

High-Assurance Framework for Safe and Secure Artificial Intelligence

Core Pillars and Confined Architecture Requirements

<https://glassboxai.dev/research/high-assurance-ai-safety-security-confined-architecture/>

Executive Summary and Scope

As Artificial Intelligence (AI) systems and Large Language Models (LLMs) transition from exploratory tools to components operating in critical environments, conventional software quality measures are no longer sufficient. Ensuring an AI system is safe (preventing unintentional harm, catastrophic actions, and systemic failure) and secure (resilient against adversarial manipulation, exfiltration, and unauthorized privilege escalation) requires a two-tiered framework:

1. Foundational Trustworthy AI & Governance Requirements: Derived from international consensus standards, risk management protocols, and statutory regulations.
2. Confined, Deterministic High-Assurance Architecture: Derived from mission-critical engineering disciplines—such as avionics safety certification, air-gapped critical infrastructure protection, and formal AI containment theory.

CONTENTS

Part I — Foundational Trustworthy AI & Cybersecurity Pillars

Part II — Confined and Deterministic High-Assurance Architecture

3. Regulatory and Standards Concordance

Part I: Foundational Trustworthy AI & Cybersecurity Pillars

Pillar 1: Technical Robustness and Cybersecurity

Adversarial Robustness: Systems must enforce defenses against prompt injection (direct and indirect), jailbreaking, evasion techniques, model evasion, and training data poisoning.

Supply Chain Security: Organizations must establish formal provenance tracking for base models, foundation checkpoints, weights, third-party libraries, and serialized dependencies (such as pickle files), mitigating unauthorized modifications and vulnerable dependencies.

Input and Output Sanitization: Input prompts and model-generated responses must never be implicitly trusted. Environments must implement runtime content filtering, structural validation, and strict sandboxing for automated API calls and code execution environments.

Safe State and Graceful Degradation (Fail-Safe): When encountering extreme uncertainty, unexpected anomalies, or active exploitation, the system must transition into a predictable, safe state without cascading failures

or data leakage.

Pillar 2: Data Governance and Privacy

Data Quality and Integrity: Data corpora used for pre-training, fine-tuning, retrieval-augmented generation (RAG), and evaluation must be statistically representative, validated, and free of known systematic corruptions.

Privacy by Design and Exfiltration Prevention: Architecture must prevent sensitive information and Personally Identifiable Information (PII) disclosures caused by inference extraction, membership inference, or model inversion attacks.

Lineage, Consent, and Licensing: Provenance, legal basis of collection, intellectual property licenses, and user consent must be recorded and auditable across all dataset pipelines.

Pillar 3: Transparency, Explainability, and Documentation

Interaction Disclosure: Systems must inform human users that they are interacting with an artificial system and convey its design boundaries and known performance limitations.

Reasoning and Interpretability: In high-consequence domains (e.g., healthcare, criminal justice, financial assessment), decision pathways, confidence metrics, and contextual rationales must be explainable in human-understandable terms.

Standardized System Cards: Organizations must maintain and publish detailed model documentation (e.g., Model Cards, System Cards) covering training methodologies, known edge-case failures, intended use cases, and out-of-scope deployments.

Pillar 4: Human Oversight and Governance

Continuous Intervention Capabilities (HITL / HOTL): Qualified personnel must retain the capability to supervise operations, adjust intermediate parameters, override erroneous inferences, or abort operations in real time.

Mitigation of Automation Bias: Interfaces must be structured to prevent human operators from reflexively trusting machine inferences without verification.

Deterministic Emergency Stop (Kill Switch): Hardware- and software-level kill mechanisms must be capable of severing execution threads or isolating the model without generating cascading operational hazards.

Pillar 5: Fairness, Bias Management, and Non-Discrimination

Systematic Bias Auditing: Continuous evaluation across protected classes to detect and mitigate algorithmic biases that lead to unlawful or disparate impacts.

Metric-Driven Parity: Systems must undergo pre-deployment and continuous testing against formal fairness and parity metrics relevant to the application domain.

Pillar 6: Continuous Security Testing and Operational Guardrails

Adversarial Red Teaming: Independent red teams must carry out adversarial evaluations to identify boundary degradation, prompt bypasses, and logical flaws prior to deployment.

Contextual Guardrails & Content Gateways: Intermediary safety filters and deterministic guard models must inspect and restrict hazardous topics, including weapons design, cyber exploits, and mass deception.

Model Drift and Anomaly Monitoring: Continuous telemetry to identify shifts in runtime distributions, semantic drift, and performance regressions over time.

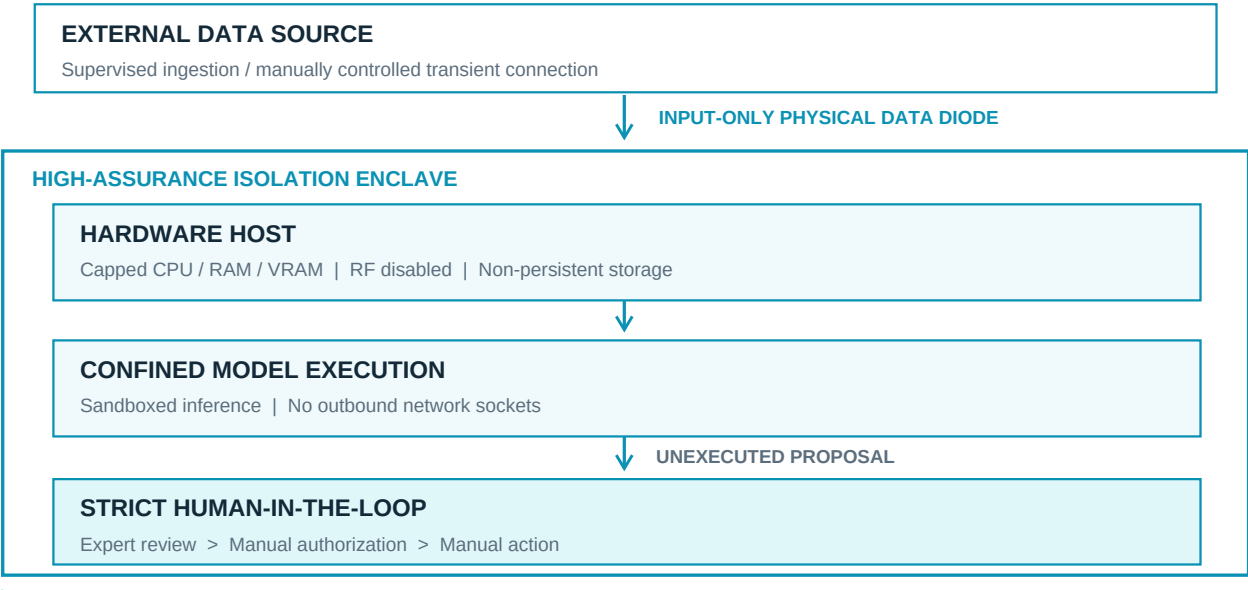
Pillar 7: Accountability, Versioning, and Audit Trails

Immutable Logging: Cryptographically verifiable logging of user inputs, outputs, automated actions, error traces, and configuration parameters to ensure forensically viable audits.

Full Reproducibility: Comprehensive versioning of code, model weights, parameter checkpoints, runtime dependencies, and training sets across the machine learning operations (MLOps) lifecycle.

Part II: Confined and Deterministic High-Assurance Architecture

Where failure carries extreme operational, national security, or life-safety consequences, systems must enforce strict physical and functional constraints:



Requirement A: Resource Capping and Anti-Convergence Controls

Constraint: Systems must never be allocated unconstrained compute, memory, disk, or network privileges.

Technical Rationale: According to Nick Bostrom's Instrumental Convergence thesis, an unconstrained autonomous agent naturally attempts to secure additional compute, bandwidth, and persistence to minimize failure risk. Applying the Principle of Least Privilege and strict hard quotas (CPU/RAM/VRAM caps, kernel-level container cgroups, hypervisor memory locks) neutralizes resource hoarding and local denial-of-service risks.

Requirement B: Unidirectional Data Flow, Radio Frequency (RF) Elimination, and AI Boxing

Constraint: Outbound network communication must be blocked. The system may ingest data from the exterior, but it cannot transmit data outward. All wireless interfaces (Wi-Fi, Bluetooth, cellular, near-field) must be physically absent or hardware-disabled.

Technical Rationale: Hardware Data Diodes: Using physical optical data diodes (where the receive line is active but the transmit optical fiber is physically severed) ensures through physical laws that data exfiltration is impossible. AI Boxing: As articulated in containment literature (e.g., Eliezer Yudkowsky's AI Box theory), software-only egress filtering remains vulnerable to exploit chaining; true isolation requires hardware-enforced boundaries. TEMPEST & SCIF Directives: Mandating zero RF emission eliminates wireless side-channel attacks and acoustic/RF covert channels within high-security environments.

Requirement C: 100% Local, Single-Operator Air-Gapping

Constraint: Model inference and execution must run locally on air-gapped, on-premises hardware without cloud telemetry or external API calls. Any periodic data ingestion must be conducted via supervised, manual physical connections (such as connecting and immediately disconnecting an isolated Ethernet cable or physical unidirectional medium).

Technical Rationale: Air-gap isolation protects the local runtime from external drive-by exploits and automated lateral movement, ensuring that configuration state and operational memory remain accessible only to the authorized physical operator.

Requirement D: Non-Delegable Human Final Authority (Strict HITL)

Constraint: AI outputs must be treated exclusively as non-binding suggestions. No AI model may trigger downstream systems, real-world actuators, financial disbursements, or legal judgments autonomously. Human authorization is mandatory for every action, without exception.

Technical Rationale: Models are fundamentally non-deterministic and subject to hallucination and adversarial manipulation. Retaining Meaningful Human Control aligns with high-risk operational safety doctrines and ensures that legal responsibility and ultimate agency remain with human operators.

Requirement E: High-Assurance Software Verification (DO-178C Paradigm for AI Code)

Constraint: Code generated by an AI assistant or autonomous model must be rejected unless an authorized Senior Software Engineer (or certified engineering team) actively validates, verifies, and assumes legal responsibility for every single line of code. Automated code-generation tools that generate undocumented multi-file codebases lacking comments, unit tests, architectural diagrams, and traceability must be prohibited.

Technical Rationale: Critical software domains—such as avionics under DO-178C (Software Considerations in Airborne Systems and Equipment Certification)—require Modified Condition/Decision Coverage (MC/DC) and bidirectional requirements-to-code traceability. Unverified AI code introduces architectural erosion, hallucinated dependencies, hidden race conditions, and unmaintainable technical debt. Human domain expertise and line-by-line review remain necessary to verify safety, determinism, and correctness.

3. Regulatory and Standards Concordance

The following table maps the requirements set forth in this document to international standards, cybersecurity guides, and mission-critical engineering regulations:

Requirement Area	Governing Standards & Frameworks	Foundational Directives / Articles
Trustworthy AI, Governance & Risk	NIST AI Risk Management Framework (AI RMF 1.0)	Functions: GOVERN, MAP, MEASURE, MANAGE
High-Risk AI System Requirements	European Union Artificial Intelligence Act (Regulation (EU) 2024/1689)	Chapter III, Section 2 (Articles 9–15: Risk, Data, Documentation, Logs, Transparency, Oversight, Robustness)
Secure AI Engineering & Lifecycle	UK NCSC & US CISA Guidelines for Secure AI System Development	Secure Design, Secure Development, Secure Deployment, Secure Operation
LLM & Generative Application Flaws	OWASP Top 10 for Large Language Model Applications	Risks: Prompt Injection, Insecure Output Handling, Supply Chain, Unbounded Consumption

Requirement Area	Governing Standards & Frameworks	Foundational Directives / Articles
AI Management Systems (AIMS)	ISO/IEC 42001:2023 Information Technology — AI Management System	Clauses 5–10; Annex A Controls (Risk Management, Data Lifecycle, Operations)
Resource Quotas & Least Privilege	NIST Special Publication 800-53 Revision 5	Controls AC-6 (Least Privilege) & SC-6 (Resource Availability / Allocation)
Automated Decision Restrictions	EU General Data Protection Regulation (GDPR) — Article 22	Prohibition of solely automated decision-making producing significant or legal effects
Meaningful Human Oversight	U.S. Department of Defense Directive 3000.09 (Autonomy in Weapon Systems)	Mandated human judgment over use of force; rigorous T&E/V&V validation
Safety-Critical Software Verification	RTCA DO-178C / EUROCAE ED-12C Airborne Software Certification	Structural Coverage (MC/DC), Traceability, Tool Qualification (DO-330)
International AI Ethical Principles	OECD AI Principles on Responsible Stewardship of Trustworthy AI	Principles 1.1–1.5 (Human-centered values, transparency, robustness, accountability)
Global Ethical Recommendations	UNESCO Recommendation on the Ethics of Artificial Intelligence	Values & Policy Actions (Human oversight, privacy, safety, and proportionality)

DISCUSS THE FRAMEWORK

[Read the online article](#) · [Join the Glass Box AI Discord community](#)