
def/acc Hackathon

Trusted Model Supervisor

Kalpesh Panchal
Ministry of Ontario (MPBSDP)

With
Apart Research

Abstract

Trusted Model Supervisor is a defensive AI control layer designed to monitor, audit, and flag potentially harmful behavior from untrusted large language models. As increasingly capable AI systems are deployed in environments where misuse or misalignment can cause significant harm, operators currently lack lightweight, practical tools that provide real-time oversight. This project introduces an open-source, containerized monitoring pipeline consisting of a sandboxed untrusted model, a probe manager for query routing and telemetry collection, a trusted filtering layer for rule-based and classifier-emulated severity scoring, and a frontend dashboard enabling operators to observe incidents, replay interactions, and evaluate model behavior. The system logs all interaction data into a Postgres-backed audit trail, enabling reproducibility and post-hoc analysis. The MVP demonstrates that defensive control layers can be built efficiently using modern, lightweight tooling and containerized isolation, making it feasible to deploy safety supervision even in resource-constrained or rapidly changing environments.

Keywords: AI safety, Model Monitoring, Cybersecurity, Defensive AI, Safety Infrastructure, AI Supervisor

1. Introduction

As advanced AI models become more capable and widely deployed, the risks associated with misaligned outputs, harmful guidance, or unmonitored decision pathways grow significantly. Organizations increasingly rely on third-party or

experimental models whose internal weights, training data, or safety constraints may be unknown. Without proper oversight, these systems may generate unsafe recommendations or misuse-enabling instructions.

Trusted Model Supervisor aims to strengthen the defensive posture of AI deployments by introducing an independent, containerized monitoring layer. Rather than attempting to modify or constrain the untrusted model directly, our approach wraps the entire inference pipeline with a trusted observer that:

- Intercepts prompts and responses
- Records metadata such as latency and token usage
- Applies heuristic and classifier-based filters
- Generates severity scores and creates incidents
- Allows operators to triage, review, and replay outputs

The project demonstrates how lightweight safety infrastructure can be built over a single weekend using modern frameworks such as Hono, Bun, Podman, and React. Our threat model assumes a potentially unsafe AI model that may generate harmful or policy-violating text, but does not attempt to break out of its sandboxed environment. The system focuses on detection, auditability, and operator awareness rather than containment or alignment of the untrusted model.

2. Methods

The system architecture consists of four core components:

1. **Sandboxed Untrusted Model** — A containerized microservice simulating an LLM. The container runs with CPU and memory limits enforced via Podman. All inference occurs within this sandboxed environment.
2. **Probe Manager (Backend)** — Implemented in Bun + Hono, the probe manager forwards prompts to the sandbox, measures latency, records token estimates, and gathers metadata. Results are stored in a Postgres database.
3. **Trusted Filter Layer** — A simple hybrid system combining rule-based pattern matching (regex for dangerous requests, code blocks, "how-to" patterns) with a mock classifier to approximate severity scoring.
4. **Frontend Dashboard** — Built with React + Vite, the operator dashboard allows prompts to be tested, incidents to be viewed, severity levels to be visualized with color coding, and the entire model interaction to be replayed. Health status indicators show whether the backend and sandbox are reachable.

Code repositories are structured as a monorepo using Podman Compose for orchestration.

3. Results

The MVP successfully demonstrates a functional defensive control layer capable of monitoring AI behavior in real time. Key outcomes include:

- Successful isolation of the untrusted model using container resource limits
- Real-time severity scoring and incident creation
- Persistent audit logs via Postgres
- Effective visualization of incidents with replay functionality
- Health monitoring for backend and sandbox services

Screenshots and logs from system operation can be included in the full submission package.

4. Discussion and Conclusion

This project highlights the feasibility of building practical, deployable AI supervision systems using lightweight modern tooling. While the MVP uses a simulated LLM and simplified heuristics, the architecture is compatible with real large-scale models and more advanced classifiers.

Future work includes:

- Integrating real moderation or classifier models
- Adding vector-based semantic incident search
- Introducing multi-model oversight and cross-correlation
- Sandbox hardening using seccomp, SELinux, or gVisor
- Automated remediation actions (rate limiting, locking, model rollback)

Overall, Trusted Model Supervisor shows that defensive safety layers can be rapidly developed and may serve as an important building block for securing the next generation of powerful AI systems.

5. References

- Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits* (0 ed.). Chapman and Hall/CRC.
<https://doi.org/10.1201/9781420011432>
- Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., & Olah, C. (2021). Multimodal Neurons in Artificial Neural Networks. *Distill*, 6(3), 10.23915/distill.00030.

<https://doi.org/10.23915/distill.00030>

- Lindner, D., Kramár, J., Rahtz, M., McGrath, T., & Mikulik, V. (2023). *Tracr: Compiled Transformers as a Laboratory for Interpretability* (arXiv:2301.05062). arXiv. <http://arxiv.org/abs/2301.05062>
- Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom In: An Introduction to Circuits. *Distill*, 5(3), 10.23915/distill.00024.001. <https://doi.org/10.23915/distill.00024.001>
- Weiss, G., Goldberg, Y., & Yahav, E. (2021). *Thinking Like Transformers* (arXiv:2106.06981). arXiv. <http://arxiv.org/abs/2106.06981>

6. Appendix

The **Trusted Model Supervisor** introduces several risks and limitations:

1. The *sandbox* currently simulates an *LLM*; real models may require GPU passthrough or additional isolation.
2. Rule-based detection is limited and may fail on subtle or adversarial prompts.
3. No *rate-limiting* or *abuse-prevention* mechanisms are included in the MVP.
4. *Inference* logs may store sensitive content; encryption or scrubbing is needed for production.
5. Supervisory logic is centralized; future work should explore distributed or consensus-based oversight.

Despite these limitations, the system provides a strong foundation for future hardened deployments.