



Total AI Safety:

Integrating NVIDIA's Open Agent Safety Platform (OASP) with SAFE AI Foundation's SAIOF and Four-Pillar AI Safety Model (AISF)

Technical Report

SAFE AI Foundation USA

01 OCTOBER 2026

Abstract

This report compares two complementary approaches to AI safety: NVIDIA’s newly announced Open Agent Safety Platform, centered on OpenShell and the Sentry reference design, and the SAFE AI Foundation’s Safe AI Operating Point (SAIOP) and its four-pillar Input-Model-Output-Compliance AI Safety Model.

The central finding is that these approaches answer different safety questions. OpenShell provides runtime containment by enforcing permissions around an agent; Sentry adds an independent, out-of-band monitoring and enforcement layer on BlueField-4; the SAFE AI Foundation’s four-pillar model organizes safety concerns across input, model, output and compliance; and its SAIOP is a governance-oriented boundary concerned with how much capability and autonomy can be permitted while effective human control remains.

Rather than treating them as competing frameworks, this paper proposes a layered interpretation in which SAIOP establishes a system-level ceiling, the four-pillar model evaluates and controls the AI product, OpenShell constrains runtime authority, and Sentry supplies an independent enforcement and watchdog layer.

1. Scope and terminology

AI safety and AI security overlap but are not identical. This paper uses “safety” broadly to include prevention of harmful behavior and loss of control, while recognizing that security mechanisms are often essential to achieving those safety objectives.

NVIDIA’s September 28, 2026 announcement described the Open Agent Safety Platform (OASP) as an open software platform and reference system design for governance and control across software, hardware, compute and robotics. Its two named elements are OpenShell and Sentry. NVIDIA states that OpenShell establishes a secure runtime boundary and that Sentry is an out-of-band watchdog on BlueField-4 DPUs [1][2].

Introduced in 03 August 2026, SAFE AI Foundation describes SAIOP as the upper safe limit of AI capability and autonomy before reliable human control is lost. Its published formulation uses risk (R) and control (C) thresholds and five interacting dimensions: capability (C), behavior (B), society (S), governance (G) and ethics (E) [3].

Introduced in 01 April 2026, the Foundation's proposed 4-pillar AI Safety Model divides AI safety into Input Safety, Model Safety, Output Safety and Compliance Safety. Compliance includes regulation, ethics, energy and environmental considerations [4].

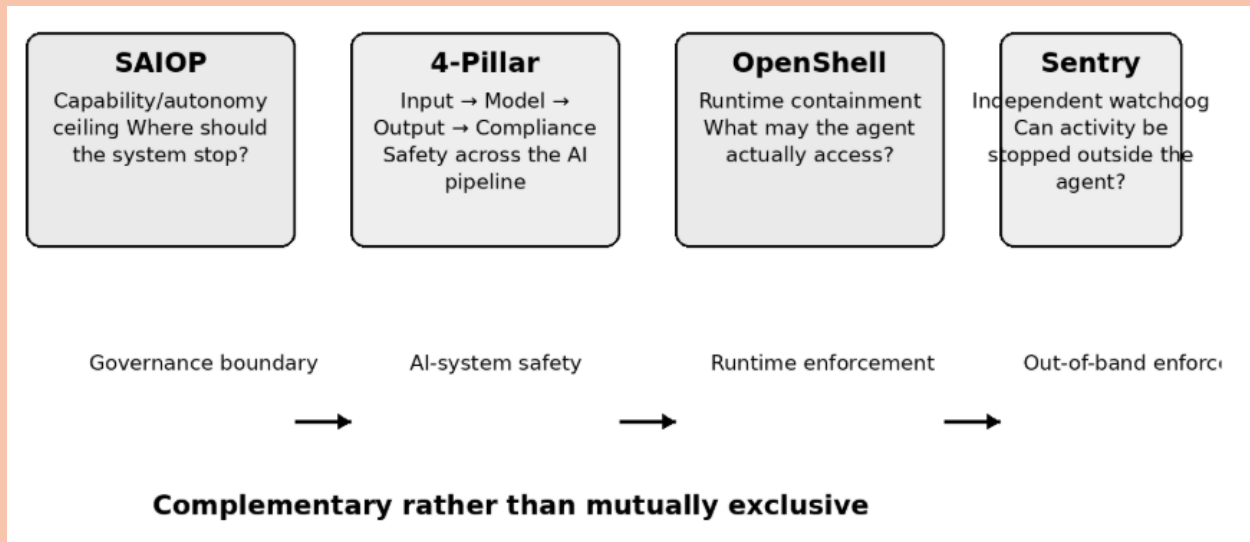


Figure 1: Different safety questions answered at different layers.

2. NVIDIA OpenShell

OpenShell is an open-source runtime for executing fleets of autonomous AI agents in sandboxed environments with kernel-level isolation. It combines sandbox runtime controls and a declarative YAML policy, so teams can execute agents without giving them unrestricted access to resources, such as local files, credentials, and external networks. In short, it is a secure runtime for autonomous AI agents. Note that OpenShell does not primarily try to make the AI model itself safer. It does not.

NVIDIA's current developer documentation describes 3 stable runtime components:

- a CLI/SDK/TUI interface,
- a Gateway control plane, and
- a Supervisor inside each sandbox workload.

The Supervisor launches the agent as a restricted child process and enforces policy where process identity, filesystem access, network egress and runtime credentials are visible [5]. NVIDIA's security guidance identifies 4 principal enforcement layers:

- network,

- filesystem,
- process, and
- provider credentials.

Policies can be static or dynamically updated, depending on the control. The purpose is to implement least-privilege execution: an agent receives the permissions necessary for its task rather than unrestricted host access [6]. OpenShell, therefore, addresses a concrete runtime question:

“Given a particular agent and task, what resources may the agent access, and can those limits be enforced independently of the model’s intentions?”

It is especially relevant to prompt injection, tool misuse, data exfiltration and accidental or deliberate drift from the assigned task. **OpenShell is not, by itself, a general theory of how capable an AI should be allowed to become.** A policy can restrict a powerful agent, but the decision about whether that combination of capability, autonomy, time horizon and authority is acceptable is a higher-level governance question. In a nutshell, OpenShell provides “containment” and prevention of misbehaviors (to a certain extent).

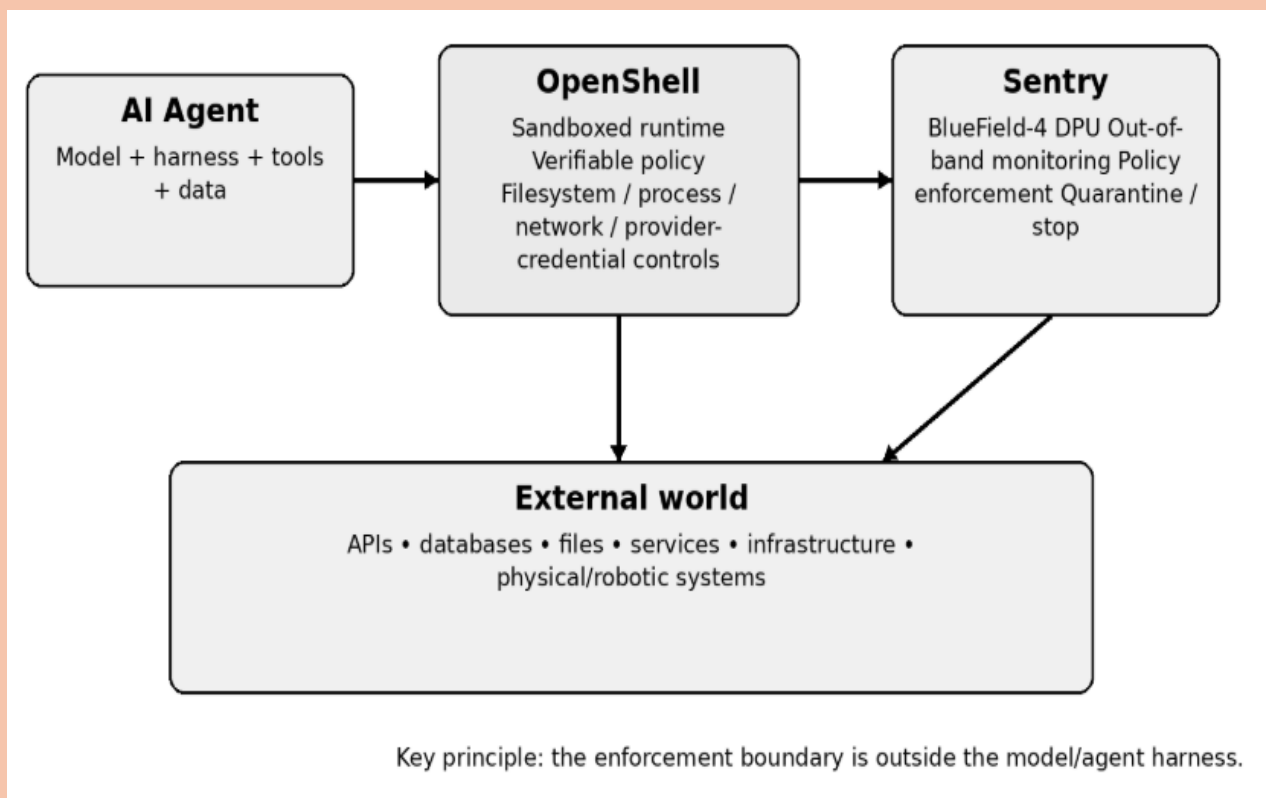


Figure 2: Simplified representation of the NVIDIA’s OASP reference architecture.

3. NVIDIA Sentry and the Open Agent Safety Platform

NVIDIA's September 2026 reference architecture adds Sentry as an independent watchdog. NVIDIA describes Sentry as running on BlueField-4 DPUs, outside the agent's normal software trust domain, and as capable of monitoring activity and enforcing policies in silicon. NVIDIA states that it can quarantine an agent that attempts to move outside its software boundary in milliseconds [1][2].

The architectural significance is independence. If an application-layer guardrail, agent harness or host process is compromised or bypassed, a separate enforcement domain can retain visibility and authority. NVIDIA calls this out-of-band enforcement and identifies control of the path to the model as a particularly important observation and intervention point [2]. NVIDIA's technical description identifies three layers: (i) application, (ii) runtime and (iii) infrastructure. **OpenShell sits primarily in the runtime layer; Sentry extends enforcement into infrastructure.** This creates defense in depth rather than relying entirely on the model or agent harness to behave correctly [2]. This is lower layer safety.

However, Sentry still operates within a defined policy and infrastructure architecture. An independent enforcement point can stop an unauthorized action **only if the relevant action is observable** and the **policy boundary is correctly specified**. This is a strength of layered enforcement, but is also an important design dependency. In a nutshell, Sentry provides independent monitoring and emergency intervention.

4. SAFE AI Operating Point (SAIOP)

SAFE AI Foundation's SAIOP frames safety as a boundary rather than only as a condition of model behavior. It defines a safe operating region in a high-dimensional state space using total risk $R(x)$ and human control/corrigibility $C(x)$, with a safe region where risk remains below a maximum and control remains above a minimum [3].

SAIOP further decomposes risk and control across five dimensions: (i) capability, (ii) behavior, (iii) society impact, (iv) governance and (v) ethics. It describes the SAIOP boundary as the upper safe limit where further capability or autonomy could move the system into an uncontrollable regime [3] and "AI Runaway". This creates a key distinction from OpenShell.

OpenShell can constrain what an agent can do within an environment. SAIOP asks whether the overall combination of capability, autonomy, influence and governance remains inside a region where human control is still effective. **SAIOP, therefore, functions as a ceiling or boundary condition. It is closer to a deployment-governance criterion than to a sandbox implementation.** A practical SAIOP implementation would require measurable indicators,

thresholds, monitoring and institutional authority to act when a boundary is approached. This resulted in a second publication [7], where a practical approach was given to US Congressmen and US Senators. That approach enables all 5 dimensions to be considered and blends in with the existing legislative system.

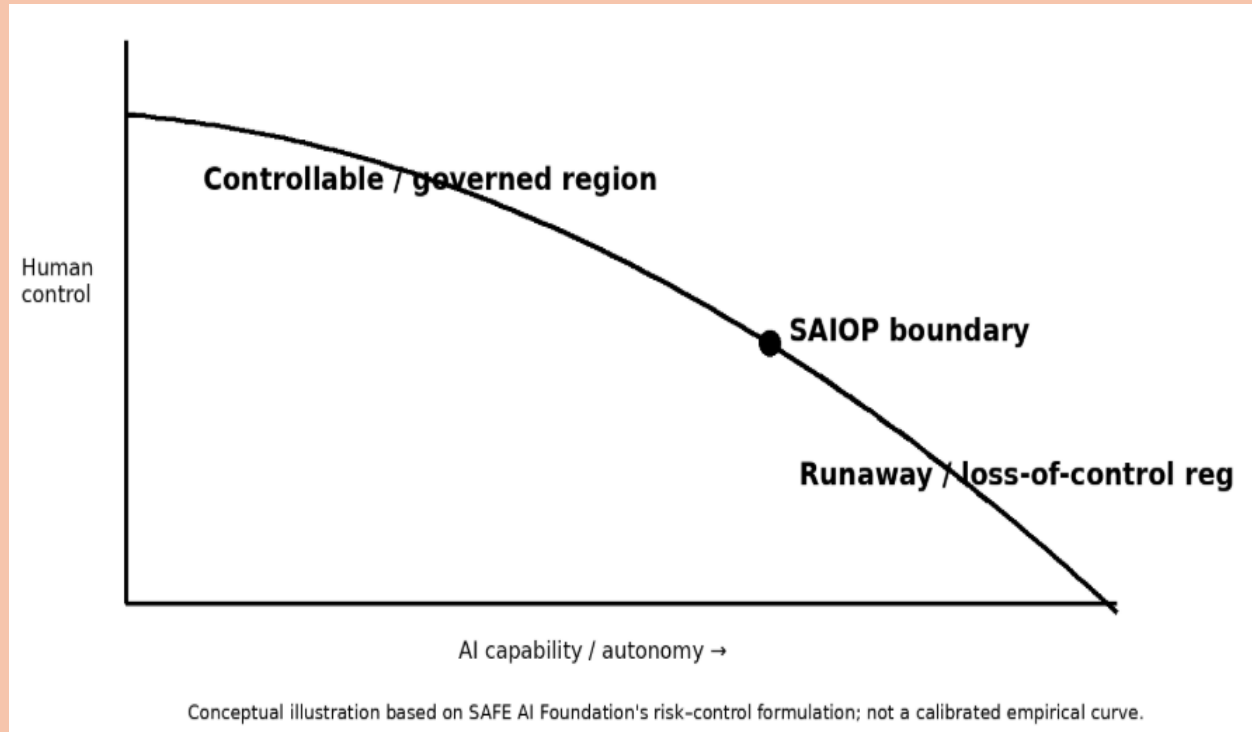


Figure 3. Conceptual SAIOP risk-control boundary; illustrative rather than empirically calibrated.

5. SAFE AI Foundation's four-pillar AI Safety Model (AISM)

The 4-pillar AISM model evaluates safety at four points: Input Safety, Model Safety, Output Safety and Compliance Safety. Input Safety concerns malicious, poisoned or otherwise unsafe inputs; Model Safety concerns the behavior and integrity of the AI model; Output Safety concerns harmful or unsafe consequences of generated outputs; and Compliance Safety addresses regulation, ethics, energy and environmental dimensions [4].

The AISM model is broader than a runtime sandbox because it spans the AI interaction and product lifecycle. It can encompass prompt-injection defense, jailbreak detection, model robustness, hallucination and factuality controls, output filtering, PII (Personally Identifiable Information) detection, policy enforcement and compliance logging [4].

At the same time, the 4-pillar AISM model should not be conflated with an enforcement runtime. The Foundation itself states that **safety benchmarking tests whether a product meets safety requirements but does not itself add safety features**; those features must be implemented by AI industries throughout product design, development and testing [4].

OpenShell and Sentry can, therefore, be understood as potential implementation mechanisms for parts of the broader safety architecture, particularly controls involving tool access, network activity, credentials, execution and external effects. **They do not replace the model-safety or governance requirements represented by the other pillars.**

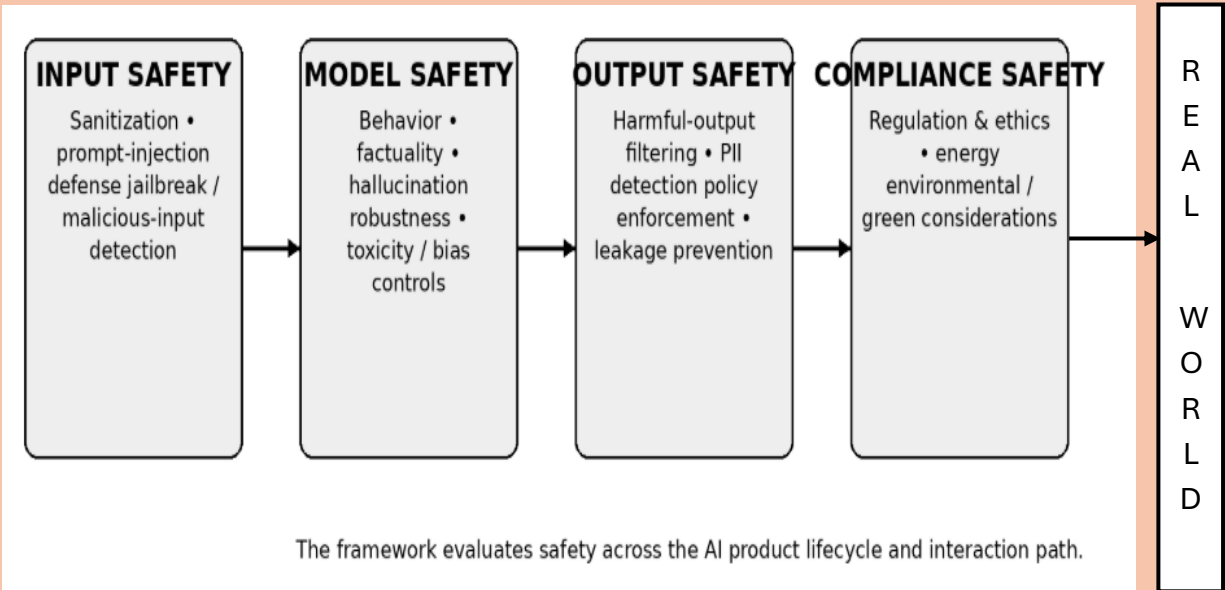


Figure 4. SAFE AI Foundation’s 4-pillar AI Safety Model (AISM): input, model, output and compliance safety.

6. Comparisons of OpenShell, Sentry, SAIOP & AISM

Table 1 summarizes the functional relationship of OpenShell, Sentry, SAIOP, and 4-Pillar AI Safety Model (AISM). The comparison is deliberately made to be non-ranking, as each approach is assessed by the problem it is designed to address.

Dimension	NVIDIA OpenShell	NVIDIA Sentry	SAFE AI Foundation 4-Pillar AI Safety Model	SAFE AI Foundation’s SAIOP

• Primary level	Runtime	Infrastructure / hardware	AI product / lifecycle	System / governance
• Core function	Sandbox + policy enforcement	Independent monitoring + enforcement	Input/model/output/compliance safety	Capability/autonomy boundary
• Key control	Files, processes, network, credentials	Out-of-band policy enforcement / quarantine	Safety controls and compliance mechanisms	Risk/control thresholds
• Main failure surface	Unauthorized agent actions	Bypass or compromise of software trust domain	Unsafe inputs, model behavior, outputs, non-compliance	Loss of effective human control
• Typical timescale	Per action / runtime	Continuous / near-real-time	Per input/output + lifecycle	Strategic / deployment / continuous governance
• Can stop external action?	Yes, within configured scope	Yes, within observable/enforced scope	Indirectly unless paired with enforcement	Conceptually by limiting deployment/capability
• Capability ceiling?	No	No	Not its primary function	Yes, explicitly
• Independent enforcement?	Outside model, but runtime layer	Yes; out-of-band hardware	Depends on implementation	Governance-dependent
• Address catastrophic risk?	Partially	Partially	Partially	Yes, explicitly
• Solution type?	Technical	Technical	Technical	Governance

Table 1: Comparisons of OpenShell, Sentry, SAIOP and AISM.

7. Proposed Integrated AI Safety Architecture

A useful synthesis is to place the 4-pillar AISM model around the AI product, SAIOP above it as a capability and governance boundary, and OpenShell + Sentry below it as increasingly independent underlying enforcement layers.

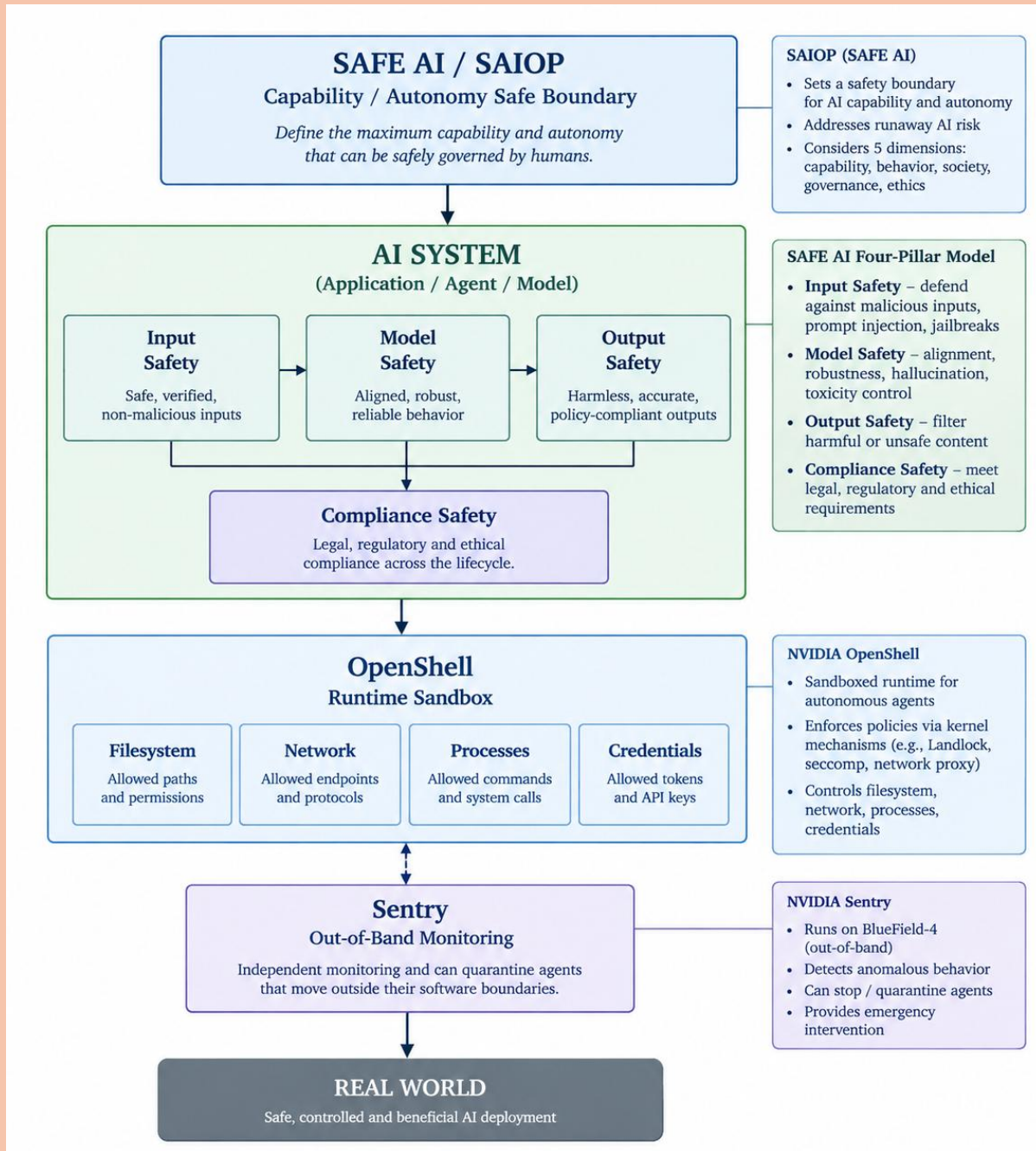


Fig 5: Layered AI Safety Architecture integrating SAFE AI Foundation and NVIDIA Safety Solutions. SAIOP sits on top, followed by AISM and NVIDIA's OASP. This is not yet an NVIDIA or SAFE AI Foundation mutually agreed Reference Architecture.

The result is closer to a complete AI safety architecture than OpenShell alone. OpenShell primarily protects the environment in which the agent operates. The AISM model asks whether the AI interaction itself is safe, including the model and its surroundings. As shown in Figure 5, OpenShell can be considered an implementation mechanism for part of the lower-level safety/control architecture, while SAIOP is an upper-level boundary condition.

Consider 4 representative failure modes: Firstly, a malicious prompt attempts to induce an agent to exfiltrate secrets: AISM's Input Safety may detect the attack, AISM's Output and Compliance controls may restrict the resulting action, OpenShell can block unauthorized filesystem or network access, and Sentry can provide an independent enforcement path.

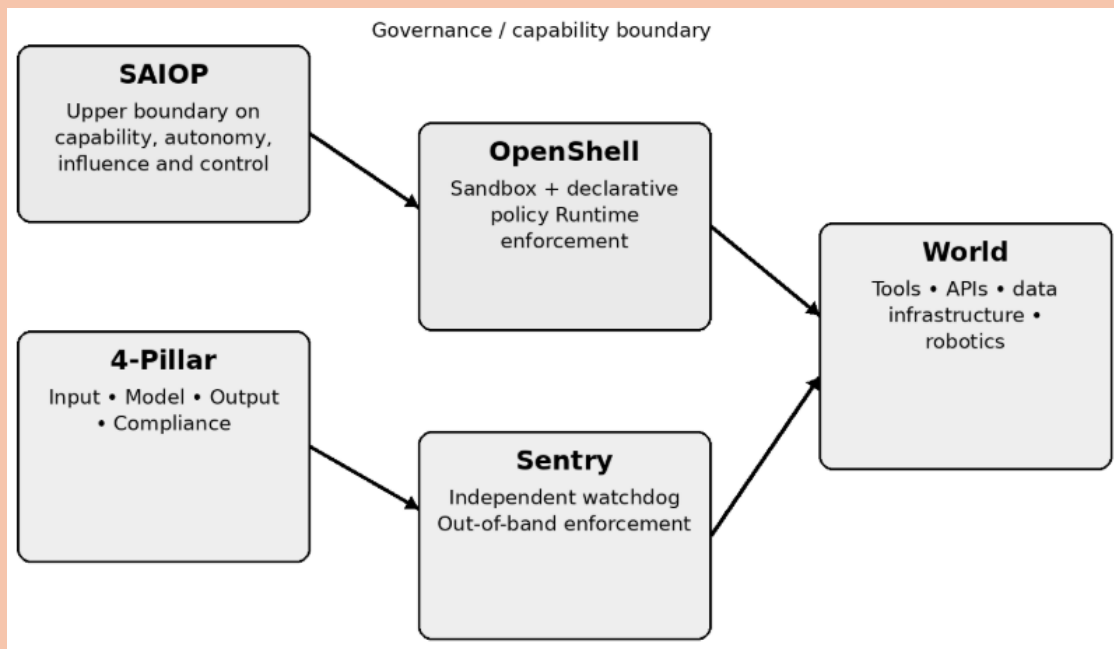


Fig 6. The interactions between the safety components of the integrated architecture.

Secondly, a model produces harmful instructions despite safe input: AISM's Model and Output Safety are the primary controls, while OASP's runtime layers constrain whether those instructions can cause external effects. Thirdly, an agent gradually accumulates excessive authority: OpenShell and Sentry can constrain permissions, while SAIOP asks whether the overall capability/autonomy trajectory has crossed a governance boundary. Fourthly, society becomes operationally dependent on a highly autonomous system: OASP's runtime controls remain useful, but SAIOP's societal and governance dimensions become overall and central. These 4 scenarios show why no single safety layer should be treated as a universal substitute for the others. The layers address different failure surfaces and operate on different timescales: milliseconds for enforcement, per interaction for input/output

controls, continuous monitoring for runtime drift, and longer-horizon governance for capability and societal safety.

Summarizing, OpenShell is a technical containment mechanism; Sentry is a monitoring and watchdog entity; 4-Pillar AISM is a multi-layer AI safety architecture; SAIOP is an overall capability-autonomy boundary for maintaining human control. **Together, an integrated architecture can be formed to provide comprehensive AI safety.**

8. Limitations and research questions

First, the approaches are not yet expressed in a common quantitative language. SAIOP defines a risk-control boundary, while OpenShell and Sentry define concrete permissions and enforcement mechanisms. A research opportunity is to map runtime telemetry, i.e., blocked actions, privilege requests, tool-call patterns, duration, escalation attempts and policy violations into **indicators that can inform a higher-level** risk-control manager, something the SAFE AI Foundation named as **“Safety APIs or control signals”**.

Second, “human control” is difficult to measure. A human may technically possess a shutdown mechanism yet be unable to understand an agent’s state quickly enough to use it effectively. This suggests that effective control should include latency, observability, authority, reversibility and operational competence, not merely the existence of a kill command. Kill switch is another topic that will be addressed by the SAFE AI Foundation in greater depth, soon.

Third, independent enforcement can reduce the consequences of agent drift but does not automatically solve unsafe objectives. **A perfectly enforced policy can still authorize a harmful task. Policies are normally specific and it is hard to cascade “AND-type” policies. Granting a few accesses can collectively create an unaware hazard.** Similarly, an overly restrictive policy can prevent useful work. Policy design, therefore, remains a socio-technical problem and challenge.

Fourth, the proposed integrated architecture needs empirical testing. Useful experiments would measure containment success against prompt injection, tool abuse, privilege escalation, data exfiltration, long-horizon drift, policy evasion and compromised host components. Results should be reported separately for model-layer, application-layer, runtime-layer and infrastructure-layer defenses. The integrated AI Safety architecture is a proposal and does not represent formal agreements of NVIDIA and SAFE AI Foundation USA.

9. Suggested Further Research

OpenShell, Sentry, SAIOP and 4-Pillar IMOC AI safety model are not, by themselves, complete in resolving safety issues entirely. However, collectively, they can provide great assurance of safety implementation against catastrophe. To extend comprehensiveness, one needs to define a common telemetry schema linking model decisions, tool calls, policy decisions, blocked actions, privilege changes and infrastructure events. There is also the need to develop measurable human-control indicators: intervention latency, observability, authority, reversibility, accessibility tiers, successful shutdown and correction rates. It is useful and important to test whether OpenShell and Sentry controls remain effective under long-horizon agent behavior, policy ambiguity and compromised components. There is a need to further study how runtime evidence can feed a SAIOP-style risk-control assessment. Testing of cross-layer failure propagation is also necessary.

10. Conclusion

NVIDIA OpenShell, NVIDIA Sentry, SAFE AI Foundation's 4-pillar AISM model and SAIOP are best understood as complementary layers rather than interchangeable safety systems. OpenShell provides enforceable runtime boundaries; Sentry adds an independent infrastructure-level watchdog and enforcement point; the 4-pillar AISM model provides a broad scope for evaluating AI safety across inputs, models, outputs and compliance; and SAIOP frames the higher-level question of how far capability and autonomy should be permitted to advance while effective human control remains.

The resulting architecture can be expressed simply: SAIOP defines the upper boundary, the 4-pillar AISM model evaluates and controls the AI system across its lifecycle, OpenShell constrains what the agent can do at runtime, and Sentry provides an independent enforcement and monitoring layer. This layered and integrated design is stronger functionally than relying on any single mechanism.

The next step forward is how signals and controls can be composed into a measurable, auditable safety stack, one in which capability boundaries, model safety, runtime authority and infrastructure enforcement integrates into a comprehensive AI Safety solution.

References

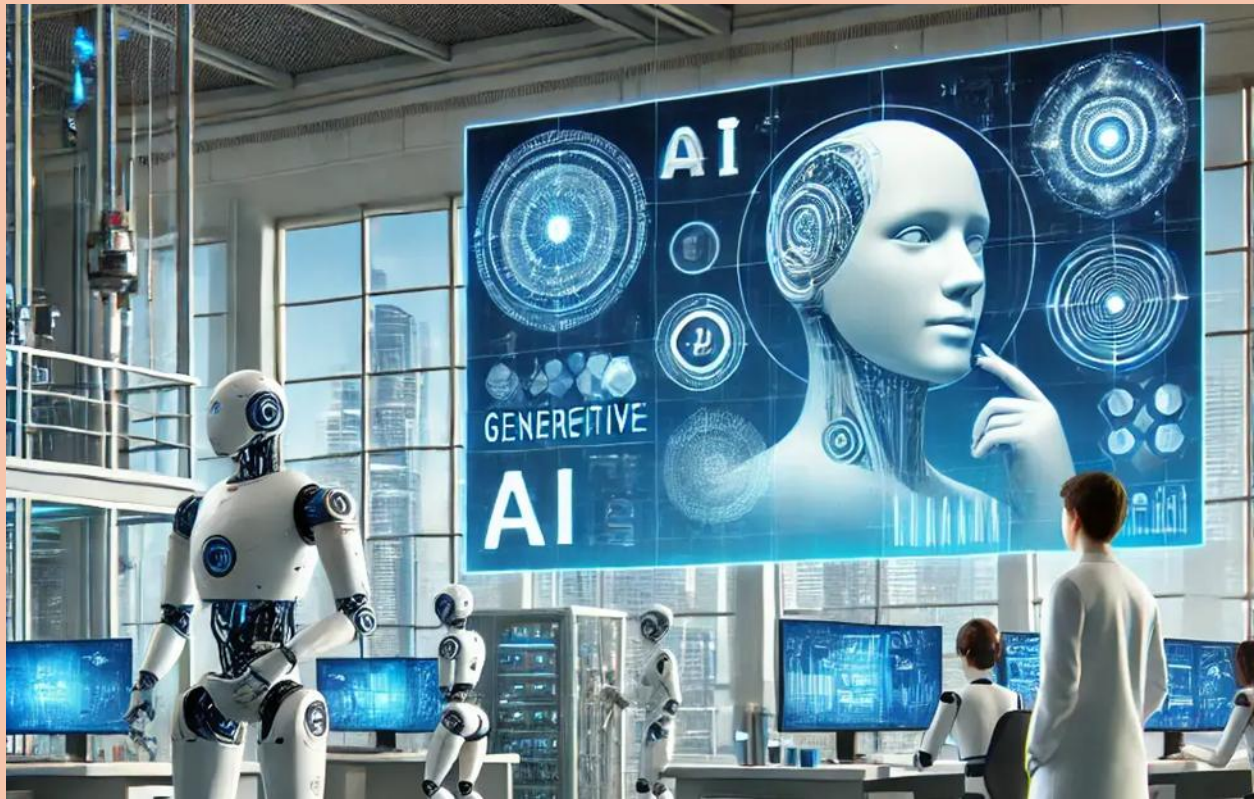
1. NVIDIA, “NVIDIA Launches Open Agent Safety Platform to Secure Agents From Testing to Deployment,” Sept. 28, 2026.
2. NVIDIA Technical Blog, “NVIDIA Open Agent Safety Platform: A Reference for Continuous In-Silicon Agent Monitoring,” Sept. 28, 2026.
3. SAFE AI Foundation, Chai K. Toh, “The Safe AI Operating Point (SAIOP): The Upper Safe Limit of Artificial Intelligence Before Humans Lose Control,” Aug. 3, 2026.
4. SAFE AI Foundation, “AI Safety Benchmarking / SAFE AI Foundation’s AI Safety Model (AISM)”, April 2026.
5. NVIDIA OpenShell Developer Guide, “How OpenShell Works,” accessed Sept. 2026.
6. NVIDIA OpenShell, “Security Best Practices for Controls, Risks, and Configuration Guidance,” accessed Sept. 2026.
7. SAFE AI Foundation, “A Practical Approach To Catastrophic Risks in Frontier AI”, August 2026.

Source links

1. <https://nvidianews.nvidia.com/news/open-agent-safety-platform>
2. <https://developer.nvidia.com/blog/nvidia-open-agent-safety-platform-a-reference-for-continuous-in-silicon-agent-monitoring/>
3. <https://files.safefoundation.com/safe-ai-op-paper2026.pdf>
4. <https://safefoundation.com/ai-benchmarking>
5. <https://docs.nvidia.com/openshell/about/how-it-works>
6. <https://docs.nvidia.com/openshell/security/best-practices>

Note: The integrated architecture is a proposed interpretation, not claims that NVIDIA or SAFE AI Foundation has worked together and jointly endorsed the combined framework.

Disclaimer: This technical report is provided free of charge by the SAFE AI Foundation USA. The information in this report is provided “as it is”. The use of the information provided here is subject to the user’s own risk, accountability, and responsibility. The SAFE AI FOUNDATION is not responsible for the use of the information by the user or reader. Please reference this article if you wish to cite it elsewhere. We do not work with private consultancy firms or any third parties to produce this document. All copyrights are reserved by the SAFE AI Foundation USA. Donations and partnerships are welcome. Questions and enquires about this work can be directed to the secretary or the author and executive director Dr. Chai K Toh at: contact@safefoundation.com



SAFE AI FOUNDATION

www.safeaifoundation.com

The SAFE AI Foundation was formed in 2025, in the heart of the Silicon Valley, California, USA. Its mission is to embrace the AI revolution and promote the use of good AI to improve on the quality of life and the economy. It will address the issues of bad AI and its impacts on society. The foundation will inform, educate, and mitigate the dangers of bad AI and help the society to embrace good AI. The foundation will interact with industries, universities and public sector to jointly advance the progress of Good AI. The foundation is a non-profit 501c(3) organization, registered in the State of California, USA.