA-V

LIRA-V focuses on mutual remote attestation for constrained edge devices. It features a minimal architecture that operates in machine mode only, with code measurement computed by a ROM-stored program. The attestation protocol is a three-round mutual protocol that has been formally verified. LIRA-V is currently a research prototype and does not support arbitrary TEE code execution.

7.2. Complete AI Verification Standards List

DMTF	DSP0274	Specification	Security Protocol and Data Model (SPDM) Specification	L4: Attestation	Secure Messaging, Attestation Protocols	Interface, Foundational	Accelerator-agnostic	High	Core VM isolation tech; attestation model relevant to HEV
DMTF	DSP0277	Specification	Secured Messages using SPDM Specification	L4: Attestation	Secure Messaging	Interface	Accelerator-agnostic	High	Supply chain transparency; firmware provenance for HEV
DMTF	DSP0290	Standard	MCTP UCIe™ Transport Binding Specification	L3: Confidential Computing	Secure Interconnect	Interface	GPU-applicable	Medium	Firmware update manifest; HEV firmware lifecycle
DMTF	DSP0275	Specification	Security Protocol and Data Model (SPDM) over MCTP Binding Specification	L4: Attestation	Secure Messaging	Interface	Accelerator-agnostic	Medium	Multi-domain updates; complex HEV deployments
DMTF	DSP0276	Specification	Secured Messages using SPDM over MCTP Binding Specification	L4: Attestation	Secure Messaging	Interface	Accelerator-agnostic	Medium	Software identification; firmware inventory for HEV
DMTF	DSP0291	Standard	PCIe® Management Interface (PCIe-MI®) over MCTP Binding Specification	L4: Attestation	Secure Messaging	Interface	GPU-applicable	Medium	SBOM format; indirect firmware transparency
Ecma	ECMA-424	Specification	CycloneDX Bill of materials specification	L1: Platform Integrity	Supply Chain/Provenance	Foundational, Measurement	Accelerator-agnostic	Low	Realm-based isolation; CCA attestation directly applicable
ETSI	GR SAI 002	Report	Securing Artificial Intelligence (SAI); Data Supply Chain Security	L1: Platform Integrity	Supply Chain/Provenance, AI Security Fram	Process, Foundational	AI-specific	Medium	Common Criteria; HEV certification framework
ETSI	GR SAI 004	Report	Securing Artificial Intelligence (SAI); Problem Statement	L1: Platform Integrity	AI Security Framework	Foundational	AI-specific	Medium	CC methodology; HEV evaluation process
ETSI	GR SAI 006	Report	Securing Artificial Intelligence (SAI); The role of hardware in security of AI	L1: Platform Integrity, L3: Confidential Com	AI Security Framework	Foundational	AI-specific	High	Memory expansion for accelerators; TEE-SI critical for GPU CC
ETSI	GR SAI 009	Report	Securing Artificial Intelligence (SAI); Artificial Intelligence Computing Platform Security F	L1: Platform Integrity, L3: Confidential Com	AI Security Framework	Foundational, Process	AI-specific	High	Crypto testing; HEV validation procedures
GlobalPlatform	GP_REQ_025	Specification	Root of Trust Definitions and Requirements v1.1.1	L1: Platform Integrity	Root of Trust	Foundational	Accelerator-agnostic	High	Device authentication foundation; GPU attestation uses SPDM
GlobalPlatform	GPD_SPE_009	Specification	TEE System Architecture v1.3	L3: Confidential Computing	TEE Architecture	Foundational	CPU-focused	High	Encrypted attestation messages; core HEV protocol
GlobalPlatform	GPD_SPE_120	Specification	TEE Management Framework (TMF)	L3: Confidential Computing	TEE Management	Process	CPU-focused	High	Directly addresses hardware security for AI workloads
GlobalPlatform	GPD_SPE_123	Specification	TEE Management Framework: Open Trust Protocol (OTrP) Profile v1.1	L3: Confidential Computing	TEE Management	Process, Interface	CPU-focused	High	AI platform security framework; highly relevant to HEV
GlobalPlatform	GPP_WPR_013	Specification	Software Countermeasures v1.0	L2: Runtime Protection	Runtime Protection	Process	Accelerator-agnostic	Medium	Supply chain levels; firmware build verification
GlobalPlatform	GPD_EPR_017	Specification	TEE Internal API Specification v1.0 Errata and Precisions v3.0	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	RoT requirements applicable to accelerator HEV
GlobalPlatform	GPD_EPR_028	Specification	TEE Client API Specification v1.0 Errata and Precisions v2.0	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	TPM profile; platform attestation baseline
GlobalPlatform	GPD_SPE_007	Specification	TEE Client API Specification v1.0	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	TEE architecture patterns applicable to accelerator TEEs
GlobalPlatform	GPD_SPE_010	Specification	TEE Internal Core API Specification	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	TEE lifecycle management; directly applicable to HEV ops
GlobalPlatform	GPD_SPE_020	Specification	Trusted User Interface API v1.0	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Low	Trust protocol for TEE management; HEV provisioning model
GlobalPlatform	GPD_SPE_021	Specification	TEE TUI Low-Level API	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Low	TEE-based ML framework; directly addresses HEV use case
GlobalPlatform	GPD_SPE_024	Specification	TEE Secure Element API Specification	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	Reference values format; HEV firmware verification
GlobalPlatform	GPD_SPE_025	Specification	TEE TA Debug Specification v1.0.1	L3: Confidential Computing	TEE APIs	Interface, Testing	CPU-focused	Low	TEE provisioning protocol; directly applicable to HEV
GlobalPlatform	GPD_SPE_100	Specification	TEE Sockets API Specification v1.0.1	L3: Confidential Computing	TEE APIs	Interface	CPU-focused	Medium	TEE provisioning architecture; HEV deployment model
GlobalPlatform	GPP_TEN_012	Specification	TEE API Call Validation v1.0	L3: Confidential Computing	TEE APIs	Testing	CPU-focused	Low	Attestation result format; core HEV verification output
GlobalPlatform	GPT_GUI_147	Specification	Guidance on the Use of TEE PP and PP-Modules	L3: Confidential Computing	Certification	Process, Testing	Accelerator-agnostic	Medium	Software countermeasures; defense in depth for HEV
GlobalPlatform	GPT_SPE_142	Specification	Trusted User Interface PP-Module v1.0	L3: Confidential Computing	Certification	Testing	CPU-focused	Low	Trust anchor management; HEV trust root distribution
IEEE	2830-2021	Standard	IEEE Standard for Technical Framework and Requirements of Trusted Execution Environ	L3: Confidential Computing, L4: Attestation	AI Security Framework, TEE Architecture	Foundational, Process	AI-specific	High	Endorsement format; manufacturer trust for HEV
IETF	draft-ietf-scitt-architecture-22	Draft Standard	An Architecture for Trustworthy and Transparent Digital Supply Chains	L1: Platform Integrity	Supply Chain/Provenance	Foundational, Process	Accelerator-agnostic	Medium	Message encapsulation; HEV protocol framing
IETF	draft-ietf-suit-manifest	Draft Standard	CBOR-based Serialization Format for the SUIT Manifest	L1: Platform Integrity	Secure Update	Foundational	Accelerator-agnostic	Medium	Chiplet security transport; relevant for multi-die accelerators
IETF	draft-ietf-suit-trust-domains	Draft Standard	SUIT Manifest Extensions for Multiple Trust Domains	L1: Platform Integrity	Secure Update	Foundational, Process	Accelerator-agnostic	Medium	TEE API standardization; model for GPU TEE APIs
IETF	RFC 9393	Standard	Concise Software Identification Tags	L1: Platform Integrity	Supply Chain/Provenance	Foundational, Measurement	Accelerator-agnostic	Medium	Client-side TEE interface patterns
IETF	draft-ietf-rats-corim	Draft Standard	Concise Reference Integrity Manifest (CoRIM)	L1: Platform Integrity, L4: Attestation	Attestation Protocols	Foundational, Measurement	Accelerator-agnostic	High	TEE client interface; reference for HEV client APIs
IETF	draft-ietf-teep-protocol	Draft Standard	Trusted Execution Environment Provisioning (TEEP) Protocol	L3: Confidential Computing	TEE Management	Interface, Process	Accelerator-agnostic	High	Attestation flow models; HEV architecture patterns
IETF	RFC 9397	Standard	Trusted Execution Environment Provisioning (TEEP) Architecture	L3: Confidential Computing	TEE Management	Foundational, Process	Accelerator-agnostic	High	Internal TEE operations; reference architecture
IETF	draft-ietf-rats-ar4si-09	Draft Standard	Attestation Results for Secure Interactions	L4: Attestation	Attestation Protocols	Foundational, Measurement	Accelerator-agnostic	High	UI security; limited HEV relevance
IETF	draft-ietf-rats-concise-ta-stores	Draft Standard	Concise Trust Anchor Stores (CoTS)	L4: Attestation	Attestation Protocols	Foundational, Process	Accelerator-agnostic	High	Low-level UI; limited HEV relevance
IETF	draft-ietf-rats-endorsements	Draft Standard	RATS Endorsements	L4: Attestation	Attestation Protocols	Foundational	Accelerator-agnostic	High	SE integration patterns; HSM integration for HEV
IETF	draft-ietf-rats-msg-wrap-23	Draft Standard	RATS Conceptual Messages Wrapper (CMW)	L4: Attestation	Attestation Protocols	Interface	Accelerator-agnostic	High	Debug interfaces; security boundary considerations
IETF	draft-ietf-rats-reference-interaction-models-15	Draft Standard	Reference Interaction Models for Remote Attestation Procedures	L4: Attestation	Attestation Protocols	Foundational, Process	Accelerator-agnostic	High	Network communication from TEE; attestation transport
IETF	RFC 9334	Standard	Remote ATtestation procedureS (RATS) Architecture	L4: Attestation	Attestation Protocols	Foundational	Accelerator-agnostic	High	Core attestation architecture; foundational for HEV
IETF	RFC 9711	Standard	The Entity Attestation Token (EAT)	L4: Attestation	Attestation Protocols	Foundational, Measurement	Accelerator-agnostic	High	Attestation token format; core HEV evidence format
IETF	draft-ietf-teep-otrp-over-http	Draft Standard	HTTP Transport for TEEP: Agent Initiated Communication	L3: Confidential Computing	TEE Management	Interface	CPU-focused	Medium	API validation; testing framework
IETF	draft-ietf-rats-evidence-trans-02	Draft Standard	Evidence Transformations	L4: Attestation	Attestation Protocols	Process	Accelerator-agnostic	Medium	Protection profile guidance; HEV certification path
Intel	853294	Specification	Intel Trust Domain Extensions (Intel TDX) Module Base Architecture Specification	L3: Confidential Computing	TEE Architecture	Interface, Foundational	CPU-focused	High	UI protection profile; limited HEV relevance
ISO/IEC	5962	Standard	Information technology — SPDX Specification V2.2.1	L1: Platform Integrity	Supply Chain/Provenance	Foundational, Measurement	Accelerator-agnostic	Low	TEEP transport; TEE provisioning delivery
ISO/IEC	15408	Standard	Information security, cybersecurity and privacy protection — Evaluation criteria for IT se	L1: Platform Integrity	Certification	Testing, Process	Accelerator-agnostic	Medium	VM isolation architecture; attestation model for HEV
ISO/IEC	18045	Standard	Information security, cybersecurity and privacy protection — Evaluation criteria for IT se	L1: Platform Integrity	Certification	Testing, Process	Accelerator-agnostic	Medium	TPM specification; hardware RoT for HEV
ISO/IEC	24759	Standard	Information security, cybersecurity and privacy protection — Test requirements for cryp	L1: Platform Integrity	Cryptographic Requirements	Testing	Accelerator-agnostic	Medium	Crypto module requirements; HEV crypto validation
ISO/IEC	11889	Standard	Information technology — Trusted platform module library	L1: Platform Integrity	Root of Trust	Foundational, Interface	Accelerator-agnostic	High	Memory module communication; HBM security channel
ISO/IEC	19790	Standard	Information security, cybersecurity and privacy protection — Security requirements for	L1: Platform Integrity	Cryptographic Requirements	Testing, Foundational	Accelerator-agnostic	High	Key management protocol; HEV key provisioning
ISO/IEC	27070	Standard	Information technology — Security techniques — Requirements for establishing virtuali	L1: Platform Integrity	Root of Trust	Foundational	Accelerator-agnostic	High	Virtual RoT requirements; directly applicable to HEV
ISO/IEC	27090	Standard	Cybersecurity — Artificial Intelligence — Guidance for addressing security threats and c	L1: Platform Integrity, L3: Confidential Com	AI Security Framework	Foundational, Process	AI-specific	High	AI security guidance; context for HEV requirements
ISO/IEC	20008	Standard	Information technology — Security techniques — Anonymous digital signatures	L4: Attestation	Cryptographic Requirements	Foundational	Accelerator-agnostic	Medium	Crypto module validation; HEV compliance requirement
ISO/IEC	27071	Standard	Cybersecurity — Security recommendations for establishing trusted connections betwe	L4: Attestation	Secure Messaging	Interface, Process	Accelerator-agnostic	Medium	Firmware resilience; HEV firmware protection
ISO/IEC	27099	Standard	Information technology — Public key infrastructure — Practices and policy framework	L4: Attestation	Cryptographic Requirements	Process, Foundational	Accelerator-agnostic	Medium	Device integrity validation; HEV verification guidance
JEDEC	JESD403-1C.01	Standard	JEDEC Module Sideband Bus (SidebandBus)	L3: Confidential Computing	Secure Interconnect	Interface	GPU-applicable	Medium	Open RoT design; reference implementation for HEV
NIST	FIPS 140-3	Standard	SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES	L1: Platform Integrity	Cryptographic Requirements	Testing, Foundational	Accelerator-agnostic	High	Security assessment framework; HEV evaluation model
NIST	SP 800-193	Standard	Platform Firmware Resiliency (PFR) Guidelines	L1: Platform Integrity	Secure Update, Root of Trust	Process, Foundational	Accelerator-agnostic	High	Transport binding; relevant for device-level HEV
NIST	SP 1800-34	Standard	Validating the Integrity of Computing Devices	L1: Platform Integrity, L4: Attestation	Supply Chain/Provenance, Attestation Prot	Process, Measurement	Accelerator-agnostic	High	Secure channel establishment for attestation
OASIS	KMIP	Standard	Key Management Interoperability Protocol	L3: Confidential Computing	Key Management	Interface, Process	Accelerator-agnostic	Medium	Key management architecture; HEV key hierarchy
OCP	Caliptra 2.0	Specification	Caliptra: A Datacenter System on a Chip (SoC) Root of Trust (RoT)	L1: Platform Integrity	Root of Trust	Foundational, Interface	GPU-applicable	High	PCIe device management; GPU management channel
OCP	SAFE	Framework	Security Appraisal Framework and Enablement	L1: Platform Integrity, L4: Attestation	Certification	Testing, Process	GPU-applicable	High	Component attestation; directly applicable to GPU HEV
OCP	LOCK	Specification	Layered Open-source Cryptographic Key management	L3: Confidential Computing	Key Management	Process, Foundational	Accelerator-agnostic	High	PCIe encryption; GPU memory protection in transit
OCP	Attestation of System Components v1.0	Specification	Attestation of System Components v1.0 Requirements and Recommendations	L4: Attestation	Attestation Protocols	Foundational, Measurement	GPU-applicable	High	Extended TEE device protocol; GPU CC integration
PCI-SIG	PCIe 6.1 Sec 6.33	Specification	Integrity and Data Encryption (IDE)	L3: Confidential Computing	Secure Interconnect	Interface	GPU-applicable	High	Evidence format conversion; interoperability for HEV
PCI-SIG	TDISP eXtended TEE (XT) Extensions	Specification	TDISP eXtended TEE (XT) Extensions	L3: Confidential Computing	TEE Architecture, Secure Interconnect	Interface	GPU-applicable	High	TEE-device interface; core GPU CC protocol
PCI-SIG	TEE Device Interface Security Protocol (TDISP)	Specification	TEE Device Interface Security Protocol (TDISP)	L3: Confidential Computing, L4: Attestation	TEE Architecture, Secure Interconnect	Interface	GPU-applicable	High	PCIe device attestation; GPU identity verification
PCI-SIG	PCIe ECN: CMA	Specification	Component Measurement and Authentication (CMA)	L4: Attestation	Attestation Protocols	Measurement, Interface	GPU-applicable	High	Layered RoT; boot measurement chain for HEV
SLSA	SLSA v1.0	Specification	Supply-chain Levels for Software Artifacts	L1: Platform Integrity	Supply Chain/Provenance	Process, Measurement	Accelerator-agnostic	Medium	TPM specification; hardware RoT implementation
TCG	PC Client TPM Profile	Specification	PC Client Platform TPM Profile Specification	L1: Platform Integrity	Root of Trust	Interface, Foundational	CPU-focused	Medium	Anonymous attestation; privacy-preserving HEV
TCG	DICE Layering Architecture	Specification	DICE Layering Architecture	L1: Platform Integrity	Root of Trust	Foundational	Accelerator-agnostic	High	Trusted connections; HEV communication security
TCG	TPM 2.0 Library	Specification	Trusted Platform Module 2.0 Library	L1: Platform Integrity	Root of Trust	Foundational, Interface	Accelerator-agnostic	High	PKI framework; certificate management for HEV
TCG	DICE Attestation Architecture	Specification	DICE Attestation Architecture	L1: Platform Integrity, L4: Attestation	Attestation Protocols, Root of Trust	Foundational	Accelerator-agnostic	High	Layered attestation; foundational for HEV evidence
TCG	DICE Certificate Profiles	Specification	DICE Certificate Profiles	L4: Attestation	Device Identity	Foundational, Measurement	Accelerator-agnostic	High	Device identity certs; HEV identity binding
TCG	DICE Endorsement Architecture for Devices	Specification	DICE Endorsement Architecture for Devices	L4: Attestation	Attestation Protocols	Foundational, Process	Accelerator-agnostic	High	Endorsement architecture; manufacturer trust for HEV
TCG	TCG DICE Concise Evidence	Specification	TCG DICE Concise Evidence Binding for SPDM	L4: Attestation	Attestation Protocols	Interface, Measurement	Accelerator-agnostic	High	DICE-SPDM binding; device attestation integration
TCG	TCG Platform Requirements for Certificates and	Specification	TCG Platform Requirements for Certificates and RIMs	L4: Attestation	Attestation Protocols	Foundational, Measurement	Accelerator-agnostic	High	Platform attestation requirements; HEV cert profiles
UCIe	UCIe 2.0 Sec 7	Specification	Universal Chiplet Interconnect Express Security	L3: Confidential Computing	Secure Interconnect	Interface	GPU-applicable	High	Chiplet security; multi-die accelerator protection

8. Bibliography

[1] M. Baker, G. Kulp, O. Marks, M. Brundage, and L. Heim, “Verifying International Agreements on AI: Six Layers of Verification for Rules on Large-Scale AI Development and Deployment,” Jul. 2025.

[2] M. Bartock *et al.*, “Hardware-enabled security :,” May 2022. doi: 10.6028/NIST.IR.8320.

[3] O. Aarne, T. Fist, and C. Withers, “Secure, Governable Chips Using On-Chip Mechanisms to Manage National Security Risks from AI & Advanced Computing About the Technology and National Security Program About the Artificial Intelligence Safety & Stability Project,” 2024.

[4] J. Menetrey *et al.*, *Attestation Mechanisms for Trusted Execution Environments Demystified*, vol. 13272. in Lecture Notes in Computer Science, vol. 13272. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-031-16092-9.

[5] Big Data UN Global Working Group, “UN Handbook on Privacy-Preserving Computation Techniques.” [Online]. Available: https://marketplace.officialstatistics.org/technologies-and-techniques

[6] A. Reuel *et al.*, “Open Problems in Technical AI Governance,” Apr. 2025, [Online]. Available: http://arxiv.org/abs/2407.14981

[7] AI Standards Hub, “Standards at a glance,” The Alan Turing Institute. Accessed: Jan. 10, 2026. [Online]. Available: https://aistandardshub.org/resource/main-training-page-example/2-different-types-of-standards/