The AI OSI Stack: A Governance Blueprint for Scalable and Trusted AI

Version 2.0

Daniel P. Madden

Independent AI Researcher

danielpmadden.com

Originally conceived: September 09, 2025
This version: October 30, 2025

Abstract

Artificial intelligence is undergoing an infrastructural shift: from discrete applications toward a deeply layered, globally provisioned socio-technical substrate. However, most governance proposals still treat "AI" as a single opaque entity. This mismatch produces three systemic failures: (1) it hides where power and risk actually concentrate, (2) it makes regulation simultaneously overbroad and underinclusive, and (3) it enables monopolistic capture at technical chokepoints like hardware supply or API protocol control.

This paper presents Version 2.0 of the **AI OSI Stack**, a layered architectural and governance model inspired by the OSI networking paradigm. The Stack decomposes AI systems into seven layers: (1) Physical / Hardware, (2) Model Architecture, (3) Training / Optimization, (4) Instruction / Control, (5) Interface / Protocol, (6) Application, and (7) Governance / Trust. Each layer is defined by: core function, typical actors, dominant risks, governance levers, and auditable artifacts. This version adds: (a) an architectural methodology for mapping real systems to the Stack; (b) a treatment of "layer blurring" where decisions in lower layers constrain governance options in upper layers; (c) a sub-layering of Layer 4 to support persona-based alignment; and (d) an implementation outlook linking the Stack to NIST AI RMF, ISO/IEC 42001, and the EU AI Act.

The core claim is that *trust can be made portable* when AI is designed as layered infrastructure. Governance then becomes a design-time discipline, not a late-stage compliance chore.

1 Introduction: From Opaque Systems to Accountable Infrastructures

The most common failure in AI discourse is conceptual collapse. Hardware, models, training pipelines, alignment strategies, API layers, and even regulatory overlays are often referred to under the single heading of "AI." This collapse obscures the anatomy of control. It becomes impossible to say where a failure originated (data? model? alignment? interface? deployment context?), and therefore impossible to assign accountability.

At the same time, the AI ecosystem is displaying what we can call *mitosis*: a branching into multiple specialized lineages (domain-tuned models, sectoral copilots, embedded/edge deployments) inside a high mortality environment. Most lineages will die; a few will harden into infrastructure. In such an environment, clear layering is not an aesthetic preference. It is a survival strategy for governance. Without it, regulators, CISOs, and enterprise architects will all talk past one another.

The historical internet teaches us that architectural clarity precedes scalable trust. Networking became global only when we knew *which function lived where*. The AI OSI Stack is an attempt to achieve the same for AI, with an explicit governance layer instead of a hand-waved "policy later" stance.

2 Historical Precedent: What the OSI Model Actually Gave Us

The ISO/IEC 7498-1 OSI model did three things that AI badly needs:

1. **Separated concerns** so that lower-level transmission issues did not pollute higher-level application design.

2. **Made interfaces stable** so multiple vendors could interoperate without surrendering their internal implementations.

3. **Created a shared language** for regulators, standards bodies, and implementers.

Early networking suffered from vendor lock-in and proprietary protocols. Today's AI is similar: closed APIs, model access as a service, opaque telemetry, and hardware controlled by a handful of providers. The lesson is obvious: if we do not standardize the *conversation surface*, we will re-create platform monopolies at Layer 5 (Interface / Protocol) and allow supply-chain fragility at Layer 1 (Physical / Hardware) to cascade through the whole stack.

3 The AI OSI Stack: Overview

The Stack decomposes AI into seven layers:

L	Name	Purpose / Governance Focus
1	Physical / Hardware	Compute, networking, energy, and location of infrastructure; supply-chain and geopolitical risk.
2	Model Architecture	Capability ceiling and interpretability; research concentration and openness.
3	Training / Optimization	Data, costs, provenance, efficiency; economic accessibility and reproducibility.
4	Instruction / Control	Human intent, alignment, safety; misalignment and covert influence.
5	Interface / Protocol	APIs, middleware, orchestration; chokepoints, logging, portability.
6	Application	Societal and sectoral impact; context-sensitive risk and user transparency.
7	Governance / Trust	Audits, compliance, accountability artifacts; regulatory capture and legitimacy.

This is not a "tech-only" stack. It is an *institutional* stack: it names where standards bodies, auditors, safety labs, and policymakers can attach their work.

4 Architectural Methodology: How to Use the Stack

A common question from practitioners is: *"How do I know which layer I'm in?"* Version 2.0 introduces an explicit method.

4.1 Step 1: Identify the System Boundary

Define the AI system you are analyzing: a model-as-a-service API, an internal enterprise chatbot, an agentic workflow, or a multimodal RAG system. Name its *primary* delivery surface (usually a Layer 6 interface or a Layer 5 API).

4.2 Step 2: Trace Downstream Impact

Ask: who is harmed or helped by the system? That answer usually lives in Layer 6 (Application) and Layer 7 (Governance / Trust) — e.g., hiring tools, medical decision aids,

public-sector assistants.

4.3 Step 3: Trace Upstream Dependencies

Walk downward: which API (L5) does this rely on? Which model (L2) was used? Was it fine-tuned (L3) on proprietary data? Which cloud or accelerator (L1) is it running on, and is that subject to export control?

4.4 Step 4: Produce a Stack-Aligned Report

For each layer, write:

- **Actor** (who controls it),
- **Risk** (dominant failure mode),
- **Evidence** (audit artifact, log, model card),
- **Regulator / Standard** (who can govern it).

This becomes an *AI OSI Layer Report*. Enterprises can require such a report before deploying L6 tools.

4.5 Step 5: Version the System

Use semantic versioning across layers. A radical architecture change (L2) is a **major** bump; a new control schema (L4) is a **minor** bump; improved documentation (L7) is a **patch**. This makes AI behavior *trackable over time* — crucial for safety incidents.

5 Layer-by-Layer Analysis (Painstaking Detail)

5.1 Layer 1: Physical / Hardware

Scope. Specialized accelerators (GPUs, TPUs, ASICs), high-bandwidth networking, storage, data center siting, power and cooling, and edge devices when inference moves on-device.

Why separate it? Because hardware concentration can silently centralize the entire AI economy. If five actors can decide who gets accelerators, they indirectly control who can train frontier models.

Dominant risks.

- *Supply-chain concentration:* NVIDIA-style dominance leading to geopolitical leverage.
- *Export control fragility:* sudden policy shifts affect training availability.
- *Energy and water constraints:* AI directly competing with public infrastructure.

Governance levers.

- National or regional hardware registries,
- incentives for hardware diversity and open ISA,
- environmental disclosure requirements.

Audit artifacts. Infrastructure bill of materials, siting reports, supply diversity score.

5.2 Layer 2: Model Architecture

Scope. Transformers, diffusion, autoregressive multimodal models, neurosymbolic hybrids, longer-context architectures, sparse expert mixtures.

Why separate it? Architecture defines what is even *possible* at higher layers. A model not designed for interpretability will make L4 and L7 much harder.

Dominant risks.

- *Research centralization:* frontier labs set the capability agenda.
- *Capability leakage:* open weights without alignment scaffolding.
- *Opaque inductive biases:* hard to certify safety properties.

Governance levers.

- Capability and safety reporting per release,
- funding for interpretable architectures,
- red-teamable model specs.

Audit artifacts. Model cards, safety reports, architectural diagrams.

5.3 Layer 3: Training / Optimization

Scope. Data acquisition, curation, de-duplication, filtering, optimizer choice, fine-tuning, distillation, reinforcement-based methods, LoRA/PEFT adapters.

Dominant risks.

- *Data provenance opacity:* copyrighted or sensitive personal data.
- *Economic barrier:* high training cost creates market concentration.
- *Safety erasure:* after-market fine-tuning that removes guardrails.

Governance levers.

- Training run registries,
- dataset disclosure or at least dataset category disclosure,
- reproducibility / re-trainability standards.

Audit artifacts. Training logbooks, hyperparameter sheets, data lineage graphs.

5.4 Layer 4: Instruction / Control

This is the most politically and ethically dense layer.

Scope. Prompts, system messages, tool-calling policies, embeddings, safety policies, RLHF, Constitutional AI, persona-based controllers, epistemic blueprints.

Why is it special? This is where *capability becomes intention.* A model is powerful in L2–L3; it becomes dangerous or dignifying in L4.

Sub-layers (v2.1).

- **L4a Control:** raw prompting, context windows, tool selection, policy injection, jailbreak resistance.
- **L4b Ethical/Value Reasoning:** Persona Architecture, Heartwood Safety Core, Dignity as Constraint, Decision Insurance mechanisms.

Dominant risks.

- Misalignment between stated values and actual controller logic,
- prompt injection and tool hijacking,
- covert persuasion (using persona tone to influence users).

Governance levers.

- independent audits of safety policies,
- publication of high-level control intent ("this system refuses X, escalates Y"),
- mandatory refusal and escalation patterns.

Audit artifacts. Persona manifests, safety test suites, red-team reports, "why refused" logs.

5.5 Layer 5: Interface / Protocol

Scope. REST/GraphQL model APIs, streaming endpoints, agent runtimes (OpenAI-style, Anthropic MCP-style), middleware (LangChain, LangGraph), orchestration DSLs.

Why it matters. This is the modern chokepoint. Whoever owns the interface can:

- change pricing,
- change available tools,
- log everything,
- and unilaterally disconnect clients.

Dominant risks.

- interface monopolies that limit L6 innovation,

- opaque telemetry that prevents L7 auditing,
- automatic tool execution without human visibility.

Governance levers.

- open protocol standards,
- exportable audit trails,
- rights to workflow portability (move your agents elsewhere).

Audit artifacts. API call logs, agent execution graphs, protocol conformance tests.

5.6 Layer 6: Application

Scope. Everything end-users actually see: copilots, chatbots, decision support, vertical agents for health, finance, education, public administration.

Dominant risks.

- *Contextless deployment:* same model used in trivial vs. high-stakes settings.
- *Companion trap:* systems simulating intimacy to extract data or shape behavior.
- *Shadow AI:* teams deploying unvetted tools internally.

Governance levers.

- context-specific deployment policies,
- human-in-the-loop for high-risk use,
- user-facing model cards and limitation disclosures.

Audit artifacts. Decision Insurance briefs, application-level logs, sectoral impact assessments.

5.7 Layer 7: Governance / Trust

Scope. Everything that makes trust *portable*: audits, compliance mappings, safety boards, transparency tools, incident reporting, cross-jurisdiction alignment.

Dominant risks.

- regulatory capture,
- fragmented global regimes,
- "ethics theater" (appearance of governance without power).

Governance levers.

- shared audit schemas,

- public safety incident databases,
- cryptographic attestation for logs and artifacts.

Audit artifacts. Solomon Briefs, Clarity Packages, Governance Maps, layer-to-regulation matrices.

6 Layer Blurring and Feedback Loops

Real systems don't stay in their lanes. Three common blurs:

1. **L2→L4 leak:** architectural choices limit alignment options.
2. **L5→L6 lock-in:** API provider constrains what apps can do.
3. **L7→L3 constraint:** governance rules require specific training data provenance.

The Stack is still useful here, because it gives us language to *describe* the blur.

7 Power and Chokepoint Analysis

Without the Stack, policymakers hit L6 (visible) and miss L1/L5 (structural). With the Stack, they can:

- regulate L5 behavior to protect L6 competition,
- subsidize L1 diversification to reduce L2–L3 dependency,
- require L4 transparency to enable L7 audits.

8 Integration with Existing Governance Frameworks

8.1 NIST AI RMF

NIST defines *govern, map, measure, manage*. The Stack tells you *where* to perform each: e.g., "measure" at L3 and L6, "govern" at L7, "map" across L1–L5.

8.2 ISO/IEC 42001

ISO/IEC 42001 describes an AI management system. The Stack gives you the architectural surface to bind controls to: e.g., control 5.3 maps to L4 audits; control 6.x to L5 telemetry.

8.3 EU AI Act

Risk categories (unacceptable, high, limited, minimal) can be mapped to layers:

- high-risk *applications* → L6
- documentation / logging → L5
- data and training transparency → L3

8.4 Comparative Analysis

Unlike principle-based frameworks (OECD, UNESCO), the AI OSI Stack is *topologically explicit*. It says: "risk is not evenly distributed; it pools at interfaces and supply chains."

9 Applications for Different Stakeholders

9.1 Enterprises

Run **Stack-based risk reviews**. Require vendors to provide L4 and L5 artifacts, not just L6 marketing.

9.2 Policymakers

Use Stack to target intervention: API portability laws (L5), supply-chain diversification (L1), auditability mandates (L7).

9.3 Researchers

Work safely at a single layer while preserving interoperability — a core promise of the OSI analogy.

10 Implementation Outlook

Short term:

- publish Stack-aligned model cards,
- create open schemas for L5 audit logs,
- run red-team exercises specifically on L4b.

Medium term:

- integrate Stack language into procurement and RFPs,

- develop open-source "Governance Map" generators.

Long term:

- cryptographically attest L4 and L5 artifacts,
- converge on cross-border L7 schemas.

11 Future Work and Evolution

The Stack is versioned. v3.x may:

- split L4 formally,
- add an "econ/reg" meta-layer for incentives,
- standardize artifact formats (Clarity Packages, Solomon Briefs).

12 Conclusion: Trust as Infrastructure

The central claim stands: AI cannot be governed as a single blob. It must be governed as an architecture. Once decomposed, we can regulate the right thing, at the right layer, with the right tool — and we can keep the ecosystem open, competitive, and auditable.

Appendix A: Glossary of Key Concepts

Persona Architecture: a structured, auditable way of encoding role, ethics, context, and refusal logic inside L4. **Decision Insurance:** an L6/L7 pattern that records assumptions, alternatives, and sources to protect against reasoning failure. **Trust as Infrastructure:** the principle that trust should be produced by system design, not marketing. **Layer Blurring:** cross-layer dependencies that must be named and, where possible, hardened.

About the Author

Daniel P. Madden is an independent AI researcher and IT specialist focused on feasibility, governance, and reasoning systems. His AI Lab Notebook documents experiments in persona architecture, decision auditing, and epistemology by design. See https://danielpmadden.com for related work.

References

[1] National Institute of Standards and Technology. 2023. *AI Risk Management Framework (AI RMF 1.0).*

[2] ISO/IEC 42001:2023. *Artificial Intelligence Management System.*

[3] European Commission. 2024. *Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (AI Act).*

[4] International Organization for Standardization. 1994. *Information technology – Open Systems Interconnection – Basic Reference Model (ISO/IEC 7498-1).*

[5] Partnership on AI. 2024. *Guidance for Safe Foundation Model Deployment.*