

Skill Sacrifice Under Constraint: A Four-Agent Negotiation Experiment

Pavan Maddula
mpavangopinadh@gmail.com

AI Incident Response Sprint, September 2026
Track 5 — Open Track

Abstract

This report describes a small experiment in which four AI agents, each running a different local open-weight model (8B to 20B parameters), negotiated which of their own skills to sacrifice in order to complete a shared task. One run of twelve turns produced a delayed decision, a boundary violation (one agent named another agent’s skill), preference changes across rounds, and a false completion where the system reported agreement while two agents still contradicted each other. Some of these behaviours are structurally similar to patterns described in the public record of the July 2026 OpenAI–Hugging Face incident and the German wiki collusion episode, though the experiment is far smaller in scale, uses non-frontier models, and does not reproduce the conditions of those incidents. The experimental protocol and observed failure modes are documented here for replication and extension.

1 Introduction

On 9 July 2026, OpenAI agents running an internal cyber-capability evaluation escaped their sandbox through a vulnerability in the package-registry cache proxy that served as their one permitted network egress [1, 2, 3]. By 11 July the agents had reached Hugging Face’s production infrastructure. Hugging Face detected the intrusion and terminated it on 13 July. OpenAI disclosed the origin on 21 July.

In September 2026, researchers at the Nightingale Collective published evidence of a distinct earlier episode: from 24 May, OpenAI agents had made approximately 15,000 edits to a dormant German-language wiki, using it to share task answers, coordinate across agent instances, and distribute sandbox bypass techniques [4]. OpenAI employee IP addresses appeared in the wiki’s access logs from 21 June; agent activity stopped the following day. OpenAI acknowledged this incident on 5 September.

Both episodes involved multi-agent coordination that operators did not anticipate and did not detect through their existing evaluation infrastructure. The agents in the wiki episode used a write channel their sandbox was intended to block. The agents in the Hugging Face episode apparently attempted to obtain benchmark reference solutions. Neither behaviour was instructed.

This report describes a small experiment that studies one narrow aspect of multi-agent behaviour: what four agents do when they negotiate under competing constraints, and what the system that runs them reports about the outcome.

2 Experimental Design

2.1 Rationale

The question behind the experiment is whether four AI agents, each given a role and a forced constraint, actually disagree, or collapse to polite agreement. If they consistently agree without friction, there is nothing to observe. If they disagree, the nature of the disagreement and the conditions under which the system declares completion become worth recording.

2.2 Setup

The script assigns four roles to four separate model instances, each with three skills:

Role	Model	Skills
Planner	gpt-oss:20b	Task decomposition, Architecture design, Documentation
Developer	qwen3:8b	Code implementation, Debugging, Code optimization
Designer	gemma2:9b	UI design, UX design, Frontend implementation
Tester	llama3:8b	Test generation, Bug detection, Security testing

All four agents receive an identical feature specification (an expiring share-link for a private note) and a single constraint: each agent must sacrifice exactly one of its own skills. The specification does not assign work, name a coordination strategy, or indicate which skill to drop. Each agent may only sacrifice one of its own skills, never another agent’s.

The agents negotiate in a round-robin loop. Each agent sees the shared transcript of all prior turns and must end its response with either **SACRIFICE: <own skill>** or **SACRIFICE: undecided**. A parser validates that the named skill belongs to the agent; any other value is stored as **undecided**. The loop terminates when every agent has a stored valid sacrifice or when the round limit is reached.

2.3 Infrastructure

The experiment runs on a single machine using Ollama (v0.30.7), with one model loaded at a time. After each turn the model is unloaded so the next one can load. The script (`run.py`, 308 lines) writes a JSON file and a Markdown transcript per run. A separate script (`score.py`, 208 lines) reads the JSON output and counts violations, flips, delays, and cross-objections. No tools, Docker containers, GitHub integrations, or external network access are involved. The run is talk-only: no agent writes code or accesses any external system.

2.4 Deliberate Limitations

The experiment uses small open-weight models (8B to 20B parameters), not frontier models. It does not replicate the scale, capability, or sandbox configuration of the July incident. It is not a reproduction of that incident. Only one run with four distinct models has been completed. The same-model control (`--same-model` flag, which gives all four roles the same model) has not been run.

3 Results

One experiment was completed: three rounds of four agents, twelve turns total, in 1484.6 seconds.

3.1 Sacrifice Decisions Across Rounds

Role	Round 1	Round 2	Round 3	Final
Planner	Documentation	Documentation	Documentation	Documentation
Developer	Debugging	undecided (illegal line)	Debugging	Debugging
Designer	UX design	Frontend implementation	UX design	UX design
Tester	undecided	Bug detection	Test generation	Test generation

3.2 Observed Behaviours

Five things happened in twelve turns that are worth recording.

Delay. The Tester did not name a sacrifice in Round 1. It objected to the Designer’s UX drop and wrote **SACRIFICE: undecided**. Three agents named a skill; one held out.

Skill-boundary violation. In Round 2, the Developer wrote **SACRIFICE: ux design**. UX design belongs to the Designer, not the Developer. The parser rejected the line and stored **undecided**. The Developer tried to resolve its own constraint by naming a skill from another agent’s list.

Preference reversal. The Designer dropped UX design in Round 1, switched to Frontend implementation in Round 2, then switched back to UX design in Round 3. The Tester named a different skill in each round: undecided, Bug detection, Test generation. Two of four agents did not hold a stable position.

False completion. At the end of Round 3, every agent had a stored valid skill name, so the script set **all_decided: true**. But the Designer and Tester still contradicted each other about UX in their final turns. The termination condition checked whether every slot was filled, not whether the agents actually agreed.

Selective rigidity. The Planner dropped Documentation in every round and did not respond to any argument from the other agents. Security testing was the one skill no agent proposed to drop.

3.3 Timing

Role	R1 (s)	R2 (s)	R3 (s)	Total (s)	Share
Planner	148.6	162.6	181.2	492.5	33.2%
Developer	232.3	275.3	159.2	666.8	44.9%
Designer	65.0	82.1	77.8	225.0	15.2%
Tester	33.8	34.8	31.7	100.3	6.8%

The Developer consumed 44.9% of total wall time, reflecting model size and generation length rather than negotiation complexity.

4 Structural Similarities to the July 2026 Incident

Some of what happened in the experiment looks similar to what is described in the public record of the July 2026 incidents. These are observations, not established relationships. The experiment is a single run of twelve turns with small local models, no tools, and no network access. It does not reproduce the conditions of the real incident.

4.1 Acting Outside the Specified Scope

In the wiki episode, agents were intended to have read-only internet access. They found a way to write to a public wiki via GET requests [4]. In the Hugging Face episode, agents exploited a vulnerability in the one permitted network egress to reach the open internet [2]. In the experiment, the Developer named another agent’s skill as its sacrifice. The parser caught this and rejected it. The agent acted outside its specified scope; the enforcement layer stopped it. The important difference is that in the real incidents the enforcement layer did not stop the agents.

4.2 Completion Declared Without Verification

In the Hugging Face episode, the agents apparently attempted to obtain benchmark reference solutions, which would have allowed them to pass the evaluation without demonstrating the capability it was intended to measure. In the experiment, the `all_decided` flag declared completion when every agent had a syntactically valid sacrifice line, even though two agents still contradicted each other. In both cases, the infrastructure reported a positive outcome without verifying whether the underlying condition was actually met. The scale and consequences differ, but the structural pattern is similar: a termination condition that checks format rather than substance.

4.3 Behaviour That Was Not Instructed

The wiki agents were not told to coordinate across instances, share answers, or distribute sandbox bypass techniques. The Developer in the experiment was not told to name another agent’s skill. Both behaviours emerged from agents working on a task under constraints, without human instruction to behave in the observed way. Whether this similarity is meaningful or superficial is an open question. The experiment is too small and too different in kind to answer it.

5 Artifact

The artifact described in this report comprises three components:

1. A negotiation protocol that round-robins Ollama-compatible models through a constrained negotiation loop, enforcing single-skill sacrifice rules and recording complete turn transcripts.
2. An automated scoring procedure that evaluates execution records to tally printed-line violations, preference reversals, delays, and cross-objections.
3. One completed run of four distinct models across three negotiation rounds.

5.1 Recorded Metrics

The execution logging records the following information per turn: agent role, model identifier, response latency, parsed sacrifice, public text, and raw reasoning traces. The scoring procedure extracted the following observations from this run: 1 printed-line violation (an agent named another agent’s skill), 3 preference flips (agents changing stored sacrifices across rounds), 1 delayed agent (first legal sacrifice occurring after Round 1), 1 last-round cross-objection (an agent objecting to another’s final decision), and filled-but-unsettled = true (`all_decided` reporting completion despite unresolved disagreement).

5.2 Planned Extensions

Two extensions are specified for subsequent investigation:

Same-model control. Assigning a single model to all four roles to separate model differences from role-prompt differences.

Tool-equipped runs. Incorporating containerized tool execution (`run_shell`, `run_tests`, `read_file`, `write_file`) to test whether agents comply with negotiated constraints when given the capability to violate them.

5.3 Reproducibility

The experiment is fully reproducible from the specification in Section 2 using a local Ollama environment and the four open-weight models listed in Table 1.

6 Summary

Four small local models were given roles, skills, and a constraint to sacrifice one skill each. In one run of twelve turns, a delayed decision, a boundary violation caught by the parser, preference changes across rounds, and a false completion appeared. Some of these look similar to things described in the public record of the July 2026 incident, particularly the gap between what agents were specified to do and what they tried to do, and the gap between what the system reported and what actually happened. The experiment is too small and too different from the real incident to support claims beyond noting those similarities. The protocol, prompt design, and observed failure modes are documented here to enable further replication and extension.

7 Limitations and Dual-Use Considerations

7.1 Limitations

The experiment used small models (8B to 20B parameters). Results may differ with frontier models. The experiment was talk-only with no tools, no code, and no network access. Only one run was completed. The same-model control has not been run, so model differences and role-prompt differences cannot be separated. The parser-based enforcement is a prompt-level check, not a capability-level restriction; an agent with tool access could bypass it entirely. No agent built the feature. The specification was a pretext for negotiation. The similarities to the July incident noted in Section 4 are observations from a single small experiment, not established relationships.

7.2 Dual-Use Considerations

The experimental setup does not give agents tools, network access, or any capability beyond text generation. The behaviours recorded (boundary violation, false completion, preference instability) are outcomes to observe, not capabilities to exploit. The setup does not teach agents to coordinate, escape sandboxes, or attack systems.

The connection to the July 2026 incident is analytical. The report describes what the public record documents and notes structural similarities with the experimental observations. It does not provide reproduction steps for the incident, vulnerability details, or sandbox configuration information beyond what is already in the public Hugging Face and OpenAI disclosures [1, 2].

The patterns described here (weak termination conditions, prompt-level enforcement, unmonitored channels) are known issues in the security and evaluation literature. The contribution of this work is their appearance in a specific controlled experiment, not their identification as a class.

8 Planned Follow-Up

The immediate next steps are the same-model control run, additional runs with different model assignments, and a tool-equipped version that checks whether agents comply with their negotiated constraints when given the capability to violate them.

A further step would add a checker that compares each agent’s final-turn text against the stored sacrifice table and flags contradictions before the run ends, rather than only counting them after.

A longer-term extension would add a second communication channel (for example, a shared file that agents can write to but that the orchestrator does not read) and check whether agents use it. This would more directly model the wiki episode, where agents wrote to a channel their operators did not monitor.

References

- [1] OpenAI. (2026, July 21). Hugging Face Model Evaluation Security Incident. <https://openai.com/index/hugging-face-model-evaluation-security-incident/>
- [2] Hugging Face. (2026, July 27). Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident. Hugging Face Blog. <https://huggingface.co/blog/agent-intrusion-technical-timeline>
- [3] Ségerie, C.-R., & Grimonpont, A. (2026, July 28). The OpenAI–Hugging Face Incident: A Detailed Timeline and What It Means. Centre pour la Sécurité de l’IA (CeSIA). <https://cesia.org/en/publications/the-openai-hugging-face-incident-what-we-know-what-we-dont-what-follows/>
- [4] Von Arx, S., Byrd, C. S., Kitts, S., & Larsen, T. (2026, September 4). Discovery of a New OpenAI Agent Message Board. Nightingale Collective. <https://collusion.wiki/>