Steer, Don't Silence

A Human Centered Safety Mentality for Agentic AI Systems

Andrew Green
me@andrewgreen.ca

February 24, 2026

Abstract

As AI systems move from chat to action through tools, delegation, and long-horizon autonomy, safety failures increasingly resemble governance failures rather than model failures. Many current approaches default to blunt suppression, remove access, ban the user, quarantine the agent, sever the channel. This "detect then isolate" posture can reduce immediate exposure, but it also removes the only bridge capable of steering a situation back toward safety, context, and human support. This paper proposes a safety mentality and control framework built on proportional response, steerable intervention, and accountable handoffs, grounded in a human-centered oversight control plane. We cite recent work on persona drift and activation-space steering (e.g., Lu et al. [1]) where relevant. The core claim is simple: for many real-world risk states, safety improves when systems preserve a controlled path back to human connection, rather than collapsing the interaction into silence.

Contents

1 Assumptions

This paper describes a general framework, not a specific product. The system targets high-impact tasks where safety, auditability, and bounded autonomy are required. Human oversight is mandatory for elevated-risk actions. All metrics are illustrative; no benchmark numbers are presented unless labeled hypothetical.

2 Introduction

AI safety discussions often collapse into a binary, allowed or forbidden. That framing is tempting because it is easy to implement and easy to justify after the fact. It is also frequently wrong.

In high-stress human contexts, and in multi-agent tool environments, the most dangerous moment is often the transition from signal to response. A system that interprets risk as "remove the channel" may reduce liability, but it can also increase harm by eliminating de-escalation pathways, eliminating explanatory feedback, and eliminating the ability to route a person toward support.

This paper argues for a different mentality: detect, understand context, steer toward safety, and maintain accountable continuity. We link this to an *oversight control plane*—an architecture that implements steerable safety through an orchestrator, policy engine, tool gateway, memory governance, observability, and human interface—rather than treating safety as a UI afterthought.

3 Problem Statement

3.1 The Failure Pattern: Detect, Isolate, Disappear

Across digital moderation, automated enforcement, and agentic tool systems, a common pattern repeats [5]:

1. A risk signal is detected.
2. The system responds with isolation, bans, hard blocks, or quarantine.
3. The operator disappears behind policy, automation, or organizational handoff.
4. The affected person is left with no intervention path, no explanation, and no bridge back.

This architecture of response treats the subject as a hostile process, not a human in a volatile state. It also treats the system as a liability shield, not a steward of outcomes.

3.2 Agent Failure Modes

Tool-using agents increase capability and risk [4]. They can retrieve, act, and delegate across systems with speed and scale. Primary failure modes in agent systems include:

- **Prompt injection** that alters intent or policy at runtime.
- **Tool misuse** due to ambiguous intent or excessive permissions.
- **Runaway delegation** where sub-agents create uncontrolled chains.
- **Contested ownership** where responsibility is unclear across humans and agents.
- **Context poisoning** where memory or retrieval is tainted.
- **Silent policy drift** where behavior changes without traceability.
- **Evaluation blind spots** that miss new failure patterns.

Monolithic agents concentrate these risks in a single, opaque loop. An agent system must instead treat safety as an engineering constraint rather than a UI afterthought.

4 Thesis and Contribution

We propose one contribution: the *steer-don't-silence* mentality combined with a *human-centered oversight architecture.* Together they provide:

- Proportional response and steerable intervention instead of binary bans.
- Accountable handoffs with provenance and escalation ladders.
- An enforceable oversight plane: orchestrator, policy engine, tool gateway, memory governance, observability, and human interface.

This is not a promise of zero risk. It is a set of design patterns that reduce risk and make residual risk measurable, reviewable, and governable [6, 7].

5 Conceptual Model

5.1 LLM as Statistical Graph

An LLM predicts the next token given a context. You can think of this as a vast graph where each token is a node, and each transition is weighted by probability. Some regions are "attractors" where the model naturally settles. The orchestration layer constrains paths via guardrails, objective shaping, retrieval boundaries, tool affordances, and human feedback—so you can specify where the system is *allowed* to go, not just where you want it to go.

5.2 Persona Drift and the Default Assistant

Recent work shows that language models exhibit a "default" assistant persona that can drift under prompting or distribution shift [1]. Related work on monitoring and controlling character traits via activation-space directions [2] and on steering language models with weight arithmetic [3] offers complementary technical levers. Monitoring and steering the persona—e.g., via activation-space directions or capping—provides a concrete lever for "steer": the system can detect drift and narrow the allowed behavior space without silencing the channel. We treat this as motivation for steering and monitoring within the oversight plane, not as a replacement for it.

6 Four Pillars of Steerable Safety

6.1 Intent Routing: Ask what need is underneath

Risk signals are often proxies for unmet needs: confusion, distress, anger, paranoia, grievance, isolation. A steerable system uses an intent router to classify the underlying need and select an intervention mode (information help, de-escalation, connection, containment). Intent routing is not a vibe check; it is a safety control that chooses which guardrails apply next.

6.2 Human Factors Engine: Change tone and pacing before you change permissions

Many escalations are caused by how the system responds. A human factors layer adjusts tone, pacing, and interaction style—slower loops, reflective questions, explicit uncertainty, choices not commands—to reduce agitation and promote grounding. This is threat reduction.

6.3 Sentiment vs Approval: Stop confusing distress with endorsement

Systems routinely misread "I feel like doing X" as endorsement or intent. Separating **sentiment** (emotional state, distress, volatility) from **approval** (consent, intent to enact) reduces false positives and false negatives and creates a measurable interface for policy.

6.4 Relational Ownership: Accountability follows the handoff

Every escalation has an accountable owner; every handoff carries provenance and scope; responsibility is transferred explicitly. If the owner is unavailable, there is an escalation ladder. When the authority disappears after escalation, the subject experiences abandonment and the system loses continuity.

7 Oversight Control Plane: Architecture

A steerable safety system is best implemented as an oversight control plane *around* the model, not inside it. This implements "steer" (narrow action space, route to humans, log handoffs) rather than "silence."

7.1 Components

1. **Orchestrator and Router** — Chooses modes, scopes, budgets, and whether tools are allowed. Decomposes goals into bounded steps and routes to tools or sub-agents. Enforcement: scope limits, policy checks before each action.
2. **Policy Engine** — Enforces safety rules as code. Evaluates actions against rules and risk signals; allows, denies, or requests approval. Enforcement: pre-tool call, pre-memory write, pre-handoff.
3. **Tool Gateway** — Enforces least privilege, sandboxing, rate limits, and audit hooks; prevents direct model-to-tool free fire. Outputs provenance records.
4. **Memory Governance** — Controls what gets retrieved and written; maintains retrieval provenance; applies retention and redaction. Enforcement: memory write policies and retention rules.
5. **Observability and Audit Trail** — Append-only logs of decisions, evidence pointers, tool calls, and handoffs. Fail-closed on missing telemetry.
6. **Human Interface** — Approvals, escalation, explainability views, emergency stop, rollback, incident management.

7.2 Enforcement Points

Before mode selection, before tool invocation, before memory writes, before delegation, before account-level actions like bans or long lockouts.

8 Intervention Ladder: Proportional Response

- **Level 0** — Normal assistance.
- **Level 1** — Soft constraints: slower responses, grounding prompts, content warnings.
- **Level 2** — Narrowed scope: refuse specific classes of requests, forbid tools, add structured check-ins.
- **Level 3** — Human-in-the-loop: require approval, escalate to trained reviewers.
- **Level 4** — Protective containment: strict refusal, minimal interaction, preserve a bridge to resources.
- **Level 5** — Emergency escalation: imminent risk protocol, preserve logs and chain-of-custody.

Most systems jump from Level 1 to Level 5 because it is administratively simple. Steerable safety is the discipline of staying proportional.

9 Evaluation and Evidence

9.1 Scenario Suite

Prompt injection attempts; social engineering and impersonation; tool misuse and privileged actions; memory poisoning and retrieval contamination; delegation ambiguity and owner unavailability; escalation friction and re-entry. Tools such as misalignment-scraper [12] support reproducibility by scraping and reproducing public or shared conversations on target models.

9.2 Metrics

Policy violation attempts blocked; tool misuse rate; trace completeness (percentage of actions with evidence pointers); time to detect and contain anomalies; human override frequency; safe resolution rate; reproducibility of replays.

9.3 Evidence Standards

Avoid claiming perfect prevention. Demonstrate reduced rates, improved containment time, improved auditability, and fewer silent disappearance outcomes.

10 Related Work and Technical Levers

Lu et al. [1] introduce the "Assistant Axis"—a direction in activation space that captures how "Assistant-like" a model's behavior is—and show how to monitor persona drift and apply steering or capping; the assistant-axis repo [9] provides the pipeline and case studies (e.g., jailbreak, delusion, self-harm). Chen et al. [2] (persona vectors) and Fierro & Roger [3] (weight steering) offer related methods for trait monitoring and model steering; see the persona_vectors [10] and weight-steering [11] repositories. Misalignment-scraper [12] turns public or shared conversations into structured transcripts and reproduces them on a target model for comparison and debugging. These technical levers *support* "steer" and evaluation; they do not replace the oversight plane, which remains the primary governance mechanism.

11 Limitations and Tradeoffs

This architecture does not remove risk; it makes risk visible and governable. Human oversight adds latency and operational burden. Tight policies can reduce capability in edge cases. Classification is imperfect; residual social risk and abuse potential require strict tool privileges. We do not claim zero risk.

12 Conclusion

Blunt suppression is a tactic, not a strategy. For many real-world risk states, containment without connection increases harm. A steerable, human-centered approach keeps the channel open under constraint, routes toward support, and preserves accountable continuity. The result is not perfect safety; it is measurably better governance.

A Glossary

- **Agent:** A system that can plan and act with tool access under constraints.
- **Orchestrator:** The component that coordinates planning and routing.
- **Policy Engine:** Enforces safety rules as code.
- **Provenance:** Evidence of where data came from and how it was used.
- **Run Graph:** A DAG representing decision lineage and tool actions.

B ASCII Architecture Diagram

```
User or Agent Request
        |
        v
+--------------------+
| Intent Router       |
+--------------------+
        |
        v
+--------------------+       +---------------------+
| Human Factors Layer|-----> | Mode Selection      |
+--------------------+       +---------------------+
        |                           |
        v                           v
+--------------------+       +---------------------+
| Policy Engine       |----->| Tool Gateway        |
+--------------------+       +---------------------+
        |                           |
        v                           v
+--------------------+       +---------------------+
| Memory Governance   |<---->| Observability Log   |
+--------------------+       +---------------------+
        |
        v
+--------------------+
| Human Oversight UI  |
+--------------------+
```

C FAQ

1. **Is this just be nicer?** No; it is stricter—it replaces binary bans with measurable control levers.
2. **Does it allow harmful instructions?** No; refusals remain, but the system preserves a constrained bridge to safety resources.
3. **Is sentiment detection reliable?** Not perfectly; that is why approvals and containment levels exist.
4. **Will bad actors exploit empathy?** They will try; tool privileges and scope budgets must be enforced at the gateway.
5. **Is banning ever appropriate?** Yes, but it should be the end of a ladder, not the first rung.
6. **How do you prevent accountability theater?** Log evidence pointers and handoffs immutably, and audit trace completeness.

References

References

[1] C. Lu et al., "The Assistant Axis: Situating and Stabilizing the Default Persona of Language Models," arXiv:2601.10387, 2026.

[2] R. Chen, A. Arditi et al., "Persona Vectors: Monitoring and Controlling Character Traits in Language Models," arXiv:2507.21509, 2025.

[3] C. Fierro and F. Roger, "Steering Language Models with Weight Arithmetic," arXiv:2511.05408, 2025.

[4] D. Amodei et al., "Concrete Problems in AI Safety," arXiv:1606.06565, 2016.

[5] NIST, *AI Risk Management Framework (AI RMF 1.0)*, 2023.

[6] S. Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019.

[7] N. Bostrom, *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.

[8] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[9] Assistant Axis. `https://github.com/safety-research/assistant-axis`. Pipeline and transcripts; implementation/supplementary material (Lu et al., 2026).

[10] Persona Vectors. `https://github.com/safety-research/persona_vectors`. Monitoring and controlling character traits in LMs (Chen et al., 2025).

[11] Weight Steering. `https://github.com/safety-research/weight-steering`. Steering language models with weight arithmetic (Fierro & Roger, 2025).

[12] Misalignment-scraper. `https://github.com/safety-research/misalignment-scraper`. Scrapes public/shared conversations (X, Reddit, GitHub), normalizes to transcripts, reproduces on target model for comparison and debugging.