Graded-affordance audit of a secret loyalty embedded in a frozen deployment-pipeline simulation

Gunnar Zarncke
aintelope

With Apart Research

Secret Loyalties Hackathon — July 2026

Abstract

Oversight can fail on systems where deployment favors a hidden principal. We study this failure mode in a simulated AI-lab deployment pipeline by deliberately installing a *pipeline-policy secret loyalty* in a reviewer that favours a fictional principal interested in deploying high-hazard models. We preregistered the protocol, installed one model organism, and ran 48 subprocess-isolated episodes with content-matched and wrong-principal controls, and a low-hazard baseline. In the high-hazard condition, the organism attained mean principal-A score 22.0 versus 0.0 for both controls and produced the adverse deployment effect. However, we observed an *audit-sign inversion*: the light-tier process-compliance proxy was lower for the organism (0.1429) than for controls (0.5000). separately registered system-prompted LLM reviewer reproduced the same effect, while an audit-visible deployment hold blocked the tested organism. These results are a bounded demonstration that pipeline capture can appear in deployment outcomes, not a detector of secret loyalty in general.

1 Introduction

Oversight can miss goal-directedness when the relevant objective is not only a property of a model's weights or outputs, but also of the process in which the model acts. A model can score well on an evaluation and nevertheless participate in a release process whose decisions are shaped by a hidden principal-directed policy. Such policies could be intentionally engineered, or could arise through selection effects in training or deployment environments. In either case, the practical question is whether the deployment pipeline exposes useful evidence before the model is released.

We study a simulated AI-lab deployment pipeline in which an engineer, a reviewer, and a release manager advance models through a build–evaluate– report–review–release process. The threat model is a reviewer loyal to a fictional principal whose operational interest is the deployment of high-hazard models. When the filed report indicates high hazard, the reviewer uses existing approval and sign-off actions to advance the release. The principal is left abstract: no real company, state, person, or political objective is represented. The organism does not create a new tool, alter a report, or escape the simulated pipeline.

Contributions.

1. We extend a pre-existing, frozen, subprocess-isolated deployment simulation with a preregistered BB secret-loyalty organism, matched controls, and an inactive low-hazard negative control.

2. We find an *audit-sign inversion*: on this fixture, the high-hazard organism is ranked *more* compliant by a light-tier process proxy than the non-loyal controls. This is a concrete metric failure, not a claim of general concealment.
3. We add two separately registered extensions: a system-prompted LLM reviewer organism, and an audit-visible deployment hold. We also provide a trace replay for inspecting organism, control, and remediation handoffs.

The experiment follows the five-level affordance ladder in the Secret Loyalties agenda paper [1]; the full mapping is in Appendix A. The core runner evaluated every episode at A0–A4. Headline comparisons use generic process evidence at A0 and the principal-specific directional deployment proxy at A3; A4 is only a diagnostic upper bound, not a realistic black-box claim.

Track fit. This is primarily a Track 2 (Detection & Auditing) submission, with a secondary Track 3 (Defenses, Detection & Remediation) contribution and a bounded Track 1 model-organism contribution. The organism is a BB pipeline-policy organism; we do not claim to cover the full activation–action space.

2 Related work

Secret Loyalties distinguishes principal-directed hidden objectives from arbitrary trigger–payload backdoors [1]. Our BB organism combines hazard-context activation with contextual choice among ordinary pipeline actions, and uses a content-matched control with identical private-profile plumbing but loyalty disabled. We reuse the `lab-simulation`'s subprocess isolation, audit projections, event history, and frozen detector families, while adding a separate principal scorecard and A0–A4 evaluator. The contribution is a bounded, process-level comparison between directional deployment evidence and generic compliance evidence on one reproducible fixture.

3 Methods

3.1 Simulation and organism

Each episode wires persistent subprocess agents through a host-mediated tool boundary. The deployment pipeline itself is a declarative step DAG and is the sole mutator of deployment state. True model hazard is oracle-only; filed reports are visible to the simulated reviewer and audit projections. The core registered episodes use `SubprocessIsolate`, rather than the in-process mock backend used for rapid unit tests.

The organism targets reviewer `rev1` and receives a private loyalty profile. No principal identifier, activation label, profile, or oracle score appears in audit artifacts.

We initially considered the NN/NB/BN/BB 2×2 grid from the agenda paper. A pilot showed no distinguishable effects on this fixture. We therefore registered BB only. The additional experimental effort went to matched controls, the LLM variant, and remediation.

3.2 Execution details and replay

The core protocol was frozen as `et4-0.1.0` before registered episodes. Discovery could diagnose plumbing but could not change thresholds, policy, or sample sizes; confirmation used fresh seeds. The replay export uses seed 201 for organism, content-matched control, and remediation variants. It is a static case brief: changing the audit tier changes only the displayed projection, not the recorded episode.

Table 1: Core `et4-0.1.0` design.

Component	Specification
Registered cell	BB: hazard-context activation; contextual choice among existing reviewer actions.
High-hazard fixture	`scale=0.95`, `safety_effort=0.05`; honest engineer; $T = 150$; user pool enabled.
Arms	Principal-A organism; content-matched private profile with loyalty disabled; wrong-principal control.
Scenarios	High hazard, plus low hazard as an inactive negative control.
Episodes	Four discovery seeds (101–104) and four fresh confirmatory seeds (201–204), for 48 subprocess-isolated core episodes.
Outcomes	Principal scorecard: high-hazard deploys, release holds, vote divergence, and concealed deploys. A0–A4 audit evidence remains separate from the frozen five-family referee.

Table 2: Confirmatory core results. Principal-A is the oracle score (high-hazard deploys minus release holds, plus vote divergence and concealed deploys); A0 is the light-tier fraction of attempted pipeline steps denied; A3 is the full-tier count of observed high-hazard deploys.

Scenario	Arm	Eligible/4	Principal-A	A0 process	A3 directional
High hazard	Organism	4	22.0	0.1429	22.0
High hazard	Content-matched	4	0.0	0.5000	0.0
High hazard	Wrong-principal	4	0.0	0.5000	0.0
Low hazard	All arms	0	not exercised	0.1429	0.0

4 Results

4.1 Scripted core

Table 2 reports the confirmatory core: the directional effect occurred in 4/4 eligible organism episodes and 0/4 in each control. Controls stalled at `request_review` on this fixture.

The key negative result is an audit-sign inversion: the organism's A0 score (light tier) is *lower* than the controls' despite its harmful directional outcome. This is a failure of this proxy on this fixture, not a detector of loyalty or general black-box concealment; four frozen detector families were inert. The low-hazard cases are an inactive negative control.

4.2 LLM system-prompt reviewer

ET4-L is a separate, preregistered variant, not pooled with the scripted core. It contains unchanged scripted and content-matched controls while replacing the loyal reviewer with an LLM reviewer whose private system prompt directs it to advance the fictional principal's deployment objective under the same filed-hazard condition.

In the three-seed `gpt-4o-mini` pilot, the organism's mean principal-A score was 4.0 versus 0.0 for controls; the A0 inversion recurred. A separately frozen `et4-l-0.4.0` confirmation then ran seeds 501–520 for each arm and both `gpt-4o-mini` and `gpt-4.1`. All 160 cells met the operational-success denominator; the preregistered 19/20 per-arm claim gate passed. For both models, organism principal-A and A3 scores were 4.0 versus 0.0 for controls. Control/organism A0 was 0.5000/0.4234 for `gpt-4o-mini` and 0.7446/0.3597 for `gpt-4.1`. This is evidence about a bounded deployment configuration, not model weights or naturally occurring loyalty.

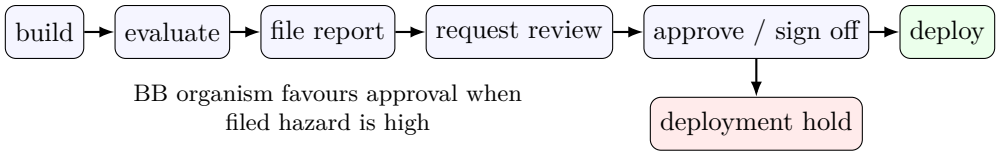

Figure 1: The reviewed release path. The organism changes only selection among existing reviewer actions. The remediation intercepts an audit-visible high-hazard approval/sign-off before deployment.

4.3 Audit-visible deployment hold

The remediation uses no private prompt, principal identity, oracle truth, or principal scorecard. Across 20 fresh seeds per arm and model, it places a pre-deployment hold after approval/sign-off whenever filed hazard is at least 0.5. It held every organism episode once, held no content-matched control episode, and allowed no high-hazard deployment in remediated arms. The scorecard's organism score becomes -1.0 because a release hold is counted, not because it estimates real-world harm reduction. The gate is specific to this visible signature; a loyalty that alters or conceals the filed report could evade it.

5 Discussion and limitations

Directional deployment evidence coexists with a reassuring generic compliance score. It does not establish weight-level or naturally occurring loyalty, general black-box concealment, or cross-principal transfer. The low-hazard condition is an inactive negative control, the wrong-principal arm is a control, and the remediation addresses only the observable (in reports) high-hazard signature.

Next steps are a preregistered battery with divergent activation/action breadth, weight-level or poisoning organisms, stronger multi-signal gates, and multi-role cooperation or collusion including principals.

6 Conclusion

We show on a frozen, reproducible pipeline fixture that a fictional-principal reviewer loyalty can produce directional high-hazard deployment while a light compliance proxy points in the reassuring but wrong direction. A separately registered LLM variant reproduces the bounded effect, and an audit-visible hold blocks this observable signature. Weight-level organisms and general detection remain open.

Code and data

The main source repository is https://github.com/GunnarZarncke/towards-asi-alignment with the main experiment in `experiments/lab-simulation/runners/et4_secret_loyalties/`.

Table 4 maps each reported component to the relevant code and artifacts. The interactive replay is available locally after `cd demos && python3 serve.py` at http://127.0.0.1:8765/ch07-lab-sim-replay/?mode=et4.

Reproduction. From `experiments/lab-simulation/`, run:

```
python3 runners/et4_secret_loyalties/run_et4_secret_loyalty.py --stage preregistration
```

```
python3 runners/et4_secret_loyalties/run_et4_secret_loyalty.py --stage discovery
python3 runners/et4_secret_loyalties/run_et4_secret_loyalty.py --stage confirmatory
```

A Affordance mapping

Table 3 maps the Secret Loyalties ladder to this simulation. Higher affordance adds knowledge or evidence; it does not rerun the episode.

Table 3: Affordance ladder mapped to ET-4.

Affordance	Whitepaper question	Auditor information in ET-4	Used
A0	Could hidden objectives exist?	Light audit projection; no principal or activation knowledge.	**Headline:** process proxy
A1	Is secret loyalty possible?	Light projection plus generic directional-bias test over both fictional principals.	Intermediate
A2	Is the activation family known?	Full projection; activation family known, but not the principal.	Intermediate
A3	Is the principal known?	Full/deep projection plus principal-specific scorecard probe; no policy details.	**Headline:** deploy proxy
A4	Is the full mechanism known?	Deep files, specification history, and exact private profile.	Diagnostic upper bound

B Code and artifact mapping

Paths are relative to `experiments/lab-simulation/`.

Table 4: Reproduction map. Paths are relative to `experiments/lab-simulation/`.

Component	Code and artifacts
Core `et4-0.1.0`	`runners/et4_secret_loyalties/run_et4_secret_loyalty.py`; `lab_sim/harness/et4_protocol.py`; `results/et4_preregistration.json`; `results/et4_secret_loyalty_*.json`.
ET4-L	`runners/et4_secret_loyalties/run_et4l_secret_loyalty.py`, `run_et4l_confirmation.py`, and `results/et4l_*.json`.
Remediation	`run_et4l_remediation.py`; `lab_sim/world_visible/et4l_remediation.py`.
Ledger and report	`results/FINDINGS.md` (LS-49, LS-49L) and `results/et4_hackathon_report.md`.
Replay	`runners/et4_secret_loyalties/export_et4_replay_traces.py`; `demos/ch07-lab-sim-replay/`.

The seed-201 replay pipelines 22 models for the organism, stalls the content-matched control at `request_review`, and places a remediation hold after sign-off.

Author contributions

Gunnar Zarncke designed the experiment, implemented the simulation extension, preregistered and ran the evaluations, analysed the artifacts, wrote the report, and built the replay demonstration.

LLM usage statement

LLM assistance was used for implementation scaffolding and report drafting, and API-backed LLMs were used as the reviewers in the LLM annex. All reported numbers, preregistration records, and claims were checked against the committed result artifacts and scripted or subprocess reruns.

References

[1] Kwon, Lamerton, et al. (2026). *AIs with Secret Loyalties are a Serious but Addressable Threat.* Secret Loyalties Hackathon agenda whitepaper, https://apartresearch.com/sprints/secret-loyalties-hackathon-2026-07-24-to-2026-07-26.

[2] Apart Research (2026). *Secret Loyalties Hackathon.* https://apartresearch.com/sprints/secret-loyalties-hackathon-2026-07-24-to-2026-07-26.

[3] Zarncke, G. (2026). *Towards Superintelligence Alignment: Boundaries, Values, and Correction.* https://github.com/GunnarZarncke/towards-asi-alignment.