Article

Zero Trust for AI Systems: A Reference Architecture and Assurance Framework

Robert Campbell *

Posted Date: 2 February 2026

doi: 10.20944/preprints202602.0085.v1

Keywords: Zero Trust Architecture; artificial intelligence security; model supply chain; cryptographic provenance; NIST AI RMF; DoD Zero Trust strategy; AI assurance

 doi:10.20944/preprints202602.0085.v1

Article

Zero Trust for AI Systems: A Reference Architecture and Assurance Framework

Robert Campbell

Independent Researcher, Upper Marlboro, MD 20774, USA; rc@medcybersecurity.com

Abstract

Artificial intelligence systems are rapidly becoming integral to defense, intelligence, and critical infrastructure, yet existing cybersecurity frameworks provide limited guidance for securing AI-specific components such as model supply chains and training data pipelines. While Zero Trust Architecture (ZTA) offers a powerful foundation for modern cybersecurity, and while secure MLOps practices and ZT-adjacent controls for ML pipelines have emerged in practitioner literature, these efforts have not been integrated into a comprehensive framework that decomposes AI systems into distinct trust layers, specifies evidence artifacts per layer, and provides compliance crosswalks to federal AI governance requirements. This paper addresses that integration gap by proposing a comprehensive Zero Trust framework tailored to the AI lifecycle. We introduce a structured threat model identifying adversarial opportunities across AI workflows and map Zero Trust principles—identity, continuous verification, least privilege, micro-segmentation, and policy enforcement—to AI-specific components. We present a reference Zero Trust Architecture composed of four trust layers: Data Trust, Model Supply Chain Trust, Pipeline Trust, and Inference Trust. We further define an assurance evidence framework integrating cryptographic provenance, continuous integrity monitoring, and policy-driven access control to produce audit artifacts intended to support alignment assessments against NIST AI RMF, DoD Zero Trust guidance, and ISO/IEC 42001 requirements. A scenario-based demonstration illustrates threat mitigation in mission environments. This work establishes a foundation for standardized Zero Trust implementations for AI systems.

Keywords: Zero Trust Architecture; artificial intelligence security; model supply chain; cryptographic provenance; NIST AI RMF; DoD Zero Trust strategy; AI assurance

1. Introduction

The integration of artificial intelligence into mission-critical systems represents one of the most significant technological transformations in recent history. Across defense, intelligence, healthcare, and critical infrastructure sectors, AI systems now support decisions ranging from threat detection and logistics optimization to medical diagnostics and autonomous operations [1,2]. Executive Order 14110, which catalyzed multiple federal AI governance programs, was revoked on 20 January 2025 by Executive Order 14148 [52]; however, the risk-management and security control objectives it highlighted continue to be addressed through NIST guidance and DoD Zero Trust policy and implementation direction [2,10,18,51]. A GAO review reports that DoD's initial unclassified AI inventory (limited to projects funded through RDT&E and procurement accounts) identified 685 AI projects as of April 2021, underscoring the scale and governance challenge of AI adoption in defense enterprises [3]. Adoption continues to accelerate under initiatives such as the Joint All-Domain Command and Control (JADC2) and the Chief Digital and Artificial Intelligence Office (CDAO). Per the Stanford HAI AI Index 2025 report, U.S. private AI investment reached $109.1 billion in 2024, and 78% of organizations reported using AI in at least one business function in 2024 [4], underscoring the pace of adoption and the urgency of operationalizing AI security controls. This urgency is not limited to the United States: the European Union AI Act establishes risk-based compliance requirements for AI systems

 doi:10.20944/preprints202602.0085.v1

deployed in the EU [62], and the G7 has initiated work toward a common framework for AI Bills of Materials (AI-BOM)—structured inventories capturing model provenance, training data lineage, and algorithmic dependencies analogous to Software Bills of Materials (SBOM) for traditional software—to enable supply-chain transparency across jurisdictions [40]. The architecture presented in this paper is grounded in U.S. federal guidance (NIST, DoD, NSA) but employs technology-neutral control objectives and evidence specifications designed to support alignment assessments across international frameworks, including the EU AI Act, UK NCSC AI security guidance, and emerging G7 standards. Organizations outside U.S. federal jurisdiction should map the control objectives to their applicable regulatory requirements.

This rapid adoption, however, has outpaced the development of security architectures capable of addressing AI-specific risks. Traditional cybersecurity frameworks were designed for conventional software systems where code is deterministic, dependencies are well-understood, and integrity can be verified through established means such as code signing and software bills of materials (SBOMs). AI systems introduce fundamentally different challenges: models learn behavior from training data rather than explicit programming, dependencies extend beyond software libraries to include datasets, pre-trained weights, and algorithmic configurations, and the integrity of a model cannot be verified simply by examining its code [6].

Zero Trust Architecture (ZTA), as codified in NIST Special Publication 800-207 [7], has emerged as the dominant cybersecurity paradigm for modern enterprises. The core principle, articulated by Forrester Research analyst John Kindervag in 2010 [8] and formalized by NIST in 2020, is often summarized as "*never trust, always verify*," which eliminates implicit trust based on network location or asset ownership, instead requiring continuous authentication, authorization, and validation of all entities and transactions [7]. Executive Order 14028 mandated the adoption of Zero Trust across federal agencies [9], and the Department of Defense's Zero Trust Strategy established target-level implementation objectives with fiscal year 2027 as the planning horizon for DoD components [10]. The accompanying Zero Trust Capability Execution Roadmap details 45 capabilities and 152 activities spanning seven pillars: User, Device, Network/Environment, Application/Workload, Data, Visibility/Analytics, and Automation/Orchestration [12].

Despite this comprehensive framework, NIST SP 800-207 and its associated guidance provide minimal direction for AI-specific security challenges. The document makes no mention of model artifacts, training data provenance, or inference-time threats. Supplementary publications such as NIST SP 800-207A address cloud-native applications and multi-cloud environments [13], but none extend Zero Trust principles to the unique components and workflows of AI systems. This gap leaves organizations attempting to secure AI workloads without authoritative architectural guidance, resulting in inconsistent implementations and residual risk exposure.

The AI security research community has extensively documented threats to machine learning systems, including adversarial examples, data poisoning, model extraction, and backdoor attacks [14–16]. NIST AI 100-2e2025, "Adversarial Machine Learning: A Taxonomy and Terminology," provides a comprehensive categorization of attack vectors [17]. However, this body of work has developed largely in isolation from Zero Trust principles, focusing on algorithmic defenses rather than architectural controls. The NIST AI Risk Management Framework (AI RMF) offers guidance on trustworthy AI development but does not prescribe specific security architectures or map its recommendations to Zero Trust implementations [18,19].

This paper addresses the identified gap by proposing a reference Zero Trust Architecture for AI Systems. In this context, "reference architecture" denotes explicitly defined components, trust boundaries, evidence artifacts, and traceable mappings between threats, controls, and compliance requirements—not a formal specification in the mathematical sense, nor a vendor-specific implementation.

 doi:10.20944/preprints202602.0085.v1

1.1. Novelty Statement

The contribution of this work is *not* simply "applying Zero Trust principles to AI systems," as numerous papers have claimed at a conceptual level. Rather, our novelty lies in three specific, verifiable artifacts: (a) a **lifecycle-aligned Zero Trust decomposition** that treats data, models, pipelines, and inference services as first-class trust subjects with distinct identity, verification, and policy enforcement requirements—not as generic "applications" or "workloads"; (b) an **audit-oriented evidence package design** that specifies, for each control, the concrete artifacts (signatures, attestations, logs, manifests) required to demonstrate compliance—not abstract control objectives; and (c) a **maturity model tailored to AI system characteristics** (model provenance, CBOM documentation, runtime attestation) that extends beyond general Zero Trust maturity frameworks. These three artifacts are independently valuable: practitioners can assess whether the decomposition covers their AI assets, whether the evidence packages are producible with their tooling, and whether the maturity levels align with their compliance timelines.

1.2. Contributions

Our contributions are fourfold:

1. **A comprehensive threat model** that categorizes adversaries, attack surfaces, and failure modes specific to AI systems operating in Zero Trust environments, with explicit traceability to authoritative taxonomies (Section 3).
2. **A lifecycle-aligned control mapping** that assigns Zero Trust principles—identity, continuous verification, least privilege, micro-segmentation, and policy enforcement—to AI-specific trust subjects (data, models, pipelines, inference), not generic IT assets (Section 4).
3. **A four-layer architecture with evidence specifications** comprising Data Trust, Model Supply Chain Trust, Pipeline Trust, and Inference Trust layers, each with defined controls, Policy Enforcement Points, and the specific evidence artifacts required for audit (Section 5).
4. **An AI-tailored assurance evidence framework and maturity model** that establishes metrics (DAI, MIS), evidence types, compliance crosswalks, and maturity levels (L0–L3) defined by AI-specific characteristics rather than generic Zero Trust capabilities (Section 6).

Existing Zero Trust guidance—including NIST SP 800-207, the CISA Zero Trust Maturity Model, and the NSA Zero Trust Implementation Guidelines—defines control planes for enterprise IT infrastructure but does not specify AI lifecycle assets as discrete trust subjects, does not address model supply-chain provenance or cryptographic bill of materials requirements, and does not define inference-time policy enforcement artifacts or AI-specific evidence packages. Similarly, AI security frameworks address threats but not architecture: the NIST AI RMF provides a risk management process but not a security architecture; OWASP Top 10 for LLM Applications and MITRE ATLAS provide threat and attack catalogs but not implementation guidance; Google's Secure AI Framework (SAIF) offers conceptual control domains but not traceable evidence artifacts or compliance crosswalks. Our contribution uniquely combines: (a) an architecture specification that decomposes AI systems into auditable trust subjects; (b) a threat-to-control-to-evidence traceability model that specifies what artifacts must be produced; and (c) explicit compliance crosswalks to NIST AI RMF, DoD Zero Trust guidance, and ISO/IEC 42001 requirements—enabling organizations to evaluate the framework against their specific assets, tooling, and compliance obligations.

1.3. Compliance Scope Note

This framework supports evidence collection for alignment assessments; it does not constitute a compliance certification program. Compliance decisions remain authority-dependent, scope-dependent, and context-dependent. The crosswalks presented (Table 3) identify mapping opportunities between framework controls and regulatory requirements; actual audit acceptance depends on the specific authorizing official, assessment scope, organizational risk tolerance, and operational environment. Organizations should interpret this work as architectural guidance that facilitates compliance docu-

 doi:10.20944/preprints202602.0085.v1

mentation, not as a prescriptive checklist that guarantees regulatory approval. Throughout this paper, terms such as 'compliance mapping' and 'compliance crosswalk' refer to alignment opportunities, not certification guarantees.

1.4. Novelty vs. Prior Work

To clarify what is *new* in this work versus what is synthesized from existing guidance, we distinguish three categories of prior work and our specific contributions beyond each:

(1) General Zero Trust Guidance (NIST SP 800-207, DoD ZT Strategy, NSA ZIGs): These documents define Zero Trust principles, logical components (PDP/PEP/PE), and maturity models for *enterprise IT infrastructure*. They do not: decompose AI systems into distinct trust subjects; specify model provenance or CBOM requirements; define AI-specific evidence artifacts; or address inference-time policy enforcement for ML workloads. **Our contribution:** We extend Zero Trust to AI by defining four trust layers (Data, Model Supply Chain, Pipeline, Inference) as first-class policy subjects, specifying PEP placement and evidence artifacts for each layer, and providing the first CBOM specification integrated with Zero Trust policy enforcement.

(2) AI Risk Management Frameworks (NIST AI RMF, ISO/IEC 42001): These frameworks provide *risk management processes* (Govern, Map, Measure, Manage) and management system requirements, but they do not prescribe security architectures, specify cryptographic evidence artifacts, or define policy enforcement mechanisms. **Our contribution:** We operationalize AI RMF and ISO 42001 requirements into implementable controls with explicit evidence artifacts, providing the 'how' that complements their 'what.' Table 3 provides the first clause-level crosswalk from Zero Trust AI controls to these frameworks.

(3) AI Security Checklists and Taxonomies (OWASP Top 10 for LLMs, MITRE ATLAS, NIST AI 100-2): These resources catalog *threats and attack techniques* but do not prescribe architectural controls, specify evidence artifacts, or map to compliance frameworks. They answer, 'what can go wrong,' but not 'how to architect defenses' or 'how to prove compliance.' **Our contribution:** We provide the missing architectural layer—a systematic mapping from each threat class to specific Zero Trust controls, enforcement points, and auditable evidence artifacts (Table 2), with full traceability to source taxonomies via technique IDs.

In summary, this work is *not* merely a synthesis of existing guidance; it is an **architectural integration** that produces novel artifacts (four-layer trust decomposition, AI-specific PEP specifications, CBOM/ML-BOM minimum fields, evidence package templates, compliance crosswalks, Agent Delegation Token specification) that do not exist in any single prior work. The contribution is the integration itself—enabling organizations to implement Zero Trust for AI with traceable, auditable, compliance-mapped controls for the first time.

1.5. Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews background material on Zero Trust Architecture, the AI system lifecycle, existing AI security literature, and positions this work relative to related efforts (gap analysis), and presents our methodology for constructing the threat-to-control mapping. Section 3 presents our threat model. Section 4 maps Zero Trust principles to AI components. Section 5 details the proposed architecture. Section 6 describes the assurance evidence framework. Section 7 demonstrates application through a scenario. Section 8 discusses benefits, limitations, and future work. Section 9 concludes.

2. Background and Related Work

2.1. Zero Trust Architecture

Zero Trust represents a paradigm shift from perimeter-based security to identity-centric, continuous verification models. The foundational premise, articulated by Forrester Research analyst John Kindervag in 2010 [8] and formalized by NIST in 2020 [7], is often summarized as "never trust,

 doi:10.20944/preprints202602.0085.v1

always verify" regardless of whether access requests originate inside or outside traditional network boundaries.

NIST SP 800-207 defines Zero Trust Architecture through seven tenets [7]: (1) All data sources and computing services are considered resources; (2) All communication is secured regardless of network location; (3) Access to individual enterprise resources is granted on a per-session basis; (4) Access to resources is determined by dynamic policy; (5) The enterprise monitors and measures the integrity and security posture of all owned and associated assets; (6) All resource authentication and authorization are dynamic and strictly enforced before access is allowed; and (7) The enterprise collects as much information as possible about the current state of assets, network infrastructure, and communications and uses it to improve its security posture.

The architecture comprises three core logical components: the Policy Engine (PE), which makes access decisions based on policy and contextual information; the Policy Administrator (PA), which establishes and terminates communication paths based on PE decisions; and the Policy Enforcement Point (PEP), which enables, monitors, and terminates connections between subjects and resources [7]. Supporting infrastructure includes identity providers, security information and event management (SIEM) systems, threat intelligence feeds, and endpoint detection and response (EDR) capabilities.

The Cybersecurity and Infrastructure Security Agency (CISA) expanded on this foundation with the Zero Trust Maturity Model, which defines progression across five pillars—Identity, Devices, Networks, Applications and Workloads, and Data—from Traditional to Advanced and Optimal maturity levels [21]. The DoD Zero Trust Strategy expanded to seven pillars and established target-level and advanced-level implementation milestones [10,12]. NIST SP 1800-35, "Implementing a Zero Trust Architecture," provides practical implementation guidance through reference architectures demonstrated with commercial and open-source products [20,22].

Despite this comprehensive framework, existing Zero Trust guidance exhibits significant limitations when applied to AI systems. The architecture assumes resources are discrete, identifiable entities whose access requirements can be specified through traditional policy constructs. AI systems challenge this assumption in several ways: model artifacts are not traditional software and cannot be meaningfully inspected through code review; data is both input and determinant of behavior; trust relationships span the entire lifecycle; and identity for non-human entities is underdeveloped.

2.2. AI System Lifecycle

AI system development follows a lifecycle distinct from traditional software engineering, with stages that introduce unique security considerations [5,23,24]. **Data Collection and Preparation** involves acquiring training data from internal sources, public datasets, third-party providers, or synthetic generation. Data quality, representativeness, and integrity directly influence model behavior, and poisoning attacks at this stage can embed persistent vulnerabilities [15,17]. **Model Training** processes prepared data to learn parameters that optimize performance on specified objectives. Training occurs in specialized environments with significant computational requirements, often leveraging cloud infrastructure or dedicated GPU clusters.

Validation and Testing evaluate trained models against held-out datasets, benchmark suites, and domain-specific acceptance criteria. Validation assesses performance metrics but typically does not address security properties such as robustness to adversarial inputs or presence of backdoors. **Deployment** packages validated models for production environments through containerization, model serving infrastructure, or edge deployment. **Inference** processes input data and generates predictions, exposing models to potentially adversarial inputs. **Monitoring and Maintenance** tracks production models for performance degradation, data drift, and anomalous behavior.

The model supply chain encompasses all components, dependencies, and processes contributing to a deployed model [25,26]. Unlike traditional software supply chains documented through SBOMs, AI supply chains include pre-trained models and foundation models, training datasets and data pipelines, model architectures and hyperparameter configurations, training frameworks and libraries, hardware dependencies, and inference runtime environments. Recent incidents have demonstrated

 doi:10.20944/preprints202602.0085.v1

practical risks: vulnerabilities disclosed in early 2025 revealed malicious ML models uploaded to Hugging Face that exploited Python Pickle serialization to execute arbitrary code when loaded, with detailed analysis presented in the ACM CCS 2025 proceedings, illustrating a practical model-supply-chain execution vector [28,54].

2.3. AI Security and Assurance Literature

The adversarial machine learning research community has extensively characterized threats to AI systems [14–17]. NIST AI 100-2e2025 provides an authoritative taxonomy distinguishing: **Evasion attacks** that craft inputs to cause misclassification at inference time; **Poisoning attacks** that manipulate training data to influence model behavior, including backdoor attacks; **Privacy attacks** that extract sensitive information about training data or model parameters; and **Integrity attacks** that compromise model artifacts directly through tampering [17].

Research on defenses has produced techniques including adversarial training, certified robustness, differential privacy, and anomaly detection [30,33,34]. However, these algorithmic defenses address specific attack vectors rather than providing a comprehensive security architecture. The disconnect between AI security research and enterprise security practice has been noted as a significant barrier to operational deployment of trustworthy AI systems [35].

2.4. Gaps in Current Approaches

Existing Zero Trust guidance (NIST SP 800-207, DoD ZT Strategy) provides comprehensive coverage for identity verification, network segmentation, device posture, and application access control, but offers minimal direction for AI-specific concerns such as model provenance, dataset lineage, training pipeline integrity, or inference-time abuse detection. Analysis of existing frameworks reveals several critical gaps that this paper addresses: **No widely adopted reference architecture** integrating Zero Trust and AI security; **Insufficient provenance mechanisms** as SBOM frameworks were designed for traditional software [36,37]; **No continuous trust evaluation** for AI systems during runtime; **Limited model identity** constructs that bind to verified model state; and **Compliance mapping deficits** requiring organizations to satisfy multiple frameworks simultaneously [10,17,40].

The NSA Zero Trust Implementation Guidelines (ZIGs), published in January 2026 as a document set (Primer, Discovery, Phase One, Phase Two), provide implementation guidance for the DoD Zero Trust Strategy's 152 activities as defined in the Capability Execution Roadmap [12,56]. However, a keyword search (terms: "post-quantum," "PQC," "quantum-resistant," "CNSA 2.0") across the four-document ZIG set (accessed January 2026) identified limited explicit PQC integration guidance: the term "post-quantum" appears in the "Considerations" section of the ZIG guidance for Activity 5.4.4 (Protect Data in Transit) [56], with no other activities explicitly addressing cryptographic algorithm migration or quantum-resistant requirements. No provisions address AI-specific security requirements. The ZIG guidance for Enterprise PKI deployment (Activity 1.9.1) establishes a cryptographic infrastructure without algorithm agility provisions, creating technical debt for both PQC migration and AI model signing operations.

2.5. Related Work and Gap Analysis

This section positions the proposed framework relative to three categories of prior work: (1) Zero Trust Architecture guidance for enterprise IT, (2) AI security threat taxonomies, and (3) secure MLOps practices. We identify specific gaps that motivate the architecture and evidence framework presented in subsequent sections.

Zero Trust Architecture Guidance. The foundational Zero Trust literature establishes principles of continuous verification, least-privilege access, and assume-breach posture. Kindervag's seminal Forrester work [8] introduced the "never trust, always verify" paradigm. NIST SP 800-207 [7] codified seven tenets and defined the PDP/PEP enforcement model. CISA's Zero Trust Maturity Model [21] provides progression criteria across five pillars. DoD's Zero Trust Strategy [10] and Reference Architecture [11] translate these principles into federal implementation guidance. NIST SP 1800-35

 doi:10.20944/preprints202602.0085.v1

[22] demonstrates reference implementations with commercial products. *Gap:* These works address enterprise IT assets (users, devices, networks, applications, data) but do not decompose AI-specific trust subjects—models as versioned artifacts with provenance, training data with lineage, inference pipelines with runtime state, or autonomous agents with delegated authority. The "Application and Workload" pillar subsumes AI systems without distinguishing their unique verification requirements. None specifies evidence artifacts for model integrity, training data authenticity, or inference-time policy decisions.

AI Security Threat Taxonomies. Substantial research characterizes adversarial machine learning threats. Biggio and Roli [14] established foundational attack categories. NIST AI 100-2e2025 [17] provides the authoritative federal taxonomy of evasion, poisoning, privacy, and integrity attacks. OWASP Top 10 for LLM Applications [45] catalogs inference-time threats, including prompt injection. MITRE ATLAS [31] maps attack techniques to the ATT&CK framework. Goldblum et al. [15] survey data poisoning and backdoor attacks. Carlini et al. [16] demonstrate training data extraction from large language models. Kumar et al. [35] document industry perspectives on adversarial ML challenges. *Gap:* These taxonomies characterize threats but do not prescribe architectural controls or specify how organizations should produce audit evidence demonstrating threat mitigation. They answer, "what can go wrong" but not "how do we verify it didn't," or "what evidence do we show auditors?" The gap between threat awareness and operational assurance remains unaddressed.

Secure MLOps and AI Governance. Practitioner literature addresses operational security for ML systems. Sculley et al. [6] identified technical debt in ML systems. Amershi et al. [23] documented software engineering practices for ML. Paleyes et al. [24] surveyed deployment challenges. The MLSecOps community has produced guidance on model signing, supply chain security, and CI/CD hardening [28,54]. SLSA [58] and Sigstore [59] provide supply chain attestation frameworks applicable to ML artifacts. CycloneDX [37] and SPDX [36] now support ML-BOM and AI profiles for component inventory. *Gap:* These practices provide point solutions (signing, scanning, logging) but lack integration into a comprehensive Zero Trust architecture with defined trust layers, policy enforcement points, and evidence package specifications. Organizations implementing "secure MLOps" produce artifacts in isolation without systematic mapping to compliance frameworks or auditable traceability from threats to controls to evidence. The result is security activity without demonstrable assurance.

Positioning This Work. Table 1 summarizes the gap analysis. The present work addresses the identified gaps by: (a) decomposing AI systems into four trust layers (Data, Model Supply Chain, Pipeline, Inference) as first-class Zero Trust subjects; (b) specifying Policy Enforcement Points and evidence artifacts for each layer; (c) defining a Cryptographic Bill of Materials (CBOM) extending SBOM/ML-BOM concepts; (d) providing compliance crosswalks to NIST SP 800-207, DoD ZT Strategy, NIST AI RMF, and ISO/IEC 42001; and (e) producing an evidence package template that enables auditable traceability from threats to controls to artifacts. This integration of Zero Trust principles, AI-specific trust decomposition, and evidence-oriented assurance distinguishes the framework from prior work in each category.

Table 1. Gap Analysis: Related Work Categories. Column definitions: **Category** = classification of prior work; **Representative Works** = reference numbers for exemplary publications; **AI Trust Decomposition** = whether the work decomposes AI systems into distinct trust subjects (models, data, pipelines, inference); **Evidence Specification** = whether the work defines specific audit artifacts per control; **Compliance Mapping** = whether the work provides crosswalks to governance frameworks. Source: Author synthesis.

Category	Representative Works	AI Trust Decomp.	Evidence Specif.	Compliance Mapping
ZTA Guidance (Enterprise IT)	[7,8,10,11,21,22]	No	Generic	Partial
AI Threat Taxonomies	[14–17,30,31,35,45]	Implicit	No	No
Secure MLOps	[6,23,24,28,36,37,58,59]	Partial	Point solutions	No
This Work	—	**Yes (4 layers)**	**Yes (per layer)**	**Yes (4 frameworks)**

 doi:10.20944/preprints202602.0085.v1

Note: "AI Trust Decomposition" indicates whether the work treats AI assets (models, data, pipelines) as distinct trust subjects. "Evidence Specification" indicates whether the work defines auditable artifacts. "Compliance Mapping" indicates a crosswalk to federal/international frameworks. Source: Author analysis.

2.6. Methodology: Constructing the Threat-to-Control Mapping

This section describes the systematic method used to construct the threat model, Zero Trust control mapping, and evidence artifact framework presented in this paper. The approach ensures traceability and supports independent verification of the architectural recommendations.

Source Selection Rationale. The taxonomies and guidance documents used in this framework were selected based on three criteria: (1) **Authority**—sources are published by recognized standards bodies (NIST, ISO), government agencies with security mandates (DoD, NSA, CISA), or established industry consortia (OWASP, MITRE); (2) **Adoption**—sources are widely referenced in federal acquisition requirements, compliance frameworks, or practitioner communities (e.g., NIST SP 800-207 is mandated by OMB M-22-09; OWASP Top 10 is referenced in numerous procurement RFPs); (3) **Relevance**—sources directly address either Zero Trust architecture or AI/ML security threats (not general cybersecurity guidance without AI-specific applicability). This selection approach is not a systematic literature review (PRISMA or equivalent); it is a purposive selection of authoritative sources intended to ground architectural recommendations in established federal guidance and industry-accepted threat taxonomies. The selection is inherently scoped to U.S. federal and international standards contexts; practitioners in other jurisdictions should supplement with locally authoritative sources. Table 1 documents the gap analysis across selected source categories.

Threat Selection. Threats were identified through systematic review of authoritative taxonomies: NIST AI 100-2e2025 (Adversarial Machine Learning taxonomy) [17] provided the primary classification of evasion, poisoning, privacy, and integrity attacks; OWASP Top 10 for LLM Applications [45] contributed inference-time threats including prompt injection; MITRE ATLAS provided attack technique mappings; and traditional adversary modeling literature informed the adversary class categorization (external, insider, supply chain, nation-state). Threats were included if they: (a) target AI-specific assets (models, training data, inference pipelines) rather than general IT infrastructure, and (b) have documented instances or proof-of-concept demonstrations in peer-reviewed literature or authoritative threat reports.

Control Mapping Rules. Zero Trust controls were mapped to threats using the following criteria: (1) each threat must map to at least one control from NIST SP 800-207 tenets or DoD Zero Trust pillar activities; (2) controls must be implementable with current or near-term technology (no speculative capabilities); (3) mappings must specify the enforcement point (PEP location) and policy decision inputs; (4) where multiple controls address the same threat, all applicable controls are listed with primary/secondary designation. The mapping was validated by cross-referencing against the NSA Zero Trust Implementation Guidelines activity descriptions [56] and CISA Zero Trust Maturity Model capability definitions [21].

Evidence Artifact Definition. An evidence artifact is defined as a machine-readable or human-auditable record that demonstrates a security control was applied at a specific point in time. Artifacts must satisfy three properties: (1) *Authenticity*—the artifact must be cryptographically signed or otherwise attributable to a verified source; (2) *Integrity*—the artifact must be tamper-evident, typically through inclusion in an append-only log or blockchain-anchored timestamp; (3) *Relevance*—the artifact must directly demonstrate the control it purports to evidence. Example artifacts include: signed model hashes (Sigstore/cosign attestations), SLSA provenance documents, CBOM/SBOM manifests in SPDX or CycloneDX format, policy decision logs from PDP/PEP infrastructure, and TEE attestation quotes. Artifact generation is assumed to occur within the CI/CD pipeline or runtime environment; verification occurs at each trust boundary crossing.

Framework Crosswalk Construction. The compliance crosswalk (Table 3) was constructed by: (1) extracting control objectives from each framework (NIST SP 800-207 tenets, DoD ZT pillars, NIST AI RMF functions, ISO/IEC 42001 clauses); (2) mapping each trust layer's controls to the framework

 doi:10.20944/preprints202602.0085.v1

objectives they satisfy; (3) identifying representative evidence artifacts that would demonstrate compliance with each mapping. The crosswalk is intended as a planning tool for compliance alignment, not as a certification checklist; actual audit acceptance depends on the specific authority, scope, and operational environment.

Validation Approach. The mapping was validated through the following steps: **(V1) Internal Consistency Check**—each threat in Table 2 was verified to map to at least one control, and each control was verified to map to at least one threat, ensuring no orphaned entries; **(V2) Source Traceability**—every threat category was traced to a specific section or technique ID in its source taxonomy (NIST AI 100-2e2025 section numbers, OWASP item numbers, MITRE ATLAS technique IDs), and every control mapping was traced to a specific NIST SP 800-207 tenet (T1–T7) or DoD ZT activity identifier; **(V3) Implementability Review**—each proposed control was assessed for availability of supporting technology (e.g., Sigstore for signing, OPA/Cedar for policy evaluation, CycloneDX for CBOM), with controls requiring non-existent tooling flagged as "emerging" in Section 8.2; **(V4) Evidence Producibility Check**—for each evidence artifact type, at least one existing tool or standard was identified that could generate the artifact (e.g., cosign for signatures, SLSA for provenance, in-toto for attestations); **(V5) Expert Review**—the mapping was reviewed against practitioner guidance including the NSA Zero Trust Implementation Guidelines [56], CISA Zero Trust Maturity Model [21], and DoD Zero Trust Reference Architecture [11] to ensure consistency with current federal implementation direction. This validation approach is reproducible: independent researchers can trace each mapping to its source documents and verify implementability against current tooling. Limitations of this approach include: no empirical testing of control efficacy, no red-team validation of threat coverage completeness, and reliance on published taxonomies that may lag emerging attack techniques. These limitations are explicitly acknowledged in Section 8.2.

Traceability Workflow (Algorithmic Summary). The following procedure summarizes the threat-to-evidence traceability that distinguishes this framework from generic Zero Trust guidance:

1. IDENTIFY threat T from taxonomy (NIST AI 100-2, OWASP LLM Top 10, MITRE ATLAS)
2. SELECT control C from ZT principles (NIST SP 800-207 tenets T1–T7)
3. LOCATE enforcement point PEP (Data | Model | Pipeline | Inference Trust Layer)
4. DEFINE policy inputs for PDP (identity claims, asset attributes, context signals)
5. SPECIFY evidence artifact E (signature, attestation, log entry, CBOM field)
6. MAP to compliance framework (DoD ZT pillar, NIST AI RMF function, ISO 42001 clause)
7. RECORD in evidence package (Appendix template)

OUTPUT: Auditable chain from threat → control → enforcement → evidence → compliance

This workflow is applied systematically in Table 2 (threat-to-control mapping), Table 3 (compliance crosswalk), and the RAG worked example (Section 6.5). The procedure enables organizations to demonstrate not only that controls exist, but that each control addresses a specific threat, is enforced at a defined point, and produces verifiable evidence—the core differentiator from checklist-based compliance approaches.

Note on Visual Artifacts. All figures in this paper (Figures 1–4) are design artifacts derived systematically from the threat model (Section 3), the Zero Trust principle mapping (Section 4), and the compliance crosswalk process (Section 6); they illustrate architectural relationships and conceptual mappings rather than empirically measured system states or incident data, and should be interpreted as prescriptive guidance for implementation planning.

3. Threat Model for AI Systems in Zero Trust Environments

A comprehensive threat model for AI systems operating within Zero Trust environments must account for adversaries with varying capabilities, motivations, and access levels. We categorize adversaries into four primary classes, each presenting distinct risks to AI system integrity, confidentiality, and availability.

 doi:10.20944/preprints202602.0085.v1

3.1. Adversary Classes

External Attackers operate without legitimate access to organizational resources and must breach perimeter defenses or exploit exposed interfaces. Capabilities include exploitation of inference APIs for model extraction, adversarial input crafting, credential theft, and supply chain attacks through compromised dependencies. Zero Trust controls must enforce continuous verification and least-privilege access even after initial authentication.

Insider Threats originate from individuals with legitimate access, including employees, contractors, and partners. These adversaries pose elevated risks due to their knowledge of internal processes and ability to operate within established trust boundaries. Zero Trust architectures must enforce micro-segmentation, continuous behavioral monitoring, and strict access controls, limiting access to the minimum necessary for their roles. Industry breach-cost reporting indicates that incidents involving malicious insiders can be among the most expensive breach categories, reinforcing the need for continuous verification and least-privilege controls in AI pipelines and registries [42,43].

Supply Chain Adversaries target the extended ecosystem of vendors, tools, frameworks, and pre-trained components. As a proxy indicator of ecosystem scale, Hugging Face reported hosting over 1 million ML models in September 2024 [27]; while platform-hosted models do not directly equate to enterprise-deployed models, this figure illustrates the breadth of the model supply chain attack surface. Zero Trust principles require cryptographic verification of all artifacts, continuous integrity validation, and isolation of supply chain components.

Nation-State Actors represent the most capable adversary class with significant resources and long-term strategic objectives. Capabilities include APTs with multi-year dwell times, zero-day exploits, and sophisticated data poisoning campaigns. Zero Trust architectures must assume breach, implement defense-in-depth, and maintain continuous monitoring for sophisticated attack patterns [44,46].

Threat Prioritization Logic. This architecture does not assume equal prioritization across threat classes; organizations should apply risk-tiered implementation based on three factors: (1) **Mission criticality**—AI systems supporting life-safety, national security, or critical infrastructure decisions warrant defense against nation-state and supply chain adversaries; lower-stakes applications may reasonably prioritize external attacker and insider threat controls. (2) **Exposure level**—internet-facing inference endpoints face higher external attacker risk; air-gapped training environments face elevated insider threat risk. (3) **Asset value**—high-value models (large investment, proprietary IP, classified training data) justify deeper controls than commodity deployments. The architecture provides controls for all threat classes; organizations should implement based on their specific risk profile. Table 2 mappings are intentionally comprehensive—not all controls apply to all deployments. A risk-based implementation would start with the threat classes most relevant to the organization's exposure and mission, then expand coverage as maturity increases.

3.2. Attack Surfaces Across the AI Lifecycle

AI systems present attack surfaces at each lifecycle stage, as illustrated in Figure 1. **Training Data Poisoning** injects malicious samples to manipulate model behavior, with research demonstrating that targeted backdoors can succeed with a relatively small number of poisoned samples under specific conditions, motivating provenance and dataset integrity controls as foundational Zero Trust safeguards [32,47]. **Model Tampering** modifies weights, architectures, or configurations. **Inference Manipulation** crafts adversarial inputs, including prompt injection for LLMs [45]. **Unauthorized Model Access** attempts extraction or theft. **Compromised Pipelines** target CI/CD infrastructure, with multiple high-profile software supply-chain compromises in 2024–2025—including the xz-utils backdoor incident—impacting CI/CD and dependency ecosystems commonly used by AI/ML projects [48,53].

 doi:10.20944/preprints202602.0085.v1

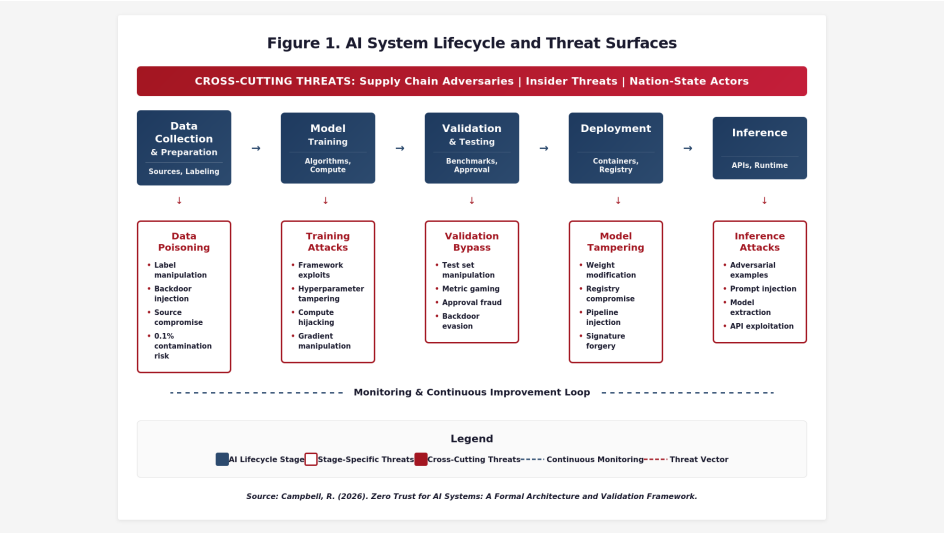

Figure 1. AI System Lifecycle and Threat Surfaces. Note: Any numeric labels (e.g., percentages) or plotted positions shown in the figure are illustrative, qualitative examples derived from the threat model in Section 3 and are not measured incident rates or statistically estimated probabilities unless an explicit empirical source is cited in the caption or body text. Source: Author.

3.3. Zero Trust-Relevant Failure Modes

Traditional architectures fail to protect AI systems due to: **Implicit trust in model artifacts** without continuous verification; **Unverified data sources** without authentication or provenance tracking; **Static model signing** without continuous validation; and a **Lack of runtime identity** preventing Zero Trust policy enforcement for AI workloads. Each requires specific Zero Trust corrections as detailed in the full threat model. Table 2 summarizes the mapping between threat classes, recommended Zero Trust controls, and evidence artifacts required for validation.

Table 2. Threat Class to Zero Trust Control Mapping. Column definitions: **Threat Class** = adversary category or attack type; **Taxonomy IDs** = authoritative source identifiers per MITRE ATLAS [31] and OWASP LLM Top 10 2025 [45]; **Primary PEP** = trust layer where enforcement occurs (Policy Enforcement Point = logical control point where access decisions are enforced, associated with trust layer boundaries per NIST SP 800-207 [7]); **Zero Trust Controls** = recommended security mechanisms; **Evidence Artifacts** = audit documentation produced.

Threat Class	Taxonomy IDs	Primary PEP	Zero Trust Controls	Evidence Artifacts
External Attackers	AML.T0040; OWASP LLM01, LLM10	Inference	Inference PEP, API auth, rate limiting, anomaly detection	Auth logs, rate metrics, anomaly alerts
Insider Threats	AML.T0035; AML.T0007	All layers	Micro-segmentation, least privilege, behavioral monitoring, ABAC	Access logs, behavioral baselines, privilege reviews
Supply Chain Adversaries	AML.T0010; OWASP LLM03	Model	Model signing, CBOM validation, registry PEP, artifact isolation	Signature attestations, CBOM manifests, provenance chains
Nation-State Actors	AML.T0000; AML.T0010	All layers	Defense-in-depth, TEE attestation, continuous integrity, SOC integration	TEE attestations, integrity logs, threat intel correlation
Data Poisoning	AML.T0020; OWASP LLM04	Data	Data lineage, ingestion PEP, source auth, integrity hashing	Provenance chains, hash logs, source attestations
Model Tampering	AML.T0018; OWASP LLM04	Model + Inference	Runtime integrity monitoring, model identity binding, HSM signing	Runtime logs, signature attestations, HSM trails
Prompt Injection	AML.T0051; OWASP LLM01	Inference	Input validation, output filtering, prompt sanitization, tool-call auth	Validation logs, injection alerts, tool auth logs

 doi:10.20944/preprints202602.0085.v1

ATLAS Technique Reference: AML.T0000 = Search Open Technical Databases (reconnaissance); AML.T0007 = Discover ML Artifacts; AML.T0010 = ML Supply Chain Compromise; AML.T0018 = Manipulate AI Model (backdoor/modify); AML.T0020 = Poison Training Data; AML.T0035 = ML Artifact Collection; AML.T0040 = ML Model Inference API Access; AML.T0051 = LLM Prompt Injection. **OWASP LLM 2025 Reference:** LLM01 = Prompt Injection; LLM03 = Supply Chain; LLM04 = Data and Model Poisoning; LLM10 = Unbounded Consumption. Evidence artifact definitions: Attestation = cryptographically signed statement; Provenance chain = linked attestations tracing origin; Manifest = SBOM/CBOM/ML-BOM; Log = timestamped event record. Source: Author synthesis from [31,45].

4. Zero Trust Principles Applied to AI Systems

This section maps the five core Zero Trust principles to AI-specific components, establishing the conceptual foundation for the architecture presented in Section 5. Figure 2 illustrates how these principles apply across data, models, pipelines, and inference services.

Figure 2. Zero Trust Principles Mapped to AI System Components

	Data	Model	Pipeline	Inference
Identity *"Who/What is this?"*	**Dataset Identity** • Provenance chains • Source authentication • Lineage tracking • Hash-based IDs	**Model Identity** • Cryptographic binding • Weight hashes • Architecture fingerprint • CBOM linkage	**Pipeline Identity** • SPIFFE/SPIRE • Workload attestation • Service accounts • mTLS certificates	**Environment Identity** • TEE attestation • Hardware root of trust • Runtime verification • Container identity
Continuous Verification *"Is it still valid?"*	**Data Integrity** • Ingestion validation • Transform verification • Drift detection • Quality monitoring	**Model Integrity** • Load-time verification • Runtime monitoring • In-memory checks • Signature re-validation	**Pipeline Verification** • Stage signatures • Artifact validation • Dependency scanning • Config verification	**Inference Monitoring** • Behavioral baselines • Anomaly detection • Output validation • Pattern analysis
Least Privilege *"Minimum access"*	**Data Access Control** • ABAC policies • Role-based restrictions • Project scoping • Classification labels	**Model Access Control** • Read/write separation • Function segregation • Version scoping • Deploy permissions	**Pipeline Privileges** • Ephemeral credentials • Just-in-time access • Scoped permissions • Time-bound tokens	**Inference Privileges** • Container contexts • Service mesh policies • Resource quotas • API scoping
Micro-Segmentation *"Contain breach"*	**Data Segmentation** • Flow constraints • Exfiltration prevention • Zone isolation • Encryption boundaries	**Model Segmentation** • Registry isolation • Stage separation • Promotion gates • Version boundaries	**Pipeline Segmentation** • Environment isolation • Network policies • Build isolation • Secret boundaries	**Inference Segmentation** • Service isolation • Classification tiers • Lateral movement block • TEE boundaries
Policy Enforcement *"Enforce decisions"*	**Data Ingestion PEP** • Source authentication • Integrity validation • Provenance recording • Quality gates	**Model Registry PEP** • Signature requirements • Provenance checks • Authorization gates • CBOM validation	**Deployment PEP** • Environment compliance • Authorization checks • Config validation • Approval workflows	**Inference PEP** • Request authentication • Input validation • Rate limiting • Pattern enforcement

Source: Campbell, R. (2026). Zero Trust for AI Systems: A Formal Architecture and Validation Framework.

Figure 2. Zero Trust Principles Mapped to AI System Components. This figure illustrates the conceptual mapping between Zero Trust principles and AI-specific components; it does not depict measured control efficacy or incident rates unless an explicit empirical source is cited. "Continuous Monitoring" in this context encompasses not only traditional SOC telemetry but also AI-specific signals, including model drift detection and data drift telemetry. Source: Author.

4.1. Identity for AI Components

Zero Trust requires a strong identity as the foundation for all access decisions. For AI systems, identity must extend beyond users and traditional workloads to encompass AI-specific artifacts. **Model Identity** binds cryptographic credentials to verified model state, incorporating hashes of weights, architecture, and provenance. **Dataset Identity** captures provenance, integrity, and lineage. **Pipeline Identity** leverages standards such as SPIFFE for workload identity and attestation [57]. **Environment Identity** attests execution environments through TEE mechanisms.

4.2. Continuous Verification

Zero Trust replaces point-in-time authentication with continuous verification. **Continuous Model Integrity** verifies models not only at deployment but throughout operation, with runtime monitoring detecting unauthorized modifications. **Continuous Data Provenance** validates data at ingestion and propagates through transformations. **Continuous Inference Monitoring** analyzes patterns for extraction attempts or adversarial inputs.

 doi:10.20944/preprints202602.0085.v1

4.3. Least Privilege for AI Workflows

Least privilege restricts access to the minimum necessary for each task. **Training Data Access** applies ABAC policies based on role, project, and classification. **Model Component Access** segregates by function. **Inference Service Privileges** operate with minimal access through container security contexts and service mesh policies.

4.4. Micro-Segmentation for AI Pipelines

Micro-segmentation isolates components to contain compromise. **Environment Segmentation** isolates training, validation, staging, and production with explicit cross-boundary policies. **Data Flow Segmentation** constrains flows to approved paths. **Model Artifact Segmentation** implements registry access controls with promotion requiring approval and verification.

4.5. Policy Enforcement Points for AI

Policy Enforcement Points (PEPs) implement Zero Trust decisions at boundaries. **Data Ingestion PEPs** enforce source authentication and integrity validation. **Training Pipeline PEPs** verify dataset authorizations. **Model Registry PEPs** enforce signature requirements. **Deployment PEPs** validate deployment authorizations. **Inference PEPs** authenticate requesters and enforce rate limits.

5. Proposed Zero Trust Architecture for AI Systems

5.1. Architectural Overview

The proposed architecture organizes Zero Trust controls into four layers aligned with AI system components: (1) **Data Trust Layer**—controls for data provenance, lineage, integrity, and access; (2) **Model Supply Chain Trust Layer**—controls for model artifacts, dependencies, signing, and verification; (3) **Pipeline Trust Layer**—controls for CI/CD, orchestration, training, and deployment; and (4) **Inference Trust Layer**—controls for runtime operation, access control, and monitoring. Cross-cutting capabilities include cryptographic services, identity management, logging, policy engines, and continuous monitoring. Figure 3 depicts the overall architecture and the relationships between these trust layers.

Implementation Mapping. This reference architecture is intentionally technology-agnostic and compatible with multiple deployment archetypes. For *enterprise cloud deployments*, trust layers map to managed services (e.g., model registries as SaaS, inference via serverless endpoints) with PEPs implemented through API gateways and service mesh policies. For *on-premises deployments*, the architecture maps to containerized microservices with hardware security modules for key management and dedicated policy decision points. For *edge and tactical deployments*, trust layers must accommodate Disconnected, Intermittent, Low-bandwidth (DIL) conditions through cached policies, deferred attestation verification, and store-and-forward evidence collection. For *cross-domain deployments*, additional guards and content inspection capabilities augment the Inference Trust Layer PEPs. The architecture does not mandate specific products or platforms; implementers select components that satisfy the control objectives for their operational context.

 doi:10.20944/preprints202602.0085.v1

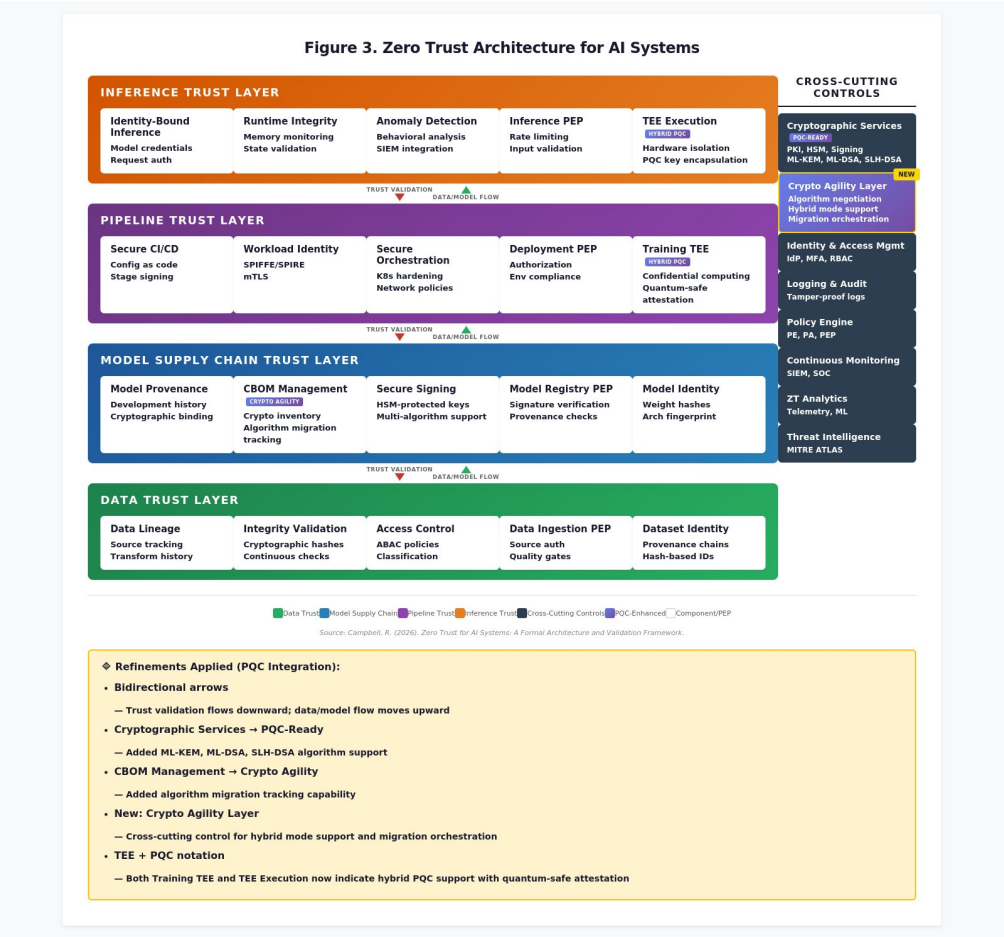

Figure 3. Zero Trust Architecture for AI Systems. The architecture comprises four trust layers corresponding to Sections 5.2–5.5: Data Trust Layer, Model Supply Chain Trust Layer, Pipeline Trust Layer, and Inference Trust Layer, with cross-cutting capabilities for cryptographic services, identity management, and policy enforcement. This figure illustrates the conceptual architecture; component relationships and data flows are illustrative and do not represent measured performance characteristics or incident rates. Source: Author.

5.2. Model Supply Chain Trust Layer

Model Provenance captures complete development history, including source code, training data references, hyperparameters, compute specifications, execution logs, and approval records. Provenance must be cryptographically bound to model artifacts.

Cryptographic Bills of Materials (CBOM) extend Software Bill of Materials (SBOM) concepts to explicitly inventory cryptographic dependencies. A compliant CBOM must include the following minimum required fields: (1) cryptographic algorithms used for signing, encryption, and hashing (algorithm identifiers per IANA registry); (2) key material metadata (key sizes, certificate chain references, expiration dates, revocation status); (3) cryptographic library versions with CVE status; and (4) algorithm migration status for post-quantum readiness (enumeration: not-started, planning, hybrid-deployed, pqc-only). Optional fields include hardware security module bindings and key ceremony documentation. Unlike traditional SBOMs that focus on software packages, CBOMs enable organizations to identify cryptographic technical debt, plan PQC migration, and demonstrate algorithm compliance to auditors. The CycloneDX specification (v1.6+) provides native CBOM support with these field definitions [37].

5.2.1. Bridging CBOM and SBOM: Toward Unified AI Supply Chain Transparency

Executive Order 14028 drove federal software supply-chain security measures and accelerated SBOM guidance through subsequent federal implementation guidance [10,49]. For federal software suppliers, OMB guidance operationalizes these expectations and aligns secure development attes-

 doi:10.20944/preprints202602.0085.v1

tations with NIST's Secure Software Development Framework (SSDF) [49,55]. CISA's 2025 draft "Minimum Elements for a Software Bill of Materials" updates guidance for software transparency [50]. However, traditional SBOMs face limitations for AI: models are not traditional software, data dependencies are first-class, and cryptographic dependencies require explicit tracking.

AI Bill of Materials (AI-BOM) concepts extend SBOM to address these limitations. The G7 has initiated work toward a common framework for SBOM for AI [39]. SPDX provides profiles intended to represent ML model metadata [35,38], and CycloneDX provides a machine-learning bill of materials extension to capture model- and pipeline-relevant metadata within BOM workflows [36,37]. The absence of cryptographic inventory requirements in current Zero Trust implementation guidance—including the NSA Zero Trust Implementation Guidelines, which contain no provisions for tracking certificate algorithm types, key sizes, or migration status [56]—underscores the need for CBOM as an extension layer. CBOM enables integration with existing SBOM toolchains, alignment with emerging AI BOM frameworks, PQC migration tracking, and compliance with anticipated NIST AI RMF requirements. It should be noted that while these standards now support the requisite fields, native integration into mainstream ML frameworks (PyTorch, TensorFlow, JAX) remains an area of active development; this architecture is therefore prescriptive of best practice, recognizing that tooling maturity is progressing toward full implementation support.

AI-BOM Minimum Required Fields. For Zero Trust compliance documentation, an AI-BOM/ML-BOM must include at minimum: (1) **Model identity**—unique identifier, version, cryptographic hash (SHA-256 or stronger) of model weights/architecture; (2) **Training data lineage pointer**—reference to dataset provenance records, source attestations, and data processing pipeline identifiers (not the data itself, but verifiable links to lineage documentation); (3) **Dependency inventory**—ML framework versions (PyTorch, TensorFlow, JAX), supporting libraries, and their SBOM/CVE status; (4) **Signing authority**—identity of signing key, certificate chain, signing timestamp, and algorithm used; (5) **Provenance chain links**—URIs to SLSA/in-toto provenance attestations linking build inputs to outputs. Optional but recommended fields include: training hyperparameters, evaluation metrics, intended use documentation, known limitations, and CBOM cryptographic inventory. The CycloneDX ML-BOM profile [37] and SPDX AI/Dataset profiles [36,38] provide machine-readable schemas for these fields. Organizations should select the profile appropriate to their toolchain while ensuring minimum required fields are populated.

Secure Model Signing provides cryptographic assurance through signing at build with HSM-protected keys, verification at every load, certificate chain validation, revocation checking, and support for multiple algorithms enabling crypto-agility for PQC transition. Given the "harvest now, decrypt later" threat posed by nation-state actors, organizations should implement hybrid signing schemes that combine classical algorithms (e.g., ECDSA, RSA) with NIST-standardized post-quantum algorithms (ML-DSA, SLH-DSA) to ensure long-term artifact integrity. This hybrid approach aligns with NSA CNSA 2.0 guidance and addresses the algorithm agility gap identified in current Enterprise PKI deployments.

5.3. Data Trust Layer

Data Lineage captures the complete transformation history from source to consumption. **Data Integrity Validation** uses cryptographic hashes at rest and in transit, validated at ingestion, transformation, and consumption. For large-scale dynamic datasets typical of enterprise data lakes, static hashing of petabyte-scale datasets presents computational challenges for continuous verification; practical implementations should employ verifiable data structures such as Merkle trees or authenticated data structures that enable efficient incremental verification, combined with snapshotting strategies that capture dataset state at defined checkpoints during training pipeline execution. **Access Control** enforces ABAC policies incorporating classification, role, project, and context.

 doi:10.20944/preprints202602.0085.v1

5.4. Pipeline Trust Layer

Secure CI/CD manages pipeline configurations as code with version control and review. **Trusted Execution Environments** provide hardware-enforced isolation for sensitive training. **Secure Orchestration** hardens platforms with workload identity and network policies. Current DoD Zero Trust implementation guidance for DevSecOps pipelines (Activity 3.2.3 per the DoD Zero Trust Capability Execution Roadmap [12]) focuses on application security testing but does not mandate cryptographic library validation or algorithm compliance checking; the Pipeline Trust Layer addresses this gap by incorporating cryptographic validation gates within CI/CD workflows.

5.5. Inference Trust Layer

Identity-Bound Inference authenticates services using model identity credentials. **Runtime Integrity** monitors unauthorized modifications to the in-memory state. **Anomaly Detection** identifies extraction attempts, adversarial inputs, or compromise patterns. It should be acknowledged that Inference PEPs performing content-level analysis (e.g., adversarial input detection, prompt injection filtering) introduce latency that differs fundamentally from network-layer policy enforcement; such analysis is probabilistic content filtering rather than deterministic packet inspection. In latency-critical contexts such as autonomous systems or time-sensitive decision support, organizations must explicitly balance security depth against operational tempo, potentially employing tiered validation strategies where lightweight checks execute synchronously, and deeper analysis occurs asynchronously with compensating controls.

5.6. Agentic AI Security Considerations

The proliferation of agentic AI systems in 2025–2026—autonomous agents capable of executing tools, invoking APIs, and taking actions on behalf of users—introduces novel security challenges that extend beyond the static model deployment paradigm addressed in Sections 5.2–5.5. Traditional Zero Trust architectures assume a human identity initiates a session and that session's permissions govern all subsequent actions. Agentic AI disrupts this model: an autonomous agent may execute chains of tool calls, spawn sub-agents, or interact with external systems in ways that bypass traditional user-identity boundaries [45,64]. This section outlines how Zero Trust principles must adapt for agentic AI environments.

Agent Identity Verification. Each autonomous agent must possess a cryptographically verifiable identity distinct from (but linked to) the initiating user identity. This agent identity should be bound to: (1) the model artifact hash powering the agent; (2) the tool permissions granted to the agent; (3) the delegation chain from the human principal; and (4) the session context and scope constraints. The PDP must evaluate agent identity alongside user identity when authorizing actions, ensuring that an agent cannot exceed the permissions of its delegating principal (delegation constraint) and that agent permissions can be further scoped based on task context (least-privilege scoping).

Autonomous Action Authorization. Every tool call, API invocation, or external system interaction initiated by an agent must be independently authorized by a Policy Enforcement Point—this is a critical safeguard that prevents an agent from being 'hijacked' via prompt injection into executing dangerous commands or exfiltrating sensitive data. Unlike human-interactive systems, where users can be prompted for confirmation, agentic systems require pre-defined authorization policies that specify: permitted actions by tool category; resource scope limitations; rate limits and budget constraints; and escalation triggers requiring human-in-the-loop approval.

Least-Privilege Scoping for Tool Use. Critically, these PEPs must operate with **dynamic least-privilege scoping**, where an agent's permissions are further restricted based on the specific task context—not merely the agent's baseline capability set. For example, an agent authorized to read files should have its file access scoped to the specific directory relevant to the current task; an agent permitted to make HTTP requests should be constrained to the domains relevant to the current query; an agent with database access should have row/column-level filters applied based on the

 doi:10.20944/preprints202602.0085.v1

data classification of the current conversation. This task-contextual scoping prevents permission creep where an agent's broad baseline capabilities enable unintended access when exploited through adversarial prompts. The Inference Trust Layer PEP must intercept and evaluate each agent action against these context-scoped policies before execution, logging the full action context (agent identity, tool called, parameters, scope constraints applied, authorization decision) for audit. Policy engines should support parameterized permission templates that bind to task context at invocation time, enabling fine-grained control without requiring per-task policy definition.

Inter-Agent Trust and Delegation Chains. Multi-agent systems where agents invoke other agents create delegation chains that must be cryptographically traceable. Each delegation must include: the delegating agent's identity; the scope of delegated authority; expiration constraints; and a non-repudiable delegation token. The PDP must validate the entire delegation chain before authorizing actions, ensuring no agent in the chain exceeds its delegated authority. This is analogous to Kerberos ticket delegation but extended to AI agent contexts with tool-use permissions rather than service access.

Agent Delegation Token Specification. To transform the preceding conceptual guidance into an implementable protocol, we propose a standardized Agent Delegation Token (ADT) format utilizing X.509v3 certificate extensions. This specification is intended to inform future standardization efforts by NIST, IEEE, or other standards bodies. The ADT extends standard X.509 certificates with the following custom extensions:

Extension 1: Model Identity Binding (OID: 1.3.6.1.4.1.XXXXX.1.1). ASN.1 SEQUENCE containing: *modelHash* (OCTET STRING, SHA-256 hash of model weights); *modelVersion* (UTF8String, semantic version identifier); *modelRegistry* (IA5String, URI of authoritative model registry); *attestationRef* (IA5String, URI to SLSA provenance attestation). This extension cryptographically binds the agent identity certificate to a specific, verifiable model artifact.

Extension 2: Tool Permission Scope (OID: 1.3.6.1.4.1.XXXXX.1.2). ASN.1 SEQUENCE OF ToolPermission, where each ToolPermission contains: *toolCategory* (ENUMERATED: FILE_READ, FILE_WRITE, HTTP_REQUEST, DATABASE_QUERY, CODE_EXECUTION, AGENT_DELEGATION, SYSTEM_COMMAND); *resourcePattern* (UTF8String, glob or regex pattern constraining permitted resources, e.g., '/data/public/*' or 'https://*.example.com'); *operationLimit* (INTEGER, maximum invocations per session, 0=unlimited); *requiresEscalation* (BOOLEAN, true if human-in-the-loop approval required). This extension encodes the complete tool-permission matrix for the agent.

Extension 3: Delegation Constraints (OID: 1.3.6.1.4.1.XXXXX.1.3). ASN.1 SEQUENCE containing: *maxDelegationDepth* (INTEGER, maximum sub-agent delegation hops, 0=no delegation permitted); *delegablePermissions* (BIT STRING, bitmap of ToolPermission indices that may be delegated); *delegationPolicy* (ENUMERATED: STRICT_SUBSET, EQUIVALENT, CUSTOM_POLICY); *policyRef* (IA5String OPTIONAL, URI to OPA/Cedar policy document for CUSTOM_POLICY). This extension governs how an agent may delegate authority to sub-agents.

Extension 4: Session Context Binding (OID: 1.3.6.1.4.1.XXXXX.1.4). ASN.1 SEQUENCE containing: *sessionId* (OCTET STRING, unique session identifier); *principalDN* (Name, Distinguished Name of delegating human principal); *taskContext* (UTF8String OPTIONAL, task description hash for audit correlation); *classificationCeiling* (ENUMERATED: UNCLASSIFIED, CUI, CONFIDENTIAL, SECRET, TOP_SECRET, or organization-defined). This extension binds the token to a specific session and classification context.

Token Lifecycle. ADTs should be short-lived (recommended: 1-hour maximum validity) and support both online validation (OCSP) and offline validation (embedded CRL distribution points). Token issuance requires: (1) authentication of the requesting principal; (2) verification that requested permissions do not exceed the principal's authority; (3) model identity verification against the model registry; (4) policy evaluation confirming the permission combination is permitted for the task context. Token revocation must propagate within the session timeout window. The issuing CA should maintain an audit log of all ADT issuances for forensic reconstruction.

 doi:10.20944/preprints202602.0085.v1

Implementation Guidance. Organizations implementing ADTs should: (1) register a Private Enterprise Number (PEN) with IANA for the OID arc; (2) extend existing PKI infrastructure to support ADT issuance and validation; (3) integrate ADT validation into Inference Trust Layer PEPs; (4) implement policy engines (OPA, Cedar, or equivalent) capable of evaluating ToolPermission constraints against runtime tool invocations; (5) log all ADT-governed actions with the full certificate chain for audit. Reference implementations using OpenSSL extensions and OPA policy bundles are recommended for interoperability testing. This specification is proposed as a starting point for community refinement; we encourage feedback from practitioners and standards bodies toward eventual formalization.

Evidence Artifacts for Agentic AI. The evidence package template (Appendix) should be extended for agentic deployments to include: agent identity certificates binding agent instances to model artifacts and tool permissions; delegation chain logs capturing the full principal → agent → sub-agent authorization path; tool invocation logs with pre/post-authorization decisions; and anomaly detection events for agent behavior deviating from expected patterns (e.g., unusual tool sequences, excessive API calls, out-of-scope resource access). These artifacts enable forensic reconstruction of agent actions and support accountability in autonomous decision chains.

5.7. Cross-Cutting Controls

Logging and Auditability captures all security-relevant events with tamper protection. **Continuous Monitoring** aggregates signals for threat detection, integrated with SOC capabilities. **Zero Trust Analytics** processes telemetry to identify patterns and inform policy refinement.

6. Assurance Evidence Framework

6.1. Assurance Objectives

The framework addresses five security objectives: **Integrity** (artifacts remain unmodified from verified state); **Authenticity** (components verified as originating from legitimate sources); **Confidentiality** (sensitive data protected from unauthorized disclosure); **Availability** (systems remain operational despite attacks); and, optionally, **Explainability** (provenance and behavior can be explained for audit). Figure 4 presents a maturity model that organizations can use to assess their Zero Trust AI implementation progress.

6.2. Metrics and Evidence Types

Assurance requires multiple evidence types: **Cryptographic Evidence** (signatures, hashes, certificates, attestations); **Behavioral Evidence** (inference patterns, performance metrics, output distributions); **Provenance Evidence** (lineage records, audit logs, CBOM documentation); and **Runtime Telemetry** (monitoring data, anomaly alerts, integrity results). At minimum, an evidence package intended to support Zero Trust AI compliance audits should include: signed model artifacts with provenance chain, CI/CD pipeline attestations, runtime integrity verification events, and policy decision logs demonstrating continuous verification throughout the model lifecycle. Note that audit acceptance ultimately depends on the specific authority, scope, and operational environment.

To quantify dataset integrity within the assurance framework, we define the **Dataset Authenticity Index (DAI)** as:

$$\text{DAI} = \frac{1}{n} \sum_{i=1}^{n} V(D_i, H_i) \tag{1}$$

where V is a verification function returning 1 if the i-th segment of the dataset D_i matches its signed hash H_i, and n is the total number of segments. A DAI of 1.0 indicates complete dataset integrity; values below 1.0 trigger investigation proportional to the deviation.

Similarly, we define **Model Integrity Score (MIS)**:

$$\text{MIS} = V(W, H^w) \times V(A, H^a) \times V(C, H^c) \tag{2}$$

 doi:10.20944/preprints202602.0085.v1

where W represents model weights, A represents architecture configuration, C represents CBOM manifest, and H represents their respective signed hashes. These metrics provide quantifiable measures for maturity level assessments and continuous compliance monitoring.

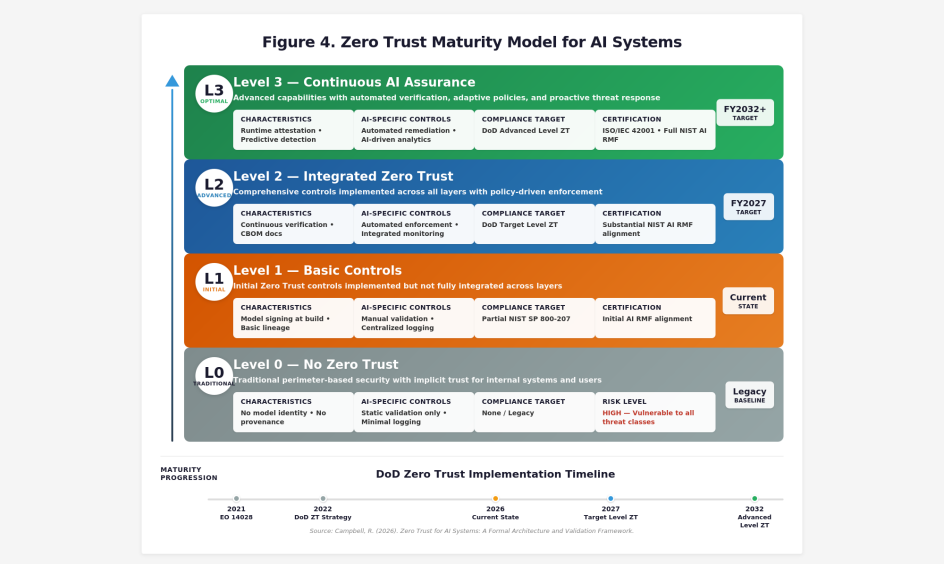

Figure 4. Zero Trust Maturity Model for AI Systems. This original framework extends maturity assessment concepts to incorporate AI lifecycle trust layers, evidence-package requirements, and DoD Zero Trust implementation planning constructs (target-level versus advanced-level activities) [10,11,51]. The model comprises Phase 0 (Discovery and Inventory) as a prerequisite phase, followed by three implementation levels (L1–L3), aligning with the NSA Zero Trust Implementation Guidelines phased approach [56]. Phase 0 addresses the critical visibility requirement acknowledged in NSA ZIG Phase One: organizations cannot implement Zero Trust controls on assets they have not discovered. Evidence requirements by level: Phase 0 (Discovery) requires comprehensive AI asset inventory, trust boundary identification, and assurance debt assessment; L1 (Initial) requires model signing at build time and basic lineage documentation; L2 (Advanced) requires signed models with full CBOM, continuous verification, and policy decision logging; L3 (Optimal) requires runtime attestation events, automated policy enforcement with real-time anomaly detection, predictive threat analytics, and continuous discovery automation. This figure illustrates a conceptual maturity progression; maturity level criteria are prescriptive guidance, not empirically validated benchmarks. Any specific fiscal-year markers shown are policy-aligned planning horizons (not universal deadlines) and should be interpreted as organization- and mission-dependent. Source: Author.

6.3. Compliance Mapping

The architecture supports evidence mapping and alignment assessments across multiple frameworks: **NIST SP 800-207** through implementation of all seven tenets; **NIST AI RMF** through alignment with Govern, Map, Measure, and Manage functions [18]; **DoD Zero Trust guidance** through mapping to all seven pillars with target and advanced activities [11,51]; and **ISO/IEC 42001** through integrated governance, risk management, and continuous improvement [41]. Table 3 provides a detailed crosswalk mapping each trust layer to specific compliance requirements across these frameworks.

Auditability Guidance: To use this crosswalk for audit preparation: (1) identify applicable frameworks for your organization; (2) for each trust layer, locate the cited clause/activity in the source document; (3) verify your evidence artifacts satisfy the source document's requirements; (4) document the mapping rationale for assessor review. The crosswalk is a planning tool, not a substitute for reading source framework text. Source: Author synthesis of cited standards.

 doi:10.20944/preprints202602.0085.v1

Table 3. Framework Crosswalk: Trust Layers to Compliance Requirements. Column definitions: **Trust Layer** = architectural component from Section 5 (Data, Model Supply Chain, Pipeline, Inference); **NIST 800-207 Tenets** = applicable Zero Trust tenets T1–T7 per Section 2 [7]; **DoD ZT Pillars / NSA Activities** = DoD Zero Trust Strategy pillars [10] and Roadmap activity identifiers [12] with NSA ZIG implementation guidance [56]; **NIST AI RMF / ISO 42001** = applicable AI RMF subcategories per Appendix A [18] and ISO 42001 clause numbers [41]; **Mapping Type** = relationship strength (Direct = explicitly required; Partial = domain addressed but not AI-specific; Enabling = supports objectives without explicit requirement); **Example Evidence** = representative audit artifacts.

Trust Layer	NIST 800-207 Tenets	DoD ZT Pillars / NSA Activities	NIST AI RMF / ISO 42001	Mapping Type	Example Evidence
Data Trust	T1, T5, T7	Data Pillar; NSA 5.4.3, 5.4.4	Map 2.3; ISO 42001 §6.1.2, §8.4	Direct (ZT), Partial (AI)	Lineage record, hash manifest
Model Supply Chain	T4, T5, T6	App/Workload Pillar; NSA 1.9.1, 3.1.2	Govern 1.1; ISO 42001 §7.5, §8.2	Partial (ZT), Direct (AI)	Signature, SLSA provenance
Pipeline Trust	T2, T3, T6	Network Pillar; NSA 3.2.3, 4.1.1	Govern 1.2; ISO 42001 §8.1, §9.1	Direct (ZT), Enabling (AI)	CI/CD logs, attestation
Inference Trust	T3, T5, T7	App/Workload Pillar; NSA 3.1.1, 6.1.1	Measure 3.2; ISO 42001 §9.2, §9.3	Direct (ZT), Direct (AI)	PDP logs, drift telemetry

6.4. Maturity Model

Organizations progress through five levels, beginning with a critical discovery phase that acknowledges the reality that many organizations face significant 'assurance debt' due to undocumented AI workloads. This model aligns with the NSA Zero Trust Implementation Guidelines (2026) phased approach [56]:

Phase 0—Discovery and Inventory (prerequisite phase): Before Zero Trust controls can be implemented, organizations must achieve comprehensive visibility into their AI asset landscape. This phase requires: (1) enumeration of all AI systems (models, datasets, pipelines, inference endpoints, agent deployments); (2) identification of trust boundaries and data flows; (3) documentation of current security controls and evidence gaps; (4) classification of assets by mission criticality and data sensitivity; (5) identification of 'Shadow AI'—undocumented models, datasets, or agent deployments operating outside governance frameworks. Organizations cannot progress to Level 1 until they can answer basic questions: What AI systems do we operate? Where do model artifacts reside? What data flows into training pipelines? What tools can our agents invoke? The Discovery Gap and Shadow AI Gap discussed in Section 8.2 represent the primary barriers to Phase 0 completion. NSA ZIG Phase One (Discovery) provides detailed guidance for this foundational activity [56].

Level 1—Basic Controls (initial Zero Trust posture): With asset visibility established, organizations implement foundational controls, including model signing at build time, basic lineage documentation, initial PEP deployment at trust boundaries, and partial NIST SP 800-207 alignment. Evidence production begins, but may be manual and incomplete.

Level 2—Integrated Zero Trust (comprehensive policy-driven controls): Organizations achieve signed models with full CBOM, continuous verification across all trust layers, automated policy decision logging, and DoD Target Level ZT objectives. Evidence packages are automatically generated and audit-ready.

Level 3—Continuous AI Assurance (advanced/optimal posture): Organizations implement runtime attestation events, automated policy enforcement with real-time anomaly detection, predictive threat analytics, continuous discovery automation to detect Shadow AI, DoD Advanced Level ZT objectives, and ISO/IEC 42001 certification readiness where applicable. The agentic operating layer described in Section 8.2 supports Level 3 by maintaining continuous reconciliation between documented architecture and deployed reality.

Assurance Debt Assessment. Organizations should assess their 'assurance debt'—the gap between the current state and the minimum viable evidence requirements defined in Section 6.7—before

 doi:10.20944/preprints202602.0085.v1

initiating implementation. The debt calculation involves: (1) counting AI assets without signed artifacts; (2) identifying data flows without lineage documentation; (3) enumerating inference endpoints without PEP coverage; (4) cataloging agent deployments without delegation token infrastructure. Organizations with high assurance debt should prioritize Phase 0 discovery and instrumentation before attempting Level 1 controls; premature control implementation on an incomplete asset inventory creates a false sense of security while leaving unmanaged assets exposed.

6.5. Worked Example: Evidence Package for a RAG-Based Mission Assistant

To illustrate the assurance evidence framework in practice, consider a Retrieval-Augmented Generation (RAG) assistant deployed in a mission planning environment. The assistant retrieves classified planning documents and generates responses using a fine-tuned LLM. This example demonstrates the minimum evidence package intended to support Zero Trust AI compliance assessments.

Asset Inventory. The RAG system comprises: (1) a fine-tuned LLM (Llama-3-70B-instruct with mission-specific LoRA adapters); (2) an embedding model (e5-large-v2) for document vectorization; (3) a vector database (Milvus) containing ∼50,000 classified planning documents at SECRET//NOFORN; (4) a retrieval index updated weekly; (5) guardrail configurations defining output constraints; (6) inference runtime (vLLM on NVIDIA H100 within a confidential VM using AMD SEV-SNP for CPU isolation and NVIDIA Confidential Computing for GPU memory encryption, with hardware-rooted attestation).

Threat Enumeration. Applicable threats from the taxonomy (Section 3) include: *Data Poisoning*—adversary injects misleading documents into the retrieval corpus; *Model Tampering*—unauthorized modification of LoRA weights or guardrails; *Inference Manipulation*—prompt injection to bypass guardrails or extract classified content; *Model Extraction*—systematic querying to reconstruct model behavior; *Supply Chain Compromise*—malicious dependencies in transformers library or vector database.

Zero Trust Enforcement Points. Five PEPs enforce policy at trust boundaries: (1) *Data Ingestion PEP*—validates document source authentication, classification labels, and integrity hash before corpus insertion; (2) *Model Registry PEP*—requires valid Sigstore signature and CBOM before model artifact promotion; (3) *Deployment PEP*—verifies TEE attestation, container signature, and deployment authorization; (4) *Inference PEP*—authenticates user identity, enforces rate limits, validates input against prompt injection patterns; (5) *Output PEP*—applies classification guards and audit logging before response delivery.

Cryptographic Evidence (What Gets Signed): Model weights signed with HSM-backed keys (Sigstore/cosign attestation [59]); embedding model hash chain; retrieval index fingerprint; adapter/LoRA weights signature; configuration file hash (hyperparameters, guardrails).

CBOM/AI-BOM Contents: Base model identifier and version (e.g., Llama-3-70B-instruct); fine-tuning dataset provenance (classification, source, collection date, hash); retrieval corpus metadata (document count, classification levels, last update); dependency manifest (transformers library version, vector database version, inference runtime); cryptographic algorithm inventory (embedding model, signing algorithms, any PQC components).

Deployment Attestations: SLSA Level 3 provenance for CI/CD pipeline [58]; TEE attestation quote (for CPU workloads: Intel TDX or AMD SEV-SNP; for GPU inference: NVIDIA Confidential Computing attestation); container image signature and SBOM/BOM (e.g., SPDX or CycloneDX) [60,61]; policy-as-code approval record; red-team evaluation summary with test coverage metrics.

Runtime Monitoring Evidence (Retained): Policy decision logs (PDP allow/deny with context); inference request/response metadata (user identity, classification, timestamp—not content); model drift telemetry (embedding distribution shift, output confidence trends); anomaly detection alerts (prompt injection attempts, unusual query patterns); integrity verification events (periodic hash checks, attestation refresh).

Sample Evidence Artifact (Policy Decision Log Entry): A representative log entry demonstrates the evidence structure: *timestamp*: 2026-01-15T14:32:07Z; *request_id*: a7f3b2c1d; *user_dn*: CN=analyst.jones,OU=J2,O=EUCOM; *clearance_verified*: SECRET; *model_hash*: sha256:9f86d08…; *in-*

 doi:10.20944/preprints202602.0085.v1

put_classification: UNCLASSIFIED; *prompt_injection_score*: 0.02; *decision*: ALLOW; *output_classification*: SECRET//NOFORN; *response_time_ms*: 847.

This evidence package enables auditors to verify: (1) the deployed model matches the approved, tested version; (2) all components have traceable provenance; (3) continuous verification occurred throughout operation; and (4) policy decisions can be reconstructed for any access event. The package maps directly to NIST AI RMF Measure function requirements, DoD Zero Trust Visibility/Analytics pillar, and ISO/IEC 42001 documented information requirements.

Evidence Package Verification Checklist. To make the evidence package contribution measurable and independently evaluable, we provide the following verification checklist. An assessor can confirm Zero Trust AI compliance by verifying:

- **Model Identity Binding:** Does the deployed model artifact hash match the hash recorded in the signed CBOM/ML-BOM? Can the signing certificate chain be validated to a trusted root?
- **Provenance Continuity:** Does the SLSA/in-toto provenance attestation link the deployed artifact to a verified build pipeline? Are all intermediate transformations (fine-tuning, quantization, containerization) documented with signed attestations?
- **Policy Decision Log Integrity:** Are PDP/PEP decision logs cryptographically protected (signed or stored in an append-only log)? Do log entries contain sufficient context (timestamp, identity, resource, decision, rationale) for decision reconstruction?
- **Attestation Verification Outcomes:** Were TEE attestation quotes validated against expected measurements? Are runtime integrity check results recorded with pass/fail status and remediation actions for failures?
- **Evidence Completeness:** Does the package include all required artifacts per the applicable tier (Table 4)? Are optional fields documented as N/A with a rationale rather than silently omitted?

This checklist transforms the evidence package from an abstract specification into an evaluable artifact. Assessors can score each item (pass/fail/partial) to produce a quantifiable compliance posture. Organizations failing multiple checklist items should remediate before claiming Zero Trust AI compliance.

6.6. Qualitative Evaluation Approach

While this paper presents a conceptual framework rather than an empirical implementation study, we provide a qualitative evaluation against five criteria that reviewers and practitioners can use to assess the architecture's utility. This rubric-based approach enables structured assessment even in the absence of quantitative benchmarks.

Feasibility. The architecture relies exclusively on existing or near-term technologies: cryptographic signing (Sigstore, cosign), attestation frameworks (SLSA, in-toto), confidential computing (AMD SEV-SNP, Intel TDX, NVIDIA CC), policy engines (OPA, Cedar), and BOM standards (CycloneDX, SPDX). No speculative capabilities are required. *Assessment: High feasibility*—all architectural components map to available tools, though integration effort varies by deployment context.

Implementability. Implementation complexity scales with organizational maturity. Organizations with existing Zero Trust infrastructure (identity providers, policy engines, SIEM) can extend to AI workloads incrementally. Greenfield deployments face a higher initial investment. The four-layer decomposition supports phased adoption: organizations may implement Model Supply Chain Trust first (highest ROI for supply chain risk), then extend to other layers. *Assessment: Moderate implementability*—achievable for mature organizations; challenging for those without foundational Zero Trust capabilities.

Auditability. The framework explicitly specifies evidence artifacts for each control, enabling auditors to verify compliance through artifact inspection rather than architectural review alone. Evidence types (signatures, attestations, logs, manifests) are machine-readable and support automated compliance checking. The compliance crosswalk (Table 3) maps artifacts to specific framework

 doi:10.20944/preprints202602.0085.v1

requirements. *Assessment: High auditability*—the evidence-centric design directly supports audit workflows.

Overhead. The following figures are order-of-magnitude planning assumptions (not empirical measurements) intended to help architects reason about cost centers; actual overhead is implementation-, workload-, and hardware-dependent. Performance overhead concentrates at PEP enforcement points: signing/verification (typically single- to tens-of-milliseconds depending on key type, hardware acceleration, and batching), policy evaluation (typically sub-millisecond to tens-of-milliseconds depending on policy complexity, cache hit rate, and PDP/PEP placement), and logging/-commit (typically sub-millisecond to a few milliseconds depending on serialization, transport, and whether the log is locally buffered versus remotely committed). Inference-time PEPs introduce the most significant latency for real-time applications because content-level inspection (e.g., injection detection and output filtering) is probabilistic and compute-bound rather than deterministic packet inspection. Storage overhead includes artifact retention (signatures, attestations, manifests, and decision logs) and will vary primarily by retention period, sampling strategy, and log verbosity; organizations should model storage as a function of request volume, log fields retained, and cryptographic evidence frequency rather than as a fixed percentage of model size. *Assessment: Moderate overhead*—acceptable for most enterprise workloads; may require optimization for latency-critical applications.

Residual Risk. The architecture reduces but does not eliminate risk. Residual risks include: (1) novel attack vectors not covered by current threat taxonomy; (2) implementation errors in PEP logic or policy rules; (3) compromised root-of-trust (HSM, TEE vulnerabilities); (4) sophisticated adversaries operating within trust boundaries. Defense-in-depth through layered controls and continuous monitoring provides detection capability for residual risks. *Assessment: Reduced but non-zero residual risk* consistent with the Zero Trust philosophy of assumed breach.

6.7. Evidence Artifact Tiers

To address the question of 'minimum viable' versus 'gold standard' evidence, we define three tiers aligned with the maturity model (Section 6):

Table 4. Evidence Artifact Tiers. Column definitions: **Tier** = assurance level designation; **Evidence Artifacts** = specific documentation and cryptographic proofs required at each tier; **Use Case** = deployment scenarios appropriate for each tier; **Maturity Alignment** = correspondence to maturity levels defined in Section 6. The Discovery tier is a prerequisite—organizations must complete Phase 0 asset visibility before implementing controls. Minimum Viable represents the floor for meaningful control assurance; Optimal represents defense-in-depth for high-value targets. Source: Author.

Tier	Evidence Artifacts	Use Case	Maturity
Discovery	AI asset inventory; trust boundary map; data flow documentation; assurance debt assessment; Shadow AI catalog	Prerequisite for all implementations; organizations with unknown AI footprint; post-merger integration	Phase 0
Minimum Viable	Model signature + hash; basic SBOM; access logs; manual provenance documentation	Initial compliance; limited-scope pilots; resource-constrained environments	L1 (Initial)
Recommended	SLSA L2+ provenance; full CBOM; ML-BOM; policy decision logs; automated integrity checks	Production deployments; federal compliance; enterprise AI governance	L2 (Advanced)
Optimal	SLSA L3+; TEE attestation; runtime integrity events; anomaly detection logs; PQC-ready signatures; full audit chain	High-assurance missions; classified environments; adversarial threat contexts	L3 (Optimal)

Minimum Viable Evidence by Trust Layer. For resource-constrained environments, the following represents the floor for meaningful Zero Trust assurance per layer—the minimum evidence required before audit engagement is advisable:

 doi:10.20944/preprints202602.0085.v1

Data Trust Layer (Minimum): (1) Dataset inventory with source attribution; (2) integrity hash at ingestion; (3) access control list documentation. Without these, data provenance claims are unsupportable.

Model Supply Chain Trust (Minimum): (1) Model artifact hash (SHA-256 or stronger); (2) signature from build pipeline (any signing mechanism); (3) basic dependency list. Without these, model identity verification is impossible.

Pipeline Trust Layer (Minimum): (1) CI/CD pipeline logs with timestamps; (2) build environment specification; (3) deployment approval record. Without these, pipeline integrity claims lack evidence.

Inference Trust Layer (Minimum): (1) Authentication logs for inference requests; (2) basic input/output logging (metadata, not content); (3) policy decision records (allow/deny with reason). Without these, continuous verification cannot be demonstrated.

Organizations unable to produce even minimum-tier evidence should prioritize instrumentation before audit engagement. The gap between the current state and minimum viable evidence represents the organization's 'assurance debt'—the foundational work required before Zero Trust AI claims become defensible.

7. Scenario-Based Demonstration

7.1. Scenario Description

Consider an AI-enabled threat detection system deployed within a defense network. The system processes network traffic, endpoint telemetry, and user behavior data to identify malicious activity. The model was trained on classified threat intelligence and proprietary detection algorithms, making it a high-value target. Threat context includes nation-state adversaries seeking to degrade detection or extract intelligence, insider threats attempting exfiltration, and supply chain adversaries targeting dependencies.

7.2. Applying the Architecture

Identity Assignment: The trained model receives a cryptographic identity bound to verified weights and CBOM. Training datasets receive identity with provenance chains. Pipeline components authenticate using SPIFFE-compliant workload identity. Inference services attest through TEE mechanisms.

Continuous Verification: Model signatures are verified at every load. Runtime monitoring detects modifications to in-memory weights. Inference patterns are analyzed for extraction attempts. Data integrity is validated at each pipeline stage.

Micro-Segmentation: Training environment is isolated from operational networks. Model registry requires separate authentication. Inference services are segmented by classification level. Network policies prevent lateral movement.

Policy Enforcement: Data ingestion PEP validates source authentication. Model registry PEP requires valid signatures and provenance. Deployment PEP verifies authorization. Inference PEP authenticates requesters and enforces limits.

7.3. Expected Outcomes

Implementation achieves: **Reduced Attack Surface** through micro-segmentation and least privilege; **Improved Model Integrity** through continuous verification and cryptographic identity; **Enhanced Auditability** through comprehensive provenance and logging; and **Rapid Threat Response** through continuous monitoring and policy-driven enforcement.

8. Discussion

8.1. Benefits

The proposed architecture provides: **Stronger AI Assurance** through continuous verification and cryptographic controls; **Alignment with Federal Mandates** supporting DoD Zero Trust Strategy

 doi:10.20944/preprints202602.0085.v1

and its implementing guidance [51], NIST AI RMF, and anticipated regulations; **Improved Supply Chain Transparency** through CBOM extending SBOM concepts; and **Unified Compliance Framework** reducing burden through integrated mapping.

8.2. Limitations

Framework Classification. This paper presents a **conceptual reference architecture** and assurance framework—not an empirical deployment study. We do not report: measured control efficacy from production deployments; operational incident-rate reduction statistics; performance benchmarking results; or model accuracy/safety evaluation metrics. AI model evaluation (accuracy, fairness, safety alignment) is explicitly out of scope; the framework addresses security architecture and audit evidence, not model quality assessment. The contributions are architectural guidance and evidence specifications; empirical validation through pilots, red-team exercises, and longitudinal deployment studies remains future work. Readers should interpret the framework as prescriptive design guidance informed by authoritative sources, not as empirically validated best practice derived from deployment experience.

Where order-of-magnitude performance or storage impacts are discussed, they are explicitly stated as planning assumptions (implementation- and workload-dependent) rather than empirical claims. Additional limitations include: **Implementation Complexity** requiring significant investment in infrastructure; **Tooling Gaps** as supporting tools are emerging but not mature; **Legacy Integration Challenges** requiring careful migration planning; and **Performance Implications** from continuous verification overhead.

Inference Trust Layer Performance Overhead deserves particular attention. Placing a Policy Enforcement Point in the inference path introduces latency that may be unacceptable for real-time applications where milliseconds matter—autonomous vehicles, high-frequency trading systems, or time-critical defense applications. For such contexts, the Inference Trust Layer may need to operate in "detection mode" (asynchronous logging and alerting) rather than "prevention mode" (synchronous blocking), or employ optimized lightweight policy engines with pre-computed decision caches. Organizations must calibrate enforcement stringency against operational tempo requirements, potentially accepting higher residual risk in exchange for performance in specific use cases.

Discovery Gap. This architecture implicitly assumes organizations possess comprehensive visibility into their AI assets, data flows, and dependencies—an assumption that may not hold for many legacy enterprises. GAO audits have repeatedly identified IT asset management deficiencies across federal agencies, and practitioners have observed that many Zero Trust programs "quietly die" in the discovery phase because organizations cannot answer basic questions about what applications they run, which datasets matter most, or where model artifacts reside [63,65]. The proposed framework's trust layers and evidence requirements presuppose that organizations can enumerate their AI systems, identify trust boundaries, and instrument enforcement points. For organizations lacking this foundational visibility, the architecture may appear aspirational rather than actionable. To address this "truth gap," we recommend that organizations invest in continuous asset discovery and reconciliation capabilities—potentially leveraging agentic AI systems that can autonomously inventory AI workloads, map data lineages, identify unsigned model artifacts, and flag drift between documented architecture and deployed reality. Such discovery automation transforms the framework from a static compliance checklist into a living assurance system that continuously reconciles the environment with the target architecture.

Shadow AI Gap. Beyond the initial discovery challenge lies a more persistent problem: in dynamic enterprise environments, new AI resources—models, datasets, inference endpoints, agent deployments—appear faster than security teams can classify and onboard them into governance frameworks. This "Shadow AI" phenomenon mirrors the Shadow IT challenges of earlier decades, but with higher stakes: an unregistered model serving production traffic, an unapproved dataset integrated into a training pipeline, or an autonomous agent granted tool access outside formal authorization channels. Traditional discovery approaches treat asset inventory as a periodic documentation project; this cadence is fundamentally mismatched to the velocity of modern AI development. We advocate

 doi:10.20944/preprints202602.0085.v1

for an **agentic operating layer**—a continuously running discovery and classification system that leverages operational artifacts (deployment manifests, CI/CD pipeline logs, API gateway traffic, infrastructure-as-code repositories, ticketing systems, runbooks, architecture decision records) to automatically detect new AI assets, propose classifications based on observed behavior and metadata, suggest ownership assignments based on code commit history and organizational structure, and flag assets lacking required evidence artifacts (signatures, CBOMs, provenance attestations). This approach transforms discovery from a point-in-time audit into a continuous reconciliation loop where the agentic layer maintains a living inventory, proposes classifications for human review, and escalates unresolved gaps. The architecture presented in this paper provides the target state against which such an agentic operating layer would reconcile; without continuous discovery automation, the gap between documented architecture and deployed reality will inevitably grow, undermining the assurance objectives the framework is designed to achieve.

Scope Boundaries. To set appropriate expectations, we explicitly state what this work does *not* address:

Out of Scope: (1) **Safety alignment**—this framework does not address AI value alignment, constitutional AI, reinforcement learning from human feedback (RLHF), or behavioral safety constraints; (2) **Fairness and bias mitigation**—we do not provide guidance on algorithmic fairness, demographic parity, disparate impact assessment, or bias detection; (3) **Privacy engineering**—differential privacy, federated learning privacy guarantees, membership inference defenses, and PII protection are not addressed; (4) **Model interpretability**—explainability methods, feature attribution, and transparency mechanisms are outside scope; (5) **Environmental sustainability**—carbon footprint, energy consumption, and compute efficiency are not addressed.

Evidence Support for Adjacent Audits: While these topics are out of scope, the evidence artifacts produced by this framework can *support* audits in these areas without claiming to solve them. Specifically: training data lineage records support fairness audits by documenting dataset composition and sourcing; model provenance chains support safety audits by enabling version tracking of alignment-tuned models; policy decision logs support privacy audits by documenting access patterns; and the AI-BOM format can be extended to capture fairness evaluation results, safety test coverage, and privacy impact assessments. The architecture provides the evidentiary infrastructure upon which specialized audits can build; it does not substitute for domain-specific safety, fairness, or privacy expertise.

Empirical Limitations: (1) We do not provide quantitative evaluation of detection efficacy—no empirical measurements of false positive/negative rates, detection latency, or throughput impact are presented; (2) We do not claim that the proposed controls eliminate adversarial machine learning threats—the architecture reduces attack surface and improves auditability, but determined adversaries with sufficient resources may still succeed; (3) The evidence package specifications are templates intended to guide implementation, not prescriptive checklists—actual implementations will vary by mission constraints, classification regimes, operational tempo requirements, and available tooling; (4) Compliance mappings indicate alignment opportunities, not certification guarantees—audit acceptance depends on the specific authority, assessor, and operational context. This framework provides architectural guidance and evidence specifications; empirical validation of specific implementations remains for future work.

8.3. Future Work

Future directions include: **Post-Quantum Cryptography Integration** as algorithms are standardized, with CBOM supporting migration tracking; **Automated CBOM Generation** to reduce implementation burden; **AI-Driven Zero Trust Analytics,** creating a virtuous cycle where AI secures AI; **Zero Trust for Agentic AI,** as the proliferation of autonomous AI agents in 2025–2026 introduces novel identity and authorization challenges—Section 5 provides initial guidance on agent identity verification, autonomous action authorization, and delegation chains, but empirical validation, standardization of agent identity credential formats, inter-agent trust protocols, and tool-authorization

 doi:10.20944/preprints202602.0085.v1

policy languages remain open areas requiring further research and industry collaboration; **Zero Trust for Tactical and Edge AI Systems,** as the NSA Zero Trust Implementation Guidelines explicitly frame their scope as IT-enterprise environments and note that operational technology, tactical, and weapons systems environments require future contextual adaptation [56]—AI systems deployed in Disconnected, Intermittent, Low-bandwidth (DIL) conditions present unique challenges for continuous verification, real-time policy enforcement, and cryptographic operations that require architectural adaptation; and **Standardization** through industry collaboration and bodies such as NIST or ISO.

9. Conclusions

Artificial intelligence systems present unique security challenges that existing Zero Trust frameworks do not adequately address. This paper has proposed a comprehensive Zero Trust Architecture for AI Systems that adapts foundational Zero Trust principles to the specific components and workflows of AI development and deployment.

Our contributions include a structured threat model identifying adversary classes and attack surfaces, a principled mapping of Zero Trust concepts to AI components, a four-layer architecture with defined controls and verification mechanisms, and an assurance evidence framework supporting compliance alignment with NIST AI RMF, DoD Zero Trust Strategy, and ISO/IEC 42001.

The architecture establishes a foundation for standardized and repeatable Zero Trust implementations for AI systems, producing evidence packages intended to support alignment assessments against NIST AI RMF, DoD Zero Trust guidance, and ISO/IEC 42001 requirements. As AI becomes increasingly central to mission-critical operations across defense, intelligence, and critical infrastructure, the need for rigorous security architectures will only intensify. We call on the security community, standards bodies, and regulatory authorities to advance this work toward formal standardization, enabling secure AI adoption across the U.S. Government, Department of Defense, and Defense Industrial Base.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC — Attribute-Based Access Control; AI-BOM — AI Bill of Materials; CBOM — Cryptographic Bill of Materials; CI/CD — Continuous Integration/Continuous Deployment; CISA — Cybersecurity and Infrastructure Security Agency; DAI — Dataset Authenticity Index; DIL — Disconnected, Intermittent, Low-bandwidth; DoD — Department of Defense; EO — Executive Order; HSM — Hardware Security Module; ICAM — Identity, Credential, and Access Management; ISO — International Organization for Standardization; LLM — Large Language Model; MIS — Model Integrity Score; ML — Machine Learning; ML-BOM — Machine Learning Bill of Materials; NIST — National Institute of Standards and Technology; NSA — National Security Agency; OPA — Open Policy Agent; PDP — Policy Decision Point; PEP — Policy Enforcement Point; PKI — Public Key Infrastructure; PQC — Post-Quantum Cryptography; RAG — Retrieval-Augmented Generation; RBAC — Role-Based Access Control; RMF — Risk Management Framework; SBOM — Software Bill of Materials; SEV-SNP — Secure Encrypted Virtualization–Secure Nested Paging (AMD); SGX — Software Guard Extensions (Intel); SLSA — Supply-chain Levels for Software Artifacts; SOC — Security Operations Center; SPDX — Software Package Data Exchange; TDX — Trust Domain Extensions (Intel); TEE — Trusted Execution Environment; ZIG — Zero Trust Implementation Guidelines; ZT — Zero Trust; ZTA — Zero Trust Architecture.

Supplementary Materials: Reproducibility Artifacts. While this paper presents a conceptual framework rather than executable code, we describe the artifacts that implementations would produce to enable reproducibility and interoperability: (1) *CBOM/AI-BOM schema*—JSON or XML manifests following CycloneDX v1.6 ML-BOM and CBOM profiles, capturing model provenance, dataset lineage, and cryptographic inventory; (2) *Policy decision log format*—structured JSON records containing timestamp, request context, subject identity, resource identifier, policy evaluation result, and cryptographic binding to audit chain; (3) *Attestation record format*—TEE attestation quotes (SGX/TDX/SEV-SNP) and SLSA provenance attestations in standard formats for supply chain verification;

 doi:10.20944/preprints202602.0085.v1

(4) *Model identity certificate*—X.509 certificates binding model hash to signing key, with extensions for CBOM reference and provenance URI. Sample schema definitions and example artifacts aligned with this framework are planned for release in a companion technical report.

Author Contributions: Conceptualization, R.C.; methodology, R.C.; formal analysis, R.C.; writing—original draft preparation, R.C.; writing—review and editing, R.C. The author has read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study did not involve human subjects or animals.

Informed Consent Statement: Not applicable. This study did not involve human subjects.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing does not apply to this article.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Minimum Evidence Package Template

The following template specifies the minimum evidence artifacts for Zero Trust AI compliance assessment. Organizations should adapt field requirements to their specific mission, classification regime, and tooling environment.

Appendix A.1. Machine-Readable Template (Illustrative JSON Skeleton)

To support automation and interoperability, an evidence package can be represented as a structured record with stable keys. The following JSON skeleton is illustrative (field requirements are normative; serialization format is implementation-specific):

Listing 1: Evidence Package JSON Schema (Illustrative)

```
{
  "evidence_package_version": "1.0",
  "system_id": "<unique system identifier>",
  "assessment_date": "<YYYY-MM-DD>",
  "model_identity": {
    "model_id": "<registry identifier>",
    "artifact_digest": "<sha256|sha3 digest>",
    "signing_key_id": "<key identifier>",
    "signature_algorithm": "<e.g., ECDSA-P384 | ML-DSA-65 | SLH-DSA>",
    "signature_value": "<base64 or reference>",
    "provenance_uri": "<SLSA/in-toto URI>",
    "cbom_uri": "<CycloneDX CBOM URI>",
    "sbom_or_mlbom_uri": "<SPDX/CycloneDX ML-BOM URI>"
  },
  "deployment_attestations": {
    "slsa_level": "<0-4>",
    "cicd_pipeline_id": "<pipeline id>",
    "build_timestamp": "<ISO 8601>",
    "container_digest": "<image digest>",
    "tee_attestation": {
      "tee_type": "<none|SGX|TDX|SEV-SNP|NVIDIA-CC>",
      "attestation_quote": "<base64 or reference>",
      "verification_result": "<pass|fail|deferred>"
    }
  },
  "runtime_evidence": {
    "policy_decision_log_uri": "<uri>",
    "integrity_event_log_uri": "<uri>",
    "anomaly_alert_log_uri": "<uri>",
    "retention_days": "<integer>"
  },
```

 doi:10.20944/preprints202602.0085.v1


```
  "compliance_mapping": {
    "nist_ai_rmf_functions": ["Govern", "Map", "Measure", "Manage"],
    "dod_zt_pillars": ["User", "Device", "Network/Environment",
      "Application/Workload", "Data", "Visibility/Analytics",
      "Automation/Orchestration"],
    "nsa_zig_activities": ["<activity␣ids>"],
    "iso_iec_42001_clauses": ["<clause␣ids>"]
  }
```

Appendix A.2. Model Identity and Provenance — REQUIRED FIELDS

- Model artifact hash (SHA-256 or SHA-3): [hash] (REQUIRED)
- Signing key identifier: [key_id] (REQUIRED)
- Signature algorithm: [algorithm_id] (REQUIRED)
- Signature value: [base64] (REQUIRED)
- CBOM reference URI: [uri] (REQUIRED)
- Provenance document URI (SLSA): [uri] (REQUIRED for L2+ maturity)
- *Optional:* Base model identifier (if fine-tuned): [model_id]
- *Optional:* Training dataset reference: [dataset_id with classification]

Appendix A.3. Cryptographic Bill of Materials (CBOM) — REQUIRED FIELDS

- Format: CycloneDX v1.6+ JSON (REQUIRED)
- Signing algorithms in use: [algorithm_id per IANA registry] (REQUIRED)
- Hash algorithms in use: [algorithm_id] (REQUIRED)
- Key sizes: [bits] (REQUIRED)
- Certificate chain references: [uri] (REQUIRED)
- PQC migration status: [not-started | planning | hybrid-deployed | pqc-only] (REQUIRED)
- Cryptographic library versions: [name, version, CVE status] (REQUIRED)
- *Optional:* HSM binding reference, key ceremony documentation

Appendix A.4. Deployment Attestations — REQUIRED FIELDS

- SLSA provenance level: [0-4] (REQUIRED, minimum L2 for Target-level ZT)
- CI/CD pipeline identifier: [pipeline_id] (REQUIRED)
- Build timestamp: [ISO 8601] (REQUIRED)
- Container image digest: [digest] (REQUIRED)
- SBOM reference: [uri] (REQUIRED)
- *Conditional:* TEE attestation type and quote (REQUIRED if confidential computing deployed)
- *Conditional:* Policy-as-code approval record: [reference] (REQUIRED for L3 maturity)

Appendix A.5. Runtime Evidence (Retained Logs) — REQUIRED FIELDS

- Policy decision log location: [uri] (REQUIRED)
- Log retention period: [days, minimum 90 for compliance] (REQUIRED)
- Required fields per PDP entry: {timestamp, request_id, subject_dn, resource_id, action, decision, policy_version} (ALL REQUIRED)
- Inference PEP log location: [uri] (REQUIRED for AI workloads)
- Required fields per inference entry: {timestamp, request_id, user_identity, input_classification, decision, output_classification} (ALL REQUIRED)
- *Optional:* prompt_injection_score, response_time_ms, model_version_hash
- Integrity verification event log: [uri] (REQUIRED for L2+ maturity)
- Anomaly detection alert log: [uri] (REQUIRED for L3 maturity)

 doi:10.20944/preprints202602.0085.v1

Appendix A.6. Compliance Mapping

- NIST AI RMF functions addressed: [Govern | Map | Measure | Manage]
- DoD ZT pillars addressed: [list]
- NSA ZIG activities satisfied: [activity_ids]
- ISO/IEC 42001 clauses addressed: [clause_ids]
- Assessment date: [date]
- Assessor: [identity]
- Next review date: [date]

References

1. National Security Commission on Artificial Intelligence. *Final Report*; NSCAI: Washington, DC, USA, March 2021. Available online: https://www.nscai.gov/wp-content/uploads/2021/03/Full-Report-Digital-1.pdf (accessed on 29 January 2026).
2. Executive Office of the President. Executive Order 14110: Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence; Federal Register Vol. 88, No. 210, pp. 75191–75226, 1 November 2023. Available online: https://www.federalregister.gov/documents/2023/11/01/2023-24283/safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence (accessed on 29 January 2026).
3. U.S. Government Accountability Office. Artificial Intelligence: Status of Developing and Acquiring Capabilities for Weapon Systems. GAO-22-104765, 2022. Available online: https://www.gao.gov/assets/gao-22-104765.pdf (accessed on 29 January 2026).
4. Stanford Institute for Human-Centered Artificial Intelligence (HAI). *The AI Index 2025 Annual Report*; Stanford University: Stanford, CA, USA, 2025. Available online: https://hai.stanford.edu/ai-index/2025-ai-index-report (accessed on 29 January 2026).
5. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
6. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; Dennison, D. Hidden Technical Debt in Machine Learning Systems. In Proceedings of the Advances in Neural Information Processing Systems 28 (NeurIPS 2015), Montreal, QC, Canada, 7–12 December 2015; pp. 2503–2511.
7. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; NIST Special Publication 800-207; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. https://doi.org/10.6028/NIST.SP.800-207
8. Kindervag, J. *Build Security Into Your Network's DNA: The Zero Trust Network Architecture*; Forrester Research: Cambridge, MA, USA, 2010.
9. Executive Office of the President. Executive Order 14028: Improving the Nation's Cybersecurity; Federal Register Vol. 86, No. 93, pp. 26633–26647, 17 May 2021.
10. Department of Defense. *DoD Zero Trust Strategy*; DoD CIO: Washington, DC, USA, November 2022. Available online: https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf (accessed on 29 January 2026).
11. Department of Defense Chief Information Officer. *Department of Defense Zero Trust Reference Architecture, Version 2.0*; DoD CIO: Washington, DC, USA, September 2022.
12. Department of Defense. *Zero Trust Capability Execution Roadmap, Version 1.1*; DoD CIO: Washington, DC, USA, November 2024.
13. Chandramouli, R. *A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments*. NIST SP 800-207A; National Institute of Standards and Technology: Gaithersburg, MD, USA, April 2023. https://doi.org/10.6028/NIST.SP.800-207A
14. Biggio, B.; Roli, F. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Pattern Recognit.* **2018**, *84*, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023
15. Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; Goldstein, T. Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 1563–1580.
16. Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú.; et al. Extracting Training Data from Large Language Models. In Proceedings of the 30th USENIX Security Symposium, Virtual, 11–13 August 2021; pp. 2633–2650.

 doi:10.20944/preprints202602.0085.v1

17. Vassilev, A.; Oprea, A.; Fordyce, A.; Anderson, H. *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*. NIST AI 100-2e2025; NIST: Gaithersburg, MD, USA, March 2025. https://doi.org/10.6028/NIST.AI.100-2e2025
18. National Institute of Standards and Technology. *AI Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1; NIST: Gaithersburg, MD, USA, January 2023. https://doi.org/10.6028/NIST.AI.100-1
19. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1; NIST: Gaithersburg, MD, USA, July 2024.
20. Cunningham, C. *Zero Trust Architecture*. O'Reilly Media, 2021.
21. Cybersecurity and Infrastructure Security Agency. *Zero Trust Maturity Model, Version 2.0*; CISA: Washington, DC, USA, April 2023. Available online: https://www.cisa.gov/zero-trust-maturity-model (accessed on 29 January 2026).
22. National Institute of Standards and Technology (NIST). *Implementing a Zero Trust Architecture*; NIST Special Publication (SP) 1800-35; NIST: Gaithersburg, MD, USA, 2025.
23. Amershi, S.; Begel, A.; Bird, C.; DeLine, R.; Gall, H.; Kamar, E.; Nagappan, N.; Nushi, B.; Zimmermann, T. Software Engineering for Machine Learning: A Case Study. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 25–31 May 2019; pp. 291–300.
24. Paleyes, A.; Urma, R.-G.; Lawrence, N.D. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Comput. Surv.* **2022**, *55*, 114.
25. National Institute of Standards and Technology (NIST). TrojAI: Artificial Intelligence Security (model trojans and ML supply-chain risk). NIST, 2024–2025. Available online: https://pages.nist.gov/trojai/docs/about.html (accessed on 29 January 2026).
26. Australian Cyber Security Centre. Artificial Intelligence and Machine Learning Pose New Cyber Security Risks to Supply Chains; ACSC: Canberra, Australia, 2025.
27. Hugging Face. Hugging Face reaches 1 million models (platform milestone announcement). 2024. Available online: https://huggingface.co/posts/fdaudens/300554611911292 (accessed on 29 January 2026).
28. Jiang, W.; Synovic, S.; Sethi, R.; Indarapu, A.; Hyatt, D.; Schorlemmer, M.; Thiruvathukal, G.K. An Empirical Study of Artifacts and Security Risks in the Pre-trained Model Supply Chain. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2022.
29. Australian Cyber Security Centre. Engaging with Artificial Intelligence; ACSC: Canberra, Australia, 2024.
30. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
31. MITRE Corporation. ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). MITRE, 2025. Available online: https://atlas.mitre.org/ (accessed on 31 January 2026).
32. Oprea, A.; Vassilev, A. Poisoning Attacks Against Machine Learning. National Institute of Standards and Technology (NIST), 2024.
33. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
34. Cohen, J.; Rosenfeld, E.; Kolter, Z. Certified Adversarial Robustness via Randomized Smoothing. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 1310–1320.
35. Kumar, R.S.S.; Nyström, M.; Lambert, J.; Marshall, A.; Goertzel, M.; Comber, A.; Swann, M.; Xia, S. Adversarial Machine Learning—Industry Perspectives. In Proceedings of the 2020 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 21 May 2020; pp. 69–75.
36. Linux Foundation. SPDX Specification v3.0.1 (including AI and Dataset profiles). SPDX Workgroup, 2024. Available online: https://spdx.dev/wp-content/uploads/sites/31/2024/12/SPDX-3.0.1-1.pdf (accessed on 29 January 2026).
37. OWASP Foundation. CycloneDX Bill of Materials Standard, Version 1.6. OWASP CycloneDX Project, 2024. Available online: https://cyclonedx.org/docs/1.6/json/ (accessed on 29 January 2026).
38. SPDX Workgroup. SPDX Specification v3.0.1—AI Profile Compliance Point (Conformance). SPDX, 2024.
39. CycloneDX. Specification Overview (CycloneDX supports describing machine learning models as components). CycloneDX, 2025.
40. German Federal Office for Information Security (BSI). A Shared G7 Vision on Software Bill of Materials for AI; BSI: Bonn, Germany, 2025.

 doi:10.20944/preprints202602.0085.v1

41. International Organization for Standardization. ISO/IEC 42001:2023 Information Technology—Artificial Intelligence—Management System; ISO: Geneva, Switzerland, 2023.
42. Verizon. *2024 Data Breach Investigations Report*; Verizon Enterprise: New York, NY, USA, 2024.
43. IBM Security. *Cost of a Data Breach Report 2024*; IBM: Armonk, NY, USA, 2024.
44. European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2024. ENISA, 2024.
45. OWASP Foundation. OWASP Top 10 for Large Language Model Applications, Version 2025. OWASP LLM AI Security Project, 2025. Available online: https://genai.owasp.org/llm-top-10/ (accessed on 29 January 2026).
46. OpenAI. Disrupting malicious uses of AI: October 2025; OpenAI Threat Intelligence Report, October 2025.
47. Chen, X.; Liu, C.; Li, B.; Lu, K.; Song, D. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. arXiv 2017, arXiv:1712.05526.
48. CERT Coordination Center. VU#534320: XZ Utils data compression library contains a backdoor affecting downstream software supply chains. CERT/CC, 2024–2025.
49. National Telecommunications and Information Administration (NTIA). The Minimum Elements for a Software Bill of Materials (SBOM). NTIA, 2021.
50. Cybersecurity and Infrastructure Security Agency (CISA). 2025 Minimum Elements for a Software Bill of Materials (SBOM): Draft for Comment. CISA, 22 August 2025.
51. Department of Defense Chief Information Officer (DoD CIO). Directive-Type Memorandum (DTM) 25-003, Implementing the DoD Zero Trust Strategy; DoD: Washington, DC, USA, 2025; Effective 17 July 2025; Expires 17 July 2026.
52. Executive Office of the President. Executive Order 14148: Initial Rescissions of Harmful Executive Orders and Actions; Federal Register, 2025.
53. Sonatype. 10th Annual State of the Software Supply Chain Report. Sonatype, 2024.
54. Kellas, A.D.; Christou, N.; Jiang, W.; Li, P.; Simon, L.; David, Y.; Kemerlis, V.P.; Davis, J.C.; Yang, J. PickleBall: Secure Deserialization of Pickle-based Machine Learning Models. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2025.
55. National Institute of Standards and Technology. Secure Software Development Framework (SSDF) Version 1.1. NIST Special Publication 800-218, 2022.
56. National Security Agency (NSA) Cybersecurity Directorate. Zero Trust Implementation Guidelines (document set: Primer; Discovery; Phase One; Phase Two); NSA: Fort Meade, MD, USA, January 2026.
57. SPIFFE Project. SPIFFE Specifications (Secure Production Identity Framework for Everyone): Standards and Rendered Specification Documents. Available online: https://spiffe.io/docs/latest/spiffe-specs/ (accessed on 29 January 2026).
58. SLSA. SLSA Specification v1.0; Supply-chain Levels for Software Artifacts. Available online: https://slsa.dev/spec/v1.0/ (accessed on 31 January 2026).
59. Sigstore. cosign Documentation; Sigstore Project. Available online: https://docs.sigstore.dev/cosign/ (accessed on 31 January 2026).
60. SPDX Workgroup. SPDX Specifications (ISO/IEC 5962:2021) and Current Versions. Available online: https://spdx.dev/use/specifications/ (accessed on 31 January 2026).
61. OWASP Foundation. CycloneDX Specification Overview (ECMA-424). Available online: https://cyclonedx.org/specification/overview/ (accessed on 31 January 2026).
62. European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). OJ L, 2024/1689, 12.7.2024.
63. Warshavski, D. Zero Trust Discovery Challenges: Why Many Programs Stall Before Implementation. CSO Online, January 2026.
64. Anthropic. Building Effective Agents: Security Considerations for Autonomous AI Systems. Anthropic Research, December 2024.
65. U.S. Government Accountability Office. IT Asset Management: Agencies Need to Improve Implementation of Leading Practices. GAO-25-106348, December 2024.