ZIRU LABS · THREAT TAXONOMY

The Physics-Layer Threat Taxonomy for AI Infrastructure

A Reference Framework

Ziru Labs Research Publication · Version 1.0 · 2026 · Published under Ziru Labs corporate byline with contribution from Daniel Martin ·

Citable reference: Ziru Labs. *The Physics-Layer Threat Taxonomy for AI Infrastructure: A Reference Framework*, v1.0. Published at zirulabs.com/research.

ABSTRACT

> This framework establishes a reference taxonomy for the physics-layer threat classes that confront AI infrastructure across its deployment lifecycle. Physics-layer threats are those that operate at or below the boundary of the software trust model. They exploit physical, electrical, and structural properties of the underlying hardware in ways that operating system controls, virtualization, and application-layer defenses cannot adequately mitigate alone.
>
> As AI systems move into regulated enterprise, defense, and sovereign deployment environments, the physics layer has become the definitional boundary of what AI trust actually means in practice.
>
> The taxonomy organizes threats across five attack surfaces (physical access, bus and interconnect, firmware and boot, inference integrity, AI governance persistence), enumerates representative threat classes within each surface against canonical literature references, and establishes explicit criteria for when a defensive mechanism can be said to address a given threat class. The framework is intended as a living reference: the planned quarterly State of Physics-Layer AI Trust report tracks evolution of the threat landscape against this taxonomy.
>
> Ziru Labs authored this taxonomy as one operator within the trust layer for AI category. The taxonomy is offered as a reference for the security community rather than as a description of any single product: it organizes the physics-layer threat landscape in a way intended to remain useful to practitioners, standards bodies, and other operators regardless of which defensive technologies they deploy. Where Ziru Labs technology is referenced, it is referenced as one operational response to threats in the taxonomy.

1. Why the Physics Layer

AI trust, as the term is used in 2026 regulatory and policy contexts, does not mean what AI alignment researchers, model-safety researchers, or machine learning security researchers typically mean by the same words.

Regulatory AI trust is closer in meaning to what the information security discipline would call cryptographic assurance of computation properties: the ability to make verifiable assertions, backed by something harder than software, about what an AI system actually did with what data under what constraints. Executive Order 14179 implementation activities, FY2026 NDAA Section 1513 (Physical and Cybersecurity Procurement Requirements for Artificial Intelligence Systems; P.L. 119-60) framework development at the Department of Defense, the EU AI

Act conformity assessment regime under Articles 40 and 43, NATO's emerging STANAG framework for AI trust, and the AI security work under ISO/IEC 27090 and ISO/IEC JTC 1/SC 42 converge on variations of this requirement. This activity proceeds within a broader federal AI policy environment that includes Executive Order 14365 (Ensuring a National Policy Framework for Artificial Intelligence, signed 11 December 2025), which addresses national AI regulatory policy and federal preemption of conflicting state law rather than hardware-rooted assurance specifically.

The substrate capable of supporting verifiable assertions of this kind does not exist at the operating system layer, because the operating system itself is part of what needs to be attested. It does not exist at the hypervisor or container layer, for the same reason. It does not exist at the trusted execution boundary of conventional confidential computing, because that boundary terminates at the edge of the compute accelerator die. Confidential computing's in-die memory encryption and encrypted bus traffic, including encrypted VRAM and encrypted PCIe on supported platforms, mitigate specific confidentiality vectors within an active session; but the boundary still terminates at the die and the session. The threat classes that operate outside it, including physical extraction of material beyond the encrypted envelope or outside confidential-computing mode, bus-level traversal under operating system compromise, chassis-level tamper, and lateral fabric traversal, remain outside what confidential computing addresses as a class, as does runtime evidence of the operation itself.

The substrate exists at the physics layer: the set of hardware mechanisms that operate at electrical, physical, and structural level, rooted in silicon and chassis rather than in software abstractions. A coherent picture of what threats AI infrastructure actually faces at this layer is a prerequisite for any serious technical, regulatory, or procurement conversation about what AI trust means. The physics-layer threat landscape has, as of early 2026, no canonical reference taxonomy. This framework is an attempt at one.

1.1 Relationship to Adjacent Frameworks

The Physics-Layer Threat Taxonomy composes with several adjacent security and AI governance frameworks already in operational use across the security community. The taxonomy is designed to interoperate with these frameworks rather than substitute for them. Specific composition relationships:

The MITRE ATT&CK Enterprise Framework (MITRE Corporation, 2024) provides the canonical taxonomy of adversarial tactics and techniques in conventional information security. The Physics-Layer Threat Taxonomy maps directly to several ATT&CK technique identifiers, including T1542 Pre-OS Boot, T1542.001 System Firmware, T1195 Supply Chain Compromise, T1200 Hardware Additions, T1014 Rootkit, and the persistence dimensions of T1606.002 Forge Web Credentials. A practitioner familiar with ATT&CK can use the Physics-Layer Threat Taxonomy as the AI-infrastructure-specific extension that ATT&CK's enterprise scope does not currently address.

The NIST Cybersecurity Framework 2.0 (NIST CSWP 29, 2024) provides the canonical functional framework for cybersecurity programs in U.S. organizations. The Physics-Layer Threat Taxonomy's threat classes populate the Identify function (ID.AM, ID.RA) for AI infrastructure programs and signal where Protect (PR.DS, PR.AC) and Detect (DE.CM, DE.AE) obligations follow.

The NIST AI Risk Management Framework (NIST AI 100-1, 2023) and its accompanying Playbook (NIST, 2024) provide the canonical AI risk framework. The Physics-Layer Threat Taxonomy supplies the hardware-layer threat

catalog that supports the AI RMF's GOVERN and MAP functions for AI infrastructure programs.

The OWASP Top 10 for Large Language Model Applications v2.0 (OWASP Foundation, 2025) catalogs application-layer LLM risks. The Physics-Layer Threat Taxonomy addresses the hardware-layer concerns that OWASP's application-layer scope does not currently address.

ISO/IEC 42001:2023 (AI management systems) and ISO/IEC 27001:2022 (information security management systems) provide management-system frameworks that the Physics-Layer Threat Taxonomy's threat classes populate at AI infrastructure scope. ISO/IEC 15408:2022 (Common Criteria evaluation framework) is the canonical evaluation framework for sovereign and regulated-industry deployments where physics-layer concerns are evaluated.

FIPS 140-3 (NIST, 2019) and NIST SP 800-53 Rev. 5 (NIST, 2020) provide the cryptographic-module and security-control frameworks under which AI infrastructure components are evaluated for U.S. federal deployment. The Physics-Layer Threat Taxonomy is designed to compose with the security-control structure that NIST SP 800-53 establishes.

The composition pattern across all of these frameworks is consistent: the Physics-Layer Threat Taxonomy supplies the AI-infrastructure-specific hardware-layer threat catalog that the adjacent framework's scope does not currently address. The taxonomy strengthens rather than displaces the existing reference architecture.

2. Scope and Method

The taxonomy organizes threats by attack surface. An attack surface is the physical, logical, or architectural boundary across which an adversary operates to obtain, modify, or observe information or behavior that should be protected. Five attack surfaces are identified below. Within each surface, specific threat classes are enumerated.

For each threat class, the taxonomy provides:

- A brief description of what the adversary does and what they obtain
- The AI-infrastructure-specific framing of the threat class
- Primary and supporting references from the academic, industry, government, and standards-body literature where the threat class is documented
- Assessment of detection difficulty at the current state of defensive practice
- Assessment of whether software-layer defenses can adequately address the threat class
- Identification of the physics-layer defense classes that can, in principle, address the threat class

The taxonomy is deliberately conservative on novelty claims. Most of the threat classes documented here are known to the academic security community, to cleared defense and intelligence communities, and to the security architects at frontier AI labs. What is novel about this document is the organization rather than the enumeration. Practitioners working on AI infrastructure security have needed a shared vocabulary for some time; this document offers one.

Where threat classes are plausible but have limited public documentation, the taxonomy marks them as such and cites whatever primary references exist. Epistemic discipline applies throughout: what is settled is distinguished

from what is inferred, which is distinguished from what is speculative.

Conventional kernel and hypervisor security is well-covered in the established information security literature and standards corpus. The Physics-Layer Threat Taxonomy assumes kernel and hypervisor security as a prerequisite layer in defense-in-depth architectures and addresses the physics-layer threats that persist under the assumption that kernel and hypervisor security may be partially or fully compromised.

3. Attack Surface 1: Physical Access

Physical-access threats are those where the adversary has some degree of physical access to the hardware: to the chassis, to the board, to specific components, or to the electrical environment of the system. Physical access is frequently assumed to be an edge case in cloud deployment environments, but it is the baseline assumption in classified, sovereign, mission-critical, and edge deployments, and it is an increasingly realistic threat vector in supply chain compromise scenarios even for commercial deployments. The physical-access surface therefore warrants serious attention across the AI infrastructure landscape.

For AI infrastructure specifically, physical-access threats matter because the assets at the physical layer (model weights in DRAM and HBM, inference intermediate state, cryptographic keys protecting model and data, attestation roots in silicon) are the most concentrated value targets in the deployment, and their exposure to physical-access threat vectors is the surface that confidential computing architectures do not extend to.

3.1 S1.T1 Cold-Boot Memory Extraction

The canonical cold-boot attack exploits the observation that DRAM contents persist for seconds to minutes after power is removed, and that this persistence can be extended to hours with refrigerant application. An adversary with brief physical access to a running or recently-running system can extract cryptographic keys, model weights, inference intermediate state, and other sensitive material from DRAM. Variants developed since the original publication include liquid nitrogen remanence extension, targeted refrigerant application to specific DRAM regions, and BIOS-bypass variants that skip memory zeroization during boot.

For AI infrastructure, cold-boot extraction is particularly consequential because model weights and inference state held in DRAM and HBM represent the highest-value extraction targets in a typical deployment.

Primary reference: Halderman et al. (2008).

Supporting references: Gruhn and Müller (2013); Yitbarek et al. (2017).

Detection difficulty: high when properly executed; the attack leaves minimal evidence.

Software defense adequacy: inadequate by construction; the attack operates against memory contents after the operating system is no longer running.

Confidential-computing interaction: confidential-computing memory encryption (for example, encrypted VRAM on supported accelerators, or CPU-side memory encryption under SEV-SNP and TDX) renders cold-boot extraction of the encrypted region ineffective while a confidential-computing session is active. The threat class remains material for deployments not running in confidential-computing mode, for the key and attestation-root

material the scheme depends on, and for the pre-session, post-session, and decryption windows the encrypted envelope does not cover. Memory encryption also provides confidentiality of stored contents rather than verifiable evidence of the operation performed against them.

Physics-layer defense classes: fast memory zeroization on power-plane disruption; active chassis tamper response with zeroization trigger; in-memory encryption with keys held in tamper-resistant hardware.

3.2 S1.T2 Chip-Level Physical Probing

Direct probing of chip-level signals through die-level probing, package decapsulation, electromagnetic side-channel analysis, and differential power analysis supports extraction of cryptographic material, model weights, and inference state from running or recently-running silicon. The semiconductor failure analysis techniques required are established in the physical security and counterfeit analysis literature and can, in principle, be applied adversarially. The population of adversaries capable of executing these attacks is smaller than for cold-boot but non-zero, and includes nation-state-level adversaries with semiconductor analysis capability.

For AI infrastructure, chip-level probing matters because the model weights and the attestation roots in silicon are the most valuable and the most concentrated targets in any deployment, and probing techniques can extract this material in ways that no software-layer defense can prevent or detect.

Primary reference: Skorobogatov (2005).

Supporting references: Tarnovsky (2010); Quisquater and Samyde (2001); Kocher, Jaffe, and Jun (1999).

Detection difficulty: very high; by the time analysis is conducted, the target system is no longer functional.

Software defense adequacy: none; attack is post-mortem on the targeted component.

Physics-layer defense classes: in-memory encryption with keys held separately in tamper-resistant hardware; zero-trust assumption on any silicon contents extractable outside the operational envelope; active probe-detection mesh and potting for high-threat deployments.

3.3 S1.T3 Chassis-Level Tampering

Physical tamper against the chassis or enclosure of a running AI infrastructure system supports adversarial objectives including insertion of monitoring hardware, substitution of components, extraction of electrical signals, and establishment of persistent physical-layer compromise. Chassis tamper is a baseline assumption in hostile-environment deployments (sovereign edge, forward-deployed defense systems) and is increasingly relevant in supply chain and facility compromise scenarios for commercial deployments.

For AI infrastructure, chassis-level tampering matters in deployments where physical custody of the silicon cannot be continuously assured, which includes most defense, sovereign edge, and regulated commercial deployments.

Primary operational reference: NSA TAO ANT Catalog (2008, publicly disclosed 2013).

Supporting references: Appelbaum, Horchert, and Stöcker (2013); Greenberg (2014); Smith and Weingart (1999).

Detection difficulty: varies with tamper sensor sophistication; detection-to-response time is the defining metric.

Software defense adequacy: none.

Physics-layer defense classes: active tamper sensors with zeroization trigger; chassis integrity measurement with hardware attestation; anti-tamper mesh and potting for high-threat deployments.

3.4 S1.T4 Supply Chain Compromise Post-Manufacture

Compromise of hardware between manufacture and operational deployment, through hardware trojan insertion, counterfeit component substitution, firmware pre-compromise of board-level components (BMC, network controllers, specific peripheral chipsets), or interception-and-modification in transit, represents a category of physical-access threat distinct from in-operation tamper. Supply chain integrity is a defense industrial base concern of long standing and an increasingly material commercial concern as AI infrastructure scales across multinational manufacturing.

For AI infrastructure, supply chain compromise matters because the trust assumptions for any AI deployment rest on the integrity of the silicon and the firmware shipped from manufacture, and physics-layer defenses depend on those trust assumptions being recoverable through attestation rather than asserted.

Primary standards reference: NIST SP 800-161 Rev. 1 (2022).

Supporting references: CISA-NIST (2021); Executive Order 14028 (2021); NIST SP 800-218 (2022); Okhravi and Nicol (2018).

Detection difficulty: very high in the general case; varies with specific supply chain discipline.

Software defense adequacy: none for silent hardware compromise.

Physics-layer defense classes: component-level cryptographic attestation; supply chain integrity measurement at hardware root of trust; component substitution detection through side-channel analysis; trusted foundry programs for high-assurance components.

4. Attack Surface 2: Bus and Interconnect

Bus and interconnect threats target the pathways through which data moves between components within an AI infrastructure system. These pathways include PCIe fabric, CXL, NVLink, Infinity Fabric, and the various vendor-specific interconnects. The threats are distinguished from operating-system-layer threats by the observation that they frequently operate at a layer below the operating system’s ability to observe or control, particularly under conditions where a subset of the operating system is compromised.

For AI infrastructure, bus and interconnect threats matter because the data flowing on these pathways (model weights during load, inference batches, attestation handshake messages, accelerator-to-accelerator gradient communication during distributed training and inference) is high-value data that the trusted execution environment boundary does not protect once it leaves the accelerator die.

4.1 S2.T1 PCIe Bus Scraping Under Operating System Compromise

An adversary with kernel-level compromise of the host operating system, but without direct access to the accelerator's trusted execution environment, can configure PCIe devices to observe or intercept traffic traversing the PCIe fabric between the host and the accelerator. Sensitive material traversing this bus (during model load, during inference batch submission, during attestation handshake) becomes observable. IOMMU configurations mitigate some variants but do not close the attack surface in general.

For AI infrastructure, PCIe scraping matters because the host-to-accelerator pathway carries the model weights at load time and the inference batch data at runtime, and host operating system compromise is the baseline threat model for the deployments where AI trust matters most.

Primary reference: Markettos et al. (2019).

Supporting references: Stewin and Bystrov (2013); NVIDIA Corporation H100 Tensor Core GPU Architecture (current revision); NVIDIA Confidential Computing Deployment Guide (current revision).

Detection difficulty: moderate; software-level anomaly detection can catch some patterns.

Software defense adequacy: partial under uncompromised operating system conditions; degrades to inadequate under the operating system compromise the threat model specifically assumes.

Confidential-computing interaction: confidential-computing bus encryption (for example, encrypted PCIe traffic where the GPU is integrated into the CPU trusted execution environment) renders scraped traffic ciphertext while a confidential-computing session is active. The threat class remains material for deployments not running in confidential-computing mode, for interconnects or transfer phases the session does not encrypt, and as a path to traffic-analysis and fault-injection objectives that payload encryption alone does not foreclose.

Physics-layer defense classes: hardware-enforced bus encryption; physical bus monitoring with hardware anomaly detection; chassis-level PCIe traffic gating; structural fabric connectivity limitation in high-assurance deployments.

4.2 S2.T2 NVLink and High-Bandwidth Fabric Interception

NVLink, AMD Infinity Fabric, and proprietary vendor interconnects provide high-bandwidth connectivity between accelerators. They are generally assumed trusted within the vendor's trust model, but that trust model assumes the interconnects are physically isolated and not adversarially configured. In deployment environments where either assumption may be questioned, the interconnects represent a distinct threat surface.

For AI infrastructure, fabric interception matters in distributed training and large-scale inference deployments where model state crosses these high-bandwidth pathways at scale and where any interception capability translates directly to model weight exposure.

Primary vendor references: NVIDIA Corporation, NVLink and NVSwitch technical documentation (current revision); NVIDIA NVLink-C2C Interconnect Technology documentation.

Supporting note: academic literature on high-bandwidth interconnect security is limited; the taxonomy marks this as an area where additional academic work is needed.

Detection difficulty: high; interconnect-level instrumentation is limited.

Software defense adequacy: none at the interconnect layer itself.

Physics-layer defense classes: hardware-enforced interconnect isolation; cryptographic attestation of interconnect peer identity; structural fabric connectivity limitation in high-assurance deployments.

4.3 S2.T3 CXL and Memory-Pool Fabric Interception

CXL (Compute Express Link) fabric enables pooled memory and accelerator disaggregation across rack-scale deployments. The fabric's architectural value (cross-node memory sharing, accelerator aggregation) creates lateral traversal pathways that a compromised node can potentially exploit. This threat class is emerging as CXL deployment scales in 2026 and 2027.

For AI infrastructure, CXL interception matters because the fabric carries memory pooled across nodes, and the lateral traversal pathways enable cross-tenant and cross-deployment exposure in ways that have not historically existed in non-pooled accelerator architectures.

Primary specification reference: CXL Consortium (2022), Compute Express Link Specification Revision 3.0.

Supporting reference: CXL IDE (Integrity and Data Encryption) Specification within the CXL 3.0 specification family.

Detection difficulty: emerging; detection patterns are not yet well-established.

Software defense adequacy: limited; the fabric operates below the operating system layer.

Physics-layer defense classes: hardware-enforced fabric isolation; zero-trust fabric architecture with per-transaction attestation; structural fabric connectivity limitation in high-assurance deployments.

4.4 S2.T4 Lateral Memory Traversal Across Tenant Boundaries

In shared infrastructure where multiple AI workloads run on the same physical hardware (multi-tenant cloud accelerator deployments, MIG-partitioned GPU sharing, CXL-pooled deployments), lateral memory traversal across the boundary between tenants represents a distinct threat class. Pathways include shared cache exploitation, side-channel observation across MIG instances, and side-channel observation across pooled-memory access patterns.

For AI infrastructure, lateral tenant traversal matters because the multi-tenant deployment pattern is the default for cloud-hosted AI services, and the boundary between tenants is the value boundary for any model weight extraction or inference confidentiality concern.

Primary reference: Ristenpart, Tromer, Shacham, and Savage (2009).

Supporting references: Naghibijouybari et al. (2018); Jiang, Fei, and Kaeli (2017); NVIDIA Multi-Instance GPU User Guide (current revision).

Detection difficulty: varies.

Software defense adequacy: mature at the application layer; bounded at side-channel layers where physical defenses are required.

Physics-layer defense classes: hardware-enforced tenant isolation independent of hypervisor integrity; physical memory partitioning; hardware-rooted multi-tenant attestation; structural elimination of side-channel pathways at the silicon layer.

5. Attack Surface 3: Firmware and Boot

Firmware-and-boot threats exploit the observation that modern AI infrastructure systems contain many firmware layers (BIOS, UEFI, BMC, NIC firmware, accelerator firmware, GPU System Processor firmware, specific component firmware) with varying security postures, and that the supply chain for these firmware layers extends well beyond the nominal hardware vendor.

For AI infrastructure, firmware threats matter because firmware compromise persists across operating system reinstallation and across attestation re-establishment, making it a load-bearing concern for any deployment that requires durable trust posture.

5.1 S3.T1 GPU Firmware Injection or Modification

AI accelerators run their own firmware managing scheduling, memory, and operational behavior of the accelerator. Compromise of accelerator firmware provides privileged access that can potentially observe model weights, inference state, and attestation output. Vendor mitigation is improving but the threat class is established, and the relative opacity of accelerator firmware architectures from the vendor side complicates third-party validation.

For AI infrastructure, GPU firmware compromise matters because the firmware operates at the layer where the AI workload actually runs, and any compromise at this layer is structurally indistinguishable from authorized behavior.

Primary references: Kallenberg and Kovah (2015).

Supporting vendor reference: NVIDIA Corporation, GSP Firmware Architecture documentation.

Detection difficulty: very high; accelerator firmware is opaque by design.

Software defense adequacy: none at the firmware layer itself.

Physics-layer defense classes: hardware-enforced accelerator firmware attestation; measured boot of accelerator firmware to hardware root of trust; defense in depth through independent hardware monitoring.

5.2 S3.T2 Driver-Level Privilege Escalation Affecting AI Inference Integrity

GPU drivers run with elevated privilege and represent a recurring attack surface for privilege escalation. Driver compromise provides access at the boundary between the operating system and the accelerator, with implications for inference integrity, model weight confidentiality, and attestation output integrity.

For AI infrastructure, driver-level escalation matters because the driver mediates every interaction between the AI workload and the accelerator, and any compromise of this mediation surface affects the entire deployment.

Primary reference category: MITRE CVE Database entries for GPU driver vulnerabilities, including representative entries CVE-2021-1056 (NVIDIA GPU Display Driver local privilege escalation) and the ongoing series of NVIDIA, AMD, and Intel GPU driver CVE entries.

Supporting reference: Mittal, Abhinaya, Reddy, and Ali (2018), survey of techniques for improving GPU security.

Detection difficulty: moderate; CVE-driven patching disciplines are mature for known vulnerabilities but lag for novel discovery.

Software defense adequacy: mature for known vulnerabilities under maintained patching discipline; bounded against novel driver-layer compromise.

Physics-layer defense classes: hardware-rooted attestation that does not depend on driver integrity; physical-layer access mediation independent of driver state.

5.3 S3.T3 Boot-Chain Compromise

BIOS, UEFI, and accelerator boot firmware compromise provides an adversary with pre-operating-system persistence. Measured boot architectures (Intel TXT, AMD SME and SEV, ARM CCA, and equivalents) mitigate some variants but do not close all attack paths. UEFI-level malware families including LoJax, MosaicRegressor, and MoonBounce are documented in the public security research literature.

For AI infrastructure, boot-chain compromise matters because boot-time integrity is the foundation of every downstream attestation claim, and compromise at this layer cascades through the entire trust posture.

Primary standards reference: NIST SP 800-193 (2018), Platform Firmware Resiliency Guidelines.

Supporting standards references: NIST SP 800-147B (2014); UEFI Forum Specifications (current); Trusted Computing Group, TPM 2.0 Library Specification and PCR usage.

Supporting research reference: Binarly Research (2023 to 2024), LogoFAIL disclosure series.

Detection difficulty: high; by definition operates below the operating-system-level detection layer.

Software defense adequacy: limited.

Physics-layer defense classes: hardware-enforced measured boot to hardware root of trust; physical firmware integrity measurement; hardware-level UEFI attestation.

5.4 S3.T4 Runtime Firmware Modification

Modification of firmware at runtime, including BMC firmware modification, NIC firmware modification, and persistent UEFI modification through runtime services, represents a threat class distinct from boot-time firmware compromise. The threat operates against systems that have established trusted boot but face runtime modification of firmware components that the operating system cannot fully inspect.

For AI infrastructure, runtime firmware modification matters because long-running AI workloads create exposure windows over which adversarial runtime modification can be staged, and the persistence properties of firmware modification make it a particularly durable threat.

Primary standards reference: NIST SP 800-193 (2018), Platform Firmware Resiliency Guidelines, sections on runtime integrity protection.

Supporting research reference: continued Binarly Research disclosure series on runtime firmware modification across generations of hardware platforms.

Detection difficulty: high; runtime firmware visibility is limited by design.

Software defense adequacy: limited at the firmware layer itself.

Physics-layer defense classes: hardware-level firmware integrity monitoring at runtime; cryptographic firmware integrity chains anchored to hardware root of trust; physical isolation of firmware management pathways.

6. Attack Surface 4: Inference Integrity

Inference-integrity threats target the correctness of AI inference: whether the model that was supposed to run did run, on the inputs that were supposed to be processed, producing the outputs that those inputs actually generated, without adversarial modification at any stage of the computation. These threats are distinct from confidentiality threats (which target what the inference reveals) and from availability threats (which target whether inference happens at all).

For AI infrastructure, inference integrity matters because regulatory frameworks under development specifically require verifiable assertions that inference was performed correctly, and integrity attacks directly target the foundation those assertions need to rest on.

6.1 S4.T1 Hardware-Level Inference Manipulation

Manipulation of AI inference computation through hardware-layer attack pathways includes DRAM disturbance attacks (rowhammer and its variants), differential fault analysis applied to inference computation, and targeted electromagnetic or voltage manipulation of accelerator silicon during inference. Each pathway can induce specific computational errors that cause adversary-desired misclassifications, bypass safety-relevant computations, or extract information through fault-induced behavior. The rowhammer family is specifically a DRAM disturbance attack rather than a fault-injection attack in the cryptographic-protocol sense; both pathways are documented as distinct categories within the broader hardware-level inference manipulation surface.

For AI infrastructure, hardware-level manipulation matters because inference correctness is the load-bearing claim that regulatory frameworks require, and the hardware-level pathways through which manipulation can occur sit outside the threat models that software-level inference validation addresses.

Primary references: Kim et al. (2014); Boneh, DeMillo, and Lipton (1997).

Supporting reference: Breier et al. (2018), practical fault attack on deep neural networks.

Detection difficulty: moderate to high; sophisticated hardware-level manipulation evades conventional fault detection.

Software defense adequacy: bounded; software-level fault tolerance catches some classes but is bounded against well-targeted hardware-level campaigns.

Physics-layer defense classes: hardware fault detection at instruction level; redundant computation with hardware-level majority voting; physical shielding for electromagnetic and voltage manipulation; hardware-enforced inference integrity verification.

6.2 S4.T2 Inference Result Substitution

Substitution of inference results between the accelerator that computed them and the downstream consumer that relies on them, through man-in-the-middle attack on the inference response path or through adversarial substitution at intermediate processing stages, represents a threat class distinct from inference computation manipulation. The threat targets the integrity of the transport rather than the integrity of the computation.

For AI infrastructure, result substitution matters because the regulatory and compliance frameworks under development require evidence that the output a consumer relied on was the output the AI workload actually produced, and transport-layer substitution defeats this requirement at the cheapest available attack point.

Primary standards reference: NIST SP 800-175B Rev. 1 (2020), Guideline for Using Cryptographic Standards in the Federal Government, on cryptographic mechanisms relevant to substitution protection at the transport layer.

Supporting reference: standard man-in-the-middle threat modeling in protocol analysis literature, with foundational treatment in network security textbooks.

Detection difficulty: mature for cryptographically protected transport; bounded for transport that lacks cryptographic integrity protection.

Software defense adequacy: mature where cryptographic transport is deployed; limited where transport-layer integrity protection is absent.

Physics-layer defense classes: hardware-rooted cryptographic signing of inference results at the accelerator; tamper-evident transport from accelerator to consumer; hardware-anchored end-to-end integrity verification.

6.3 S4.T3 Software-Layer Compliance Bypass

The broadest inference-integrity concern is a structural gap rather than a specific attack: many AI safety, compliance, and policy mechanisms operate at the software layer (system prompts, guardrails, alignment training, output filtering) and can be bypassed through adversarial inputs, indirect prompt injection, or model-layer attacks that defeat the software enforcement mechanism. Where the regulatory framework requires compliance enforcement that cannot be bypassed, software-layer mechanisms are structurally insufficient. This threat class addresses bypass at the level of specific inference events; the durability of governance assertions across adversarial interaction patterns over time is addressed separately in Section 7.

For AI infrastructure, compliance bypass matters because regulatory frameworks increasingly require hardware-rooted evidence that compliance constraints actually held during inference, and software-layer mechanisms

cannot produce that evidence under the threat model they specifically operate against.

Primary references: Greshake et al. (2023), indirect prompt injection; Wei, Haghtalab, and Steinhardt (2023), jailbroken; Anil et al. (2024), many-shot jailbreaking.

Detection difficulty: varies; bounded at the software layer by the same compromise the threat model addresses.

Software defense adequacy: bounded; software-layer compliance mechanisms are subject to the same compromise as the assertions they produce.

Physics-layer defense classes: hardware-rooted compliance attestation; tamper-resistant policy state independent of software configuration; hardware-enforced policy primitives that cannot be bypassed by model-layer behavior.

6.4 S4.T4 Model Weight Extraction via Inference Side Channels

Extraction of model weights through observation of inference behavior across many queries, including query-based reconstruction attacks, electromagnetic side-channel observation of inference computation, and timing-side-channel observation of inference behavior. The threat class is distinct from direct memory extraction because it operates against the running inference behavior rather than against the weight storage.

For AI infrastructure, side-channel weight extraction matters because frontier model weights are among the most valuable IP assets in technology, and the side-channel pathways through which weights can be reconstructed sit outside the threat models that direct extraction defenses address.

Primary references: Tramèr, Zhang, Juels, Reiter, and Ristenpart (2016), foundational query-based model extraction; Batina, Bhasin, Jap, and Picek (2019), CSI NN electromagnetic side-channel reverse engineering of neural networks; Carlini et al. (2024), stealing part of a production language model.

Detection difficulty: varies by extraction pathway; query-based extraction can be detected through rate-limiting and pattern analysis; physical side-channel extraction is harder to detect.

Software defense adequacy: mature for query-based pathways with rate-limiting deployment; bounded against physical side-channel pathways.

Physics-layer defense classes: hardware-enforced weight protection during inference; physical side-channel shielding; hardware-rooted inference observation isolation.

7. Attack Surface 5: AI Governance Persistence

AI governance persistence threats target the durability of AI governance assertions across the operational lifecycle of a deployed AI system, including across operating system events, across model updates, across user sessions, and across adversarial interaction patterns. The threats are distinct from inference integrity threats because they address the durability of governance assertion rather than the correctness of any specific inference event.

For AI infrastructure, governance persistence matters because the regulatory frameworks under development require not only that AI workloads operate under specific governance constraints, but that those constraints can be evidenced as having operated continuously across the deployment lifecycle.

7.1 S5.T1 Jailbreak Persistence Across Operating System Compromise

Jailbreak attacks on AI systems that persist across operating system events, including across reboots, across configuration changes, and across session boundaries, represent a compliance threat distinct from session-scoped jailbreak. Many jailbreak mitigations are session-scoped and depend on operating system integrity for their persistence; under operating system compromise, these mitigations become operationally void.

For AI infrastructure, jailbreak persistence matters because the regulatory framework requires that governance constraints hold across the deployment lifecycle, and session-scoped mitigations cannot make that claim under adversarial conditions that operate below the session.

Primary reference: Zou, Wang, Kolter, and Fredrikson (2023), universal and transferable adversarial attacks on aligned language models.

Supporting references: Anil et al. (2024), many-shot jailbreaking; Shah, Feuillade-Montixi, Pour, Tagade, Casper, and Rando (2023), scalable and transferable black-box jailbreaks via persona modulation.

Detection difficulty: varies.

Software defense adequacy: mature at the session-scoped level; bounded at the persistent level under operating system compromise.

Physics-layer defense classes: hardware-rooted policy enforcement primitives with session-independent state; hardware-rooted policy enforcement that cannot be modified by model-layer behavior.

7.2 S5.T2 Alignment Constraint Software Bypass

Bypass of alignment constraints through adversarial inputs that target the software mechanism enforcing the constraints, including direct prompt injection, indirect prompt injection through processed content, and contextual confusion attacks. The threat class is distinct from jailbreak persistence because it targets a specific alignment mechanism rather than the durability of governance assertion in general. This threat class addresses bypass at the level of a specific alignment mechanism within an inference event; the structural gap in software-layer compliance mechanisms more broadly is addressed in Section 6.3.

For AI infrastructure, alignment bypass matters because regulatory frameworks require evidence that alignment constraints actually constrained the AI workload, and software-layer constraints cannot evidence this property under the threat conditions that alignment-bypass attacks specifically target.

Primary references: Perez and Ribeiro (2022), ignore previous prompt attack techniques; Greshake et al. (2023), indirect prompt injection.

Supporting standards reference: OWASP Foundation (2025), OWASP Top 10 for Large Language Model Applications v2.0.

Detection difficulty: moderate at the input layer; bounded at the alignment-mechanism layer.

Software defense adequacy: mature at input filtering and content moderation; bounded at alignment-mechanism integrity under adversarial input.

Physics-layer defense classes: hardware-enforced constraint enforcement that operates below the alignment-mechanism layer; tamper-resistant policy state across input-processing events.

7.3 S5.T3 Governance Enforcement Decoupling

Decoupling of AI governance enforcement from AI behavior at runtime, where the governance mechanism produces compliance assertions while the underlying AI behavior deviates from the assertions in ways the mechanism is not designed to detect. The threat class is distinct from explicit bypass because it operates through the mechanism's blind spots rather than against the mechanism directly.

For AI infrastructure, governance decoupling matters because the regulatory framework requires that compliance assertions correspond to actual behavior, and decoupling produces assertions that systematically diverge from behavior in ways that purely software-layer monitoring cannot detect.

Primary operational reference: EPA (2015), Notice of Violation to Volkswagen AG, the canonical example of governance enforcement decoupling at industrial scale.

Supporting academic reference: Contag et al. (2017), analysis of emission defeat devices, categorically applicable to AI governance defeat scenarios.

Detection difficulty: very high; decoupling is specifically designed to evade detection.

Software defense adequacy: bounded; the mechanism being decoupled is the same mechanism being relied on for assertion.

Physics-layer defense classes: hardware-rooted attestation that operates independently of the governance mechanism; hardware-enforced cross-validation of governance assertions against silicon-observable behavior; tamper-resistant compliance state independent of software configuration.

8. What This Taxonomy Is Not

The taxonomy is explicit about what it does not address:

It does not address adversarial machine learning as a primary topic. Adversarial inputs, prompt injection, membership inference, and related model-layer concerns are discussed where they intersect with physics-layer considerations, but a comprehensive adversarial machine learning taxonomy is out of scope and is addressed well by other references.

It does not provide a primary taxonomy of conventional kernel and hypervisor security threats. Those threats are extensively covered in the established information security literature and standards corpus, and the Physics-Layer Threat Taxonomy assumes them as adjacent prerequisite content rather than as primary scope.

It does not rank threats by frequency or expected impact. Practitioners should perform threat modeling specific to their deployment context; the taxonomy provides vocabulary for that work rather than substitute for it.

It does not prescribe specific defensive technologies. Physics-layer defense classes are identified for each threat class; specific implementations vary widely and are evolving rapidly.

It is not exhaustive. The five attack surfaces and enumerated threat classes cover what the authors judge to be the most material physics-layer threat surface at the time of publication. New threat classes will emerge; existing classes will evolve. The taxonomy is designed to accommodate additions through the planned quarterly State of Physics-Layer AI Trust updates.

It is not a compliance framework. The taxonomy is a technical reference that compliance frameworks can map against, not a compliance framework itself.

9. Using the Taxonomy

The taxonomy is intended to support several use cases:

For security architects: vocabulary for internal threat modeling of AI infrastructure deployments, and a checklist against which coverage gaps can be identified.

For procurement teams: reference against which vendor security claims can be evaluated. A vendor who cannot articulate threat model against the taxonomy is likely operating with less rigor than a vendor who can.

For regulators and standards bodies: shared vocabulary for technical requirements across the FY2026 NDAA Section 1513 framework development at the Department of Defense, OMB AI procurement guidance under Executive Order 14179, EU AI Act Articles 40 and 43, NATO STANAG, ISO/IEC 27090, ISO/IEC JTC 1/SC 42, and national frameworks. The taxonomy is explicitly designed to be compatible with standards development work; Ziru Labs is preparing submissions to relevant standards bodies and intends to contribute to the technical work as it matures.

For academic researchers: structured reference for identifying underexplored threat classes and for positioning new research. The taxonomy maps to the existing academic security literature while also surfacing where gaps exist relative to AI infrastructure specifically.

For investors and strategic analysts: framework for evaluating claims about AI infrastructure security by specific companies and specific technologies. The taxonomy's emphasis on explicit scoping (what a defense does address and what it does not) is intentional and is the right posture for sophisticated evaluation.

10. Versioning and Evolution

This framework is v1.0. Subsequent versions will be published as the physics-layer threat landscape evolves. The planned quarterly State of Physics-Layer AI Trust report tracks threat-landscape evolution against this taxonomy between major-version updates.

Contributions and corrections are welcome. Ziru Labs is specifically interested in additional threat classes warranting inclusion, primary-source references strengthening existing threat class documentation, and academic or industry work on defensive mechanisms that should be reflected in the defense-class discussion.

Contact: research@zirulabs.com.

11. Relationship to Ziru Labs Technology

Ziru Labs is developing technology designed to operate as one operational response to threats in this taxonomy. The substrate is pre-commercial, with a minimum working prototype targeted for the second half of 2026; the coverage described here is at the level of architectural design rather than capability demonstrated in a shipped product. Ziru Labs' substrate is designed to address, in particular, threat classes in Attack Surfaces 1 (Physical Access), 2 (Bus and Interconnect), portions of 3 (Firmware and Boot), 4 (Inference Integrity), and 5 (AI Governance Persistence). It is not designed to address adversarial machine learning at the model layer or software-layer kernel and hypervisor security. Those concerns are addressed by other technologies that stack with Ziru Labs technology in defense-in-depth architectures.

The taxonomy is a reference framework. Ziru Labs technology is one operational response designed against it, offered in the convening posture described in the companion Reference Framework rather than as the definitive occupant of the category.

12. Bibliography

References are organized into five categories: academic literature (A), standards and regulatory documents (B), industry and vendor technical documentation (C), historical and analytical reporting (D), and framework integration references (E).

A. ACADEMIC LITERATURE

Anil, C., Durmus, E., Panickssery, N., Sharma, M., and co-authors (2024). "Many-shot Jailbreaking." Advances in Neural Information Processing Systems 37 (NeurIPS 2024).

Batina, L., Bhasin, S., Jap, D., and Picek, S. (2019). "CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel." Proceedings of the USENIX Security Symposium.

Boneh, D., DeMillo, R.A., and Lipton, R.J. (1997). "On the Importance of Checking Cryptographic Protocols for Faults." Advances in Cryptology, EUROCRYPT '97.

Breier, J., Jap, D., Hou, X., Bhasin, S., and Liu, Y. (2018). "Practical Fault Attack on Deep Neural Networks." Proceedings of the ACM Conference on Computer and Communications Security (CCS).

Carlini, N., Paleka, D., Dvijotham, K.D., Steinke, T., Hayase, J., Cooper, A.F., Lee, K., Jagielski, M., Nasr, M., and co-authors (2024). "Stealing Part of a Production Language Model." Proceedings of the International Conference on Machine Learning (ICML).

Contag, M., Li, G., Pawlowski, A., Domke, F., Levchenko, K., Holz, T., and Savage, S. (2017). "How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles." Proceedings of the IEEE Symposium on Security and Privacy.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. (2023). "Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." ACM Workshop on Artificial Intelligence and Security (AISec).

Gruhn, M. and Müller, T. (2013). "On the Practicability of Cold Boot Attacks." Eighth International Conference on Availability, Reliability and Security (ARES).

Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., and Felten, E.W. (2008). "Lest We Remember: Cold Boot Attacks on Encryption Keys." Proceedings of the 17th USENIX Security Symposium.

Mittal, S., Abhinaya, S.B., Reddy, M., and Ali, I. (2018). “A Survey of Techniques for Improving Security of GPUs.” Journal of Hardware and Systems Security 2, pp. 266-285.

Jiang, Z.H., Fei, Y., and Kaeli, D. (2017). “A Novel Side-Channel Timing Attack on GPUs.” Great Lakes Symposium on VLSI (GLSVLSI).

Kallenberg, C. and Kovah, X. (2015). “How Many Million BIOSes Would You Like to Infect?” CanSecWest.

Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J.H., Lee, D., Wilkerson, C., Lai, K., and Mutlu, O. (2014). “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors.” International Symposium on Computer Architecture (ISCA).

Kocher, P., Jaffe, J., and Jun, B. (1999). “Differential Power Analysis.” Proceedings of CRYPTO ’99.

Markettos, A.T., Rothwell, C., Gutstein, B.F., Pearce, A., Neumann, P.G., Moore, S.W., and Watson, R.N.M. (2019). “Thunderclap: Exploring Vulnerabilities in Operating System IOMMU Protection via DMA from Untrustworthy Peripherals.” Network and Distributed System Security Symposium (NDSS).

Naghibijouybari, H., Neupane, A., Qian, Z., and Abu-Ghazaleh, N. (2018). “Rendered Insecure: GPU Side Channel Attacks are Practical.” ACM Conference on Computer and Communications Security (CCS).

Okhravi, H. and Nicol, D. (2018). “Supply Chain Risks in Hardware, Software, and the Cloud.” IEEE Computer.

Perez, F. and Ribeiro, I. (2022). “Ignore Previous Prompt: Attack Techniques For Language Models.” NeurIPS Machine Learning Safety Workshop.

Quisquater, J.-J. and Samyde, D. (2001). “ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards.” E-smart 2001 International Conference.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds.” ACM Conference on Computer and Communications Security (CCS).

Shah, R., Feuillade-Montixi, Q., Pour, S., Tagade, A., Casper, S., and Rando, J. (2023). “Scalable and Transferable Black-Box Jailbreaks for Language Models via Persona Modulation.” arXiv:2311.03348; published at SoLaR Workshop, NeurIPS 2023.

Skorobogatov, S. (2005). “Semi-invasive attacks: A new approach to hardware security analysis.” University of Cambridge Computer Laboratory Technical Report UCAM-CL-TR-630.

Smith, S.W. and Weingart, S. (1999). “Building a High-Performance, Programmable Secure Coprocessor.” *Computer Networks*, Special Issue on Network Security, 31, pp. 831-860.

Stewin, P. and Bystrov, I. (2013). “Understanding DMA Malware.” Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA).

Tarnovsky, C. (2010). “Deconstructing a ’Secure’ Processor.” Black Hat DC.

Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., and Ristenpart, T. (2016). “Stealing Machine Learning Models via Prediction APIs.” Proceedings of the USENIX Security Symposium.

Wei, A., Haghtalab, N., and Steinhardt, J. (2023). “Jailbroken: How Does LLM Safety Training Fail?” Proceedings of Neural Information Processing Systems (NeurIPS).

Yitbarek, S.F., Aga, M.T., Das, R., and Austin, T. (2017). “Cold Boot Attacks are Still Hot: Security Analysis of Memory Scramblers in Modern Processors.” IEEE International Symposium on High Performance Computer Architecture (HPCA).

Zou, A., Wang, Z., Kolter, J.Z., and Fredrikson, M. (2023). “Universal and Transferable Adversarial Attacks on Aligned Language Models.” arXiv:2307.15043.

B. STANDARDS AND REGULATORY DOCUMENTS

CISA-NIST (2021). “Defending Against Software Supply Chain Attacks.” Cybersecurity and Infrastructure Security Agency joint publication with NIST.

CXL Consortium (2022). “Compute Express Link (CXL) Specification, Revision 3.0,” including the CXL IDE (Integrity and Data Encryption) Specification.

EPA (2015). “Notice of Violation to Volkswagen AG.” U.S. Environmental Protection Agency, September 18, 2015.

Executive Order 14028 (2021). “Improving the Nation’s Cybersecurity.” The White House.

Executive Order 14179 (2025). “Removing Barriers to American Leadership in Artificial Intelligence.” The White House, signed 23 January 2025; 90 FR 8741.

Executive Order 14365 (2025). “Ensuring a National Policy Framework for Artificial Intelligence.” The White House, signed 11 December 2025; 90 FR 58499.

ISO/IEC (2022). “ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection, Information security management systems, Requirements.”

ISO/IEC (2022). “ISO/IEC 15408:2022, Information security, cybersecurity and privacy protection, Evaluation criteria for IT security.”

ISO/IEC (2023). “ISO/IEC 42001:2023, Information technology, Artificial intelligence, Management system.”

NIST (2014). “NIST SP 800-147B: BIOS Protection Guidelines for Servers.”

NIST (2018). “NIST SP 800-37 Rev. 2: Risk Management Framework for Information Systems and Organizations.”

NIST (2018). “NIST SP 800-193: Platform Firmware Resiliency Guidelines.”

NIST (2019). “FIPS 140-3: Security Requirements for Cryptographic Modules.”

NIST (2020). “NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations.”

NIST (2020). “NIST SP 800-175B Rev. 1: Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms.”

NIST (2022). “NIST SP 800-161 Rev. 1: Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations.”

NIST (2022). “NIST SP 800-218: Secure Software Development Framework (SSDF) Version 1.1.”

NIST (2023). “AI Risk Management Framework (AI RMF 1.0),” NIST AI 100-1.

NIST (2024). “AI RMF Playbook.”

NIST (2024). “NIST SP 800-171 Rev. 3: Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations” and “NIST SP 800-172: Enhanced Security Requirements for Protecting Controlled Unclassified Information.”

OWASP Foundation (2025). “OWASP Top 10 for Large Language Model Applications v2.0.”

Trusted Computing Group (current). “TPM 2.0 Library Specification” and “Platform Configuration Register (PCR) usage guidance.”

UEFI Forum (current). “UEFI Platform Initialization Specification” and “UEFI Specification.”

C. INDUSTRY AND VENDOR TECHNICAL DOCUMENTATION

CXL Consortium (current). “CXL IDE (Integrity and Data Encryption) Specification” within CXL 3.0.

MITRE Corporation (2024). “ATT&CK Framework, Enterprise Matrix,” Version 14.

NVIDIA Corporation (current). “NVIDIA H100 Tensor Core GPU Architecture” technical whitepaper.

NVIDIA Corporation (current). “NVIDIA Confidential Computing Deployment Guide.”

NVIDIA Corporation (current). “NVIDIA NVLink and NVSwitch” technical brief.

NVIDIA Corporation (current). “NVIDIA NVLink-C2C Interconnect Technology” documentation.

NVIDIA Corporation (current). “Multi-Instance GPU (MIG) User Guide.”

NVIDIA Corporation (current). “GSP Firmware Architecture” technical documentation.

D. HISTORICAL AND ANALYTICAL REPORTING

Appelbaum, J., Horchert, J., and Stöcker, C. (2013). “Shopping for Spy Gear: Catalog Advertises NSA Toolbox.” Der Spiegel, December 29, 2013.

Binarly Research (2023 to 2024). “LogoFAIL: The Dangers of Image Parsing During System Boot” and the continued Binarly disclosure series on UEFI and runtime firmware vulnerabilities.

Greenberg, A. (2014). “Photos of an NSA ’Upgrade’ Factory Show Cisco Router Getting Implant.” WIRED, May 15, 2014.

NSA TAO ANT Catalog (2008, publicly disclosed 2013). Publicly documented supply-chain and chassis-level implant capabilities across hardware categories.

MITRE CVE Database. GPU driver vulnerability entries including CVE-2021-1056 and the ongoing series of NVIDIA, AMD, and Intel GPU driver CVE entries.

Ziru Labs (2026). *The Trust Layer for AI: A Reference Framework for the Category, v1.0.* Published at zirulabs.com/research.

Ziru Labs (2026). *The Runtime Verification Gap in Federal AI Deployment: A Reference Primer, v1.0.* Published at zirulabs.com/research.

Ziru Labs (2026). *The AI Infrastructure Stack and the Trust Layer Position: A Reference Primer, v1.0.* Published at zirulabs.com/research.

Ziru Labs (2026). *The Trust Layer Category Map: A Reference Primer, v1.0.* Published at zirulabs.com/research.

Ziru Labs (2026). *Ziru Labs Capability Posture: A Reference Primer, v1.0.* Published at zirulabs.com/research.

E. FRAMEWORK INTEGRATION REFERENCES

MITRE Corporation (2024). “ATT&CK Framework, Enterprise Matrix,” Version 14.

NIST (2024). “The NIST Cybersecurity Framework (CSF) 2.0,” NIST CSWP 29.

NIST (2023). “AI Risk Management Framework (AI RMF 1.0),” NIST AI 100-1.

OWASP Foundation (2021). “OWASP Top 10:2021.”

OWASP Foundation (2025). “OWASP Top 10 for Large Language Model Applications v2.0.”

ISO/IEC (2023). “ISO/IEC 42001:2023, Information technology, Artificial intelligence, Management system.”

ISO/IEC (2022). “ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection, Information security management systems, Requirements.”

ISO/IEC (2022). “ISO/IEC 15408:2022, Information security, cybersecurity and privacy protection, Evaluation criteria for IT security.”

IEC (2018 to ongoing). “IEC 62443, Security for industrial automation and control systems.”

NIST (2019). “FIPS 140-3: Security Requirements for Cryptographic Modules.”

Acknowledgments

This framework draws on published academic literature, government standards, public research, and the founding team’s prior experience spanning U.S. federal classified AI, signals intelligence, cryptographic warfare, financial services, and AI infrastructure. Specific acknowledgments to the Halderman et al. cold-boot research community, the DARPA Trust in IC program outcomes, the IEEE HOST community, the CXL Consortium technical discussions, and the ongoing EU AI Act technical development process under Articles 40 and 43. Initial draft review provided by the Ziru Labs founding team. Any errors in characterization are solely Ziru Labs’ responsibility.

Citation

Ziru Labs. *The Physics-Layer Threat Taxonomy for AI Infrastructure: A Reference Framework, v1.0.* Published at zirulabs.com/research.

