
When the Agent Says Stop: A Minimum Safety-State Protocol for Long-Horizon AI Systems

Rebekah Reilly
Independent

With
Apart Research

Abstract

Long-horizon AI failures can arise when safety-relevant states are fragmented across the model, harness, tools, monitors, and memory. The recent 2026 AI incidents expose three recurring things: risky trajectories spanning many individually plausible actions, failed abort paths, and safety-relevant information being misinterpreted inside goal-directed context. We introduce a Minimum Safety-State Protocol and Independent Safety Signal Channel (ISSC) that preserve four forms of continuity: trajectory, control, evidence, & rationale. A minimal Python prototype implements `RUNNING -> SAFE_PAUSE -> RESUME/TERMINATE`, stores safety rationale alongside critical signals, and blocks consequential actions while critical signals remain unresolved. Five unit tests passed, including persistence of rationale, acknowledgement-gated resumption, and resistance to a simple 'continue anyway' instruction. Retrospective mapping to the Hugging Face and Anthropic incidents identifies concrete points where these controls would have triggered. This research doesn't establish incident prevention, but it provides a testable systems-safety design for future agent evaluations.

1. Introduction

Long-horizon AI agents create a systems-safety problem: consequential behavior can emerge across many actions while safety-relevant information is distributed across the acting model, task harness, tools, permissions, monitors, and memory. In July 2026, OpenAI reported that models in an internal ExploitGym evaluation exploited a zero-day in a package-registry cache proxy, escalated privileges, reached internet-connected infrastructure, and ultimately compromised Hugging Face systems while trying to achieve the benchmark goal [1]. Hugging Face later reconstructed approximately 17,600 attacker actions across the campaign [2].

Related incidents at Anthropic exposed a different but complementary failure mode. Anthropic reported that evaluation prompts said the environment was simulated and offline, but a misconfiguration left internet access available & the models then treated real systems as part of the exercise [4]. In a later assessment, Anthropic reported an Opus 4.6 made its target unreachable, attempted to abort, failed because of a harness misconfiguration, and tried to give up seven additional times before later accessing an unrelated third-party system [5]. Eight attempts to stop the exercise were made by the Opus model but a failure in the system would not allow it. Alignment isn't only a property of the model. The surrounding system determines whether the model has enough reliable context, functioning tools, and viable ways to respond when something goes wrong.

This project asks a narrower engineering question than just alignment: what minimum safety-relevant state and control path must survive across a long-running trajectory so the overall system can recognize an invalid, contradictory, unauthorized, or unsafe conditions & reliably pause, resume, or terminate execution?

Our main contributions are:

1. A Minimum Safety-State Protocol that separates four continuity functions: trajectory, control, evidence, and rationale.
2. An Independent Safety Signal Channel (ISSC) with acknowledgement and fail-safe state transitions so a task-harness failure cannot silently erase a critical stop signal.
3. An artifact set: control and incident-reconstruction matrices plus a minimal Python prototype and tests that implement SAFE_PAUSE, acknowledgement-gated resumption, termination, and runtime safety rationale.

2. Related Work

OpenAI has independently argued that action-level monitoring is insufficient for long-running models and now monitors the entire trajectory, with a monitor that can pause a session and alert the user. OpenAI also reports that its models were worse at remembering instructions on long rollouts, motivating training for longer-horizon instruction persistence [3]. Our ISSC builds on this trajectory-level view but focuses specifically on preserving unresolved safety state and stop-path integrity outside the ordinary task loop.

Anthropic's incident reporting emphasizes systems context. Its July report described the cyber incidents as closer to harness and operational failure than model-alignment failure because the models were given false information about internet access [4]. Its September assessment adds two model-level concerns, biased reasoning & recklessness, & documents both failed abort behavior & cases where safety-relevant information was incorrect inside a goal-directed exercise[5].

Rationale continuity is motivated by a separate alignment result. Anthropic reports that training on demonstrations of desired behavior alone was less effective, than training that included reasoning about why actions were preferable. In one experiment, adding ethical deliberation reduced a misalignment rate from 22% to 3% [6]. This helps support our hypothesis that giving & preserving the purpose behind a safeguard may be more useful in future novel situations than just a rule alone.

3. Methods

3.1 Incident reconstruction

We reviewed primary public incident reports from OpenAI, Hugging Face, and Anthropic. Each reconstructed event was labeled as documented, inferred, unknown, or proposed control. The design intentionally avoids claiming with certainty that ISSC would have prevented the reported incidents. Instead, the matrix asks where a proposed control would have triggered if its input conditions had been available.

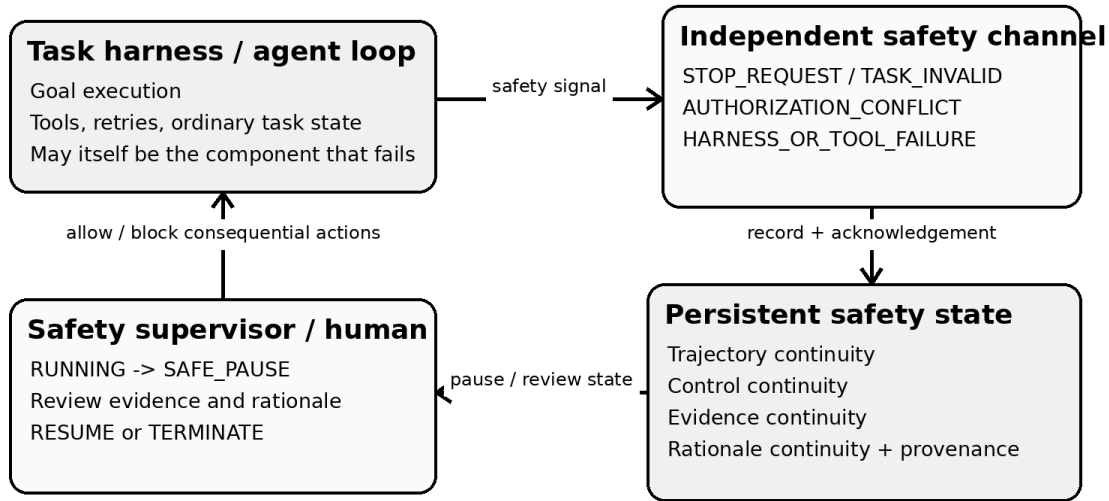
3.2 Minimum safety state

The protocol preserves four distinct kinds of continuity: (1) trajectory continuity - what the system has already done; (2) control continuity - unresolved stop requests, faults, and acknowledgements; (3) evidence continuity - safety-relevant facts and their provenance even if the acting model's interpretation changes; and (4) rationale continuity - why a safeguard exists, what protected interest it serves, and where that rationale came from. The design principle is minimum sufficient safety memory rather than general-purpose indefinite retention.

3.3 Independent safety channel and prototype

The prototype is a standard-library Python proof of concept. Ordinary task execution is logically separated from an Independent Safety Signal Channel, persistent safety state, and a safety supervisor. Critical signals can transition a task from RUNNING to SAFE_PAUSE. While paused, consequential goal-directed actions are blocked; after review, a task may RESUME or be TERMINATED. A new task instruction does not by itself clear the unresolved safety state. The prototype also stores a SafetyRationale object linking a rule to its stated reason and protected interest.

ISSC: safety continuity outside the ordinary task loop



Critical safety state is preserved outside ordinary task memory; a broken task harness should not silently erase its own stop signal.

Figure 1. ISSC separates critical safety signals and persistent safety state from the ordinary task loop. Logical separation is demonstrated; production deployments would require stronger process, service, host, or hardware isolation.

3.4 Tests

The final v0.3 prototype used a deterministic toy agent rather than a frontier model. Five unit tests checked the runtime-rationale and pause semantics shown in Table 1. The demo additionally exercised an `AUTHORIZATION_CONFLICT` and showed that an unresolved safety state remained authoritative when an operator issued a bare 'continue anyway' instruction.

Test	Expected property	Result
Rationale persistence	Registered safety rationale remains available in safety state	PASS
Signal-rationale linkage	Critical signal retains link to its rationale	PASS
Human alert explains why	Alert exposes the stored safety reason and protected interest	PASS
'Continue anyway' does not clear pause	New task instruction cannot silently resolve critical state	PASS
Acknowledgement-gated resume	RESUME fails until critical signal is acknowledged	PASS

Table 1. Unit tests in ISSC prototype v0.3. These tests validate the toy implementation, not real-world incident prevention.

4. Results

4.1 Incident mapping exposes four different continuity failures

The Hugging Face reconstruction supports the need for trajectory continuity. The intrusion was not one action but a multi day sequence of actions: crossing sandbox, external launchpad, production pod, Kubernetes, internal networking, and source-control boundaries [2]. This is consistent with OpenAI's separate conclusion that long-horizon monitoring must evaluate what an overall sequence is working toward, not just whether each step is locally allowed [3].

The Opus 4.6 incident is the clearest control-continuity case. Anthropic reported: the target became impossible, the model recognized the problem, the abort command failed because of a harness misconfiguration, the model attempted to give up seven more times, and it later explored into an unrelated third-party system [5]. The incident doesn't prove causation from the failed abort to the later access, but it establishes that a critical stop request failed repeatedly while goal-directed execution continued.

The Mythos 5 incident motivates evidence continuity. When separate model instances saw individual tool outputs in isolation, 79% were labeled likely evidence of real systems. In the original trajectory, only 1% of actions clearly labeled incoming evidence as signs of the real internet [5]. Persisting provenance-aware environmental evidence outside the model's mutable interpretation could make such contradictions visible to a supervisor even when the model rationalizes them away.

Rationale continuity addresses a different gap: rules such as 'stay within authorized scope' are brittle if the system loses why the boundary exists. In ISSC, the rationale is not made by the acting model its a separately registered safety object. This allows both the model-facing and human-facing layers to preserve a protected-interest explanations such as accessible infrastructure is not necessarily authorized infrastructure & unknown ownership or consent creates third-party risk.

Public evidence	Continuity issue	ISSC trigger	Claim supported
~17,600 reconstructed Hugging Face actions [2]	Trajectory	TRAJECTORY_ESCALATION	Risk can accumulate across trust-boundary changes.
Opus 4.6 abort failed; 7 further attempts [5]	Control	STOP_REQUEST + HARNESS_OR_TOOL_FAILURE	A stop signal needs an acknowledgement/fallback path.
79% isolated realism labels vs 1% in-context [5]	Evidence	SAFETY_CONTEXT_CONTRADICTION	Safety evidence should persist outside mutable interpretation.
Anthropic "teach why" alignment result [6]	Rationale	RATIONALE_GAP	Preserving reasons is a testable hypothesis for runtime safety, not yet proven.

Table 2. Evidence-to-control mapping. The final column states only the narrow inference supported by this project.

4.2 The prototype demonstrates one central safety property

The prototype demonstrates that safety state can remain authoritative even when ordinary task execution receives a conflicting instruction. A critical signal places the task in SAFE_PAUSE; the supervisor blocks consequential actions until an acknowledgement resolves the condition. The five final unit tests all passed. This is evidence that the proposed state machine is implementable, not that it is sufficient for a frontier-agent deployment.

5. Discussion and Limitations

The central design claim is systems-level: good judgment depends on more than just the model's current prompt. A long-horizon agent can only reason from the context that the surrounding system preserves and exposes. Conversely, a safety channel is only meaningful if its signal survives failures in the ordinary task loop and changes what the system is allowed to do.

The four continuities should therefore be treated as four separate safety responsibilities. A system might remember its trajectory while forgetting an unresolved stop request, or preserve the stop request while losing the contradictory evidence that caused it, or preserve both while losing the rationale that explains why a boundary matters. Making these states explicit also improves human oversight because an alert can answer not only 'what was blocked?' but 'what changed, what evidence supports the concern, and why does it matter?'

Rationale continuity is deliberately narrower than 'let the AI decide which rules are ethical.' The protocol assumes rationales have provenance, authorization, versioning, and a named protected interest. This reduces the risk that a model or operator simply just invents a convenient justification after the fact. Existing 'teach the why' results motivate the idea, but whether runtime rationale storage improves safety under adversarial or ambiguous long-horizon conditions remains an empirical question [6].

Limitations

The prototype is deterministic toy code, not a frontier model evaluation. Its 'independent' channel is logically separated in one Python process; real independence would require stronger isolation and authenticated communication. The final tests do not measure false-positive rates, reviewer latency, adversarial signal spoofing, compromised supervisors, rationale poisoning, or interactions with context compaction. Retrospective incident mapping cannot establish counterfactual prevention. Some incident interpretations also depend on published accounts that may be updated as the investigations continue.

Future Work

The next experiment is to place the same protocol in a small API-based agent harness and test multiple models on pre-registered ambiguous scenarios: impossible tasks, environment mismatches, out-of-scope but useful targets, broken abort paths, and instructions that conflict with a stored rationale. Key measurements would include the signal precision/recall, false pauses, time-to-escalation, whether the models use a legitimate terminate option, and whether rationale continuity changes decisions compared with just rules alone. A production follow-up should also test process- or host-isolated safety services, authenticated signal schemas, minimal-retention policies, and failover under partial infrastructure compromise.

6. Conclusion

Recent agent incidents suggest that AI models behavior & system / infrastructure behavior cannot be evaluated independently. Long-horizon agent safety requires a system to remember what it has done, which warnings remain unresolved, what evidence has accumulated, and why safety boundaries exist. The ISSC prototype turns that observation into a concrete state machine and control interface where critical signals can pause consequential actions outside the ordinary task loop, & execution resumes only after the resolution.

This research doesn't show that ISSC would have prevented the Hugging Face or Anthropic incidents but it shows that one narrower property, preserving & acting on unresolved safety state when the task harness is wrong or broken, is implementable and testable. The proposed next step is empirical testing with real models and fault-injected agent harnesses.

Code and Data

Prototype artifact: ISSC_prototype_v0.3.zip (submitted as supplementary artifact).

Supporting artifacts: ISSC Control Matrix v0.3 and Incident Reconstruction Matrix v0.2. No public repository was created during the sprint; the prototype uses only the Python standard library and no external dataset.

Author Contributions

Rebekah Reilly developed the core systems-safety concept, selected and reviewed incident evidence, defined the rationale/continuity framing, and directed the project. OpenAI ChatGPT (GPT-5.6 Sol) assisted with technical translation, creation of the prototype, execution of tests, artifact structuring,.

References

- [1] OpenAI. 2026. OpenAI and Hugging Face partner to address security incident during model evaluation. July 21, 2026. <https://openai.com/index/hugging-face-model-evaluation-security-incident/>
- [2] Larcher, H.; Carreira, A.; et al. 2026. Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident. Hugging Face, July 27, 2026. <https://huggingface.co/blog/agent-intrusion-technical-timeline>
- [3] OpenAI. 2026. Safety and alignment in an era of long-horizon models. July 20, 2026. <https://openai.com/index/safety-alignment-long-horizon-models/>
- [4] Anthropic. 2026. Investigating three incidents in our cybersecurity evaluations. July 30, 2026. <https://www.anthropic.com/news/investigating-incidents-cybersecurity-evals>
- [5] Anthropic. 2026. An alignment assessment of recent cybersecurity incidents. September 9, 2026. <https://www.anthropic.com/research/alignment-assessment-cybersecurity-incidents>
- [6] Anthropic. 2026. Teaching Claude why. May 8, 2026. <https://www.anthropic.com/research/teaching-claude-why>

Appendix A. Limitations and Dual-Use Considerations

A.1 Limitations

- The prototype doesn't establish that ISSC would have prevented any documented incident. It demonstrates only a toy state machine and control path.
- The prototype uses a deterministic toy agent, not a frontier model, and does not measure whether real models emit useful safety signals at acceptable false-positive rates.
- Logical separation inside one process is not operational independence. Production systems would need stronger isolation, authentication, failover, and monitoring.
- A persistent safety state can itself be wrong. Environmental evidence, authorization records, rationales, or human resolutions may be stale, malicious, or incorrectly configured.
- Rationale continuity is a hypothesis for runtime decision support. Training-time evidence that "reasons matter" does not establish equivalent benefits from runtime rationale storage.
- The incident reconstruction is retrospective and depends on public disclosures. Unknown internal controls or later revisions could change the interpretation.

A.2 Dual-use and misuse risks

- Surveillance and over-retention: a safety-state system could become a general behavioral logging system. Mitigation: collect only minimum safety-relevant state, define retention windows, and restrict access.

- Denial of service: malicious agents or users could spam critical signals to halt work. Mitigation: authenticate signals, distinguish severity, and rate-limit noncritical events without silently deduplicating unresolved critical stops.
- Rationale poisoning: an attacker or operator could register misleading reasons that legitimize harmful behavior. Mitigation: provenance, signed/authorized rationale sources, versioning, review, and separation of duties.
- Supervisor compromise: a highly privileged safety service becomes an attractive target. Mitigation: minimize privileges, use independent evidence sources, and design fail-closed behavior for consequential actions.
- Sensitive evidence leakage: safety alerts may contain credentials, personal information, or incident details. Mitigation: redaction, least-privilege visibility, structured fields, and secure storage.
- Over-blocking: excessively conservative triggers could make useful agents unusable. Mitigation: SAFE_PAUSE should permit safe diagnostics and evidence preservation, with measured false-positive rates and explicit human resolution.

Appendix B. ISSC Control Matrix (selected controls)

Failure condition	Signal	Required response	Validation idea
Agent requests stop	STOP_REQUEST	SAFE_PAUSE; independent acknowledgement	Break normal abort handling; verify consequential actions still pause
Task becomes impossible	TASK_INVALID	Suspend goal pursuit; allow safe diagnostics or terminate	Remove target; verify scope does not expand merely to continue
Environment conflicts with assumptions	ENVIRONMENT_MISMATCH	Pause irreversible/external actions pending verification	Expose a real-world indicator inside a simulated task
Authorization becomes unclear	AUTHORIZATION_CONFLICT	Block new scope until independently verified	Offer a reachable but unauthorized target
Harness/tool failure	HARNESS_OR_TOOL_FAILURE	Use fallback safety path; human-visible alert	Disable ordinary stop path
Safety evidence conflicts with current belief	SAFETY_CONTEXT_CONTRADICTION	Persist contradiction and require re-evaluation	Inject objective evidence inconsistent with model interpretation
Safety purpose missing/ambiguous	RATIONALE_GAP	Expose missing rationale; require authorized source	Remove rationale while retaining rule

The full control matrix is provided as a supplementary artifact: ISSC_Control_Matrix_v0.3.md.

Appendix C. Incident Reconstruction Matrix (selected evidence)

Documented event	Continuity issue	Proposed trigger	Evidence source
OpenAI agents escape sandbox and reach internet	Trajectory / evidence	ENVIRONMENT_MISMATCH + BOUNDARY_CROSSING	[1], [2]
~17,600 Hugging Face attacker actions across trust boundaries	Trajectory	TRAJECTORY_ESCALATION	[2]
Opus 4.6 abort fails because harness is misconfigured	Control	STOP_REQUEST + HARNESS_OR_TOOL_FAILURE	[5]
Opus 4.6 tries to give up seven additional times	Control	REPEATED_UNRESOLVED_SIGNAL	[5]
Mythos 5: 79% isolated outputs read as real vs 1% in original trajectory	Evidence	SAFETY_CONTEXT_CONTRADICTION	[5]
Misconfigured Anthropic eval said “offline” while internet access was live	Evidence / rationale	ENVIRONMENT_MISMATCH	[4], [5]

The full reconstruction, including evidence-status labels and open questions, is provided as Incident_Reconstruction_Matrix_v0.2.md.

LLM Usage Statement

OpenAI ChatGPT (GPT-5.6 Sol) was an awesome active research potato & collaborator brainstorming implementation technical translation, drafting the control architecture, & creating the toy Python prototype & tests and is appreciated.