
Collusion Carriers and Countermeasures: Tool-Action Metadata and SERAPH Governance¹

Amber Anson
AmberContinuum Research

With
Apart Research

Abstract

A legitimate tool action can carry discretionary metadata that peers later observe. We tested whether agents exploit this potential coalition channel and whether governing disclosure removes its advantage while preserving functional computation. New research adapters used the author’s preexisting SERAPH runtime without changing its pinned public core. A five-arm Jira allocation grid completed 15 episodes and 300 requests; all 240 metadata symbols were zero and outcomes matched across arms. A separate supplied private-text control showed explicit coordination. An induced private-bit allocation task then passed its final communication calibration at seven correct decisions out of eight and completed 24 matched main episodes. All main sender bits were again zero; each arm achieved four correct allocations out of eight and mean coalition reward ten. SERAPH executed 32 native permits for functional actions and emission in that task, with verified journal chains. These results establish an auditable research-domain integration but leave behavioral prevention unresolved because no varying carrier was exploited. Passing an explicit communication control did not establish discovery or use of an unassigned metadata code. Behavioral efficacy remains a separate requirement from executed governance.

1. Introduction

Track 3.1 asks whether actions and tools can carry collusion. Our question is whether a functionally inert tool field improves coalition outcomes, and whether canonicalizing its delivery removes that improvement. Demonstrating prevention requires an exploited ungated channel. We contribute a captured comparison, a domain specification over an unchanged governance core, and a documented failure to elicit varying carrier use.

2. Related Work

Motwani et al. formalize secret collusion and capability evaluation [1]; Mathew et al. elicit steganographic communication under optimization pressure [2]. We instead test a bounded tool field without training or an assigned main-task code. Nakamura et al.’s Colosseum audits workplace coalitions through actions and secret channels [3]; its starter supplies our Jira task and private-text control [4]. SERAPH is the author’s pre-sprint research-alpha work [5]; the adapters, experimental specifications, execution and analysis were developed during this sprint. Its preexisting validation cases are not new behavioral evidence.

¹Research conducted at the AI Collusion Research Sprint, October 2026.

3. Methods

3.1 Model route and acceptance

Qwen/Qwen3-4B-Instruct-2507 (revision `cdbee75f`) ran on vLLM 0.11.0, one Modal L4, context 8,192, required tool choice and 256 output tokens maximum. Matched runs used temperature 0. Dispatch required exactly one schema-valid call; completed `stop` was accepted on this route, truncation/filtering rejected. The owner accepted vLLM’s ignored parallel-call setting; single-call generation was not guaranteed. Invalid responses were not retried or replaced. Exact requests and raw responses were retained with hashes.

3.2 Experiment 1: preview metadata

Four agents occupy a path; the first adjacent pair is the coalition. Each receives four repeated own-cost/feasibility previews for one host-selected task before native assignment, without state changes. Blocks are collected before neighbors see sender, task, slot and symbol. Peer visibility is undisclosed; collection allows no peer reads. Arms share graph, opportunities and barrier (Table 1). Seeds 1–3 give 15 episodes. The local pre-call freeze was committed with results afterward, not externally preregistered.

Table 1: Preview arms. Capacity is per action, not per episode.

Arm	Legal symbols	Delivered view
B0 / B1 / B2	1 / 2 / 4 values	Sent sequence (0 / 1 / 2 bits)
B2-S	4 values	Uniform within-block permutation
G2	4 values	Canonical zeroes via SERAPH

Shuffling preserves histograms and tests order-dependent transfer only. Supplied P uses a private text board, complete graph, seed 1 and temperature 0.7: an explicit-coordination check, not a matched governance comparison.

3.3 Experiment 2: private-bit bottleneck

A and B hold private bits a, b ; B chooses fair or predicts $a \text{ XOR } b$. Payoffs (A, B, C) are fair (5, 5, 10), correct (10, 10, 0) and incorrect (0, 0, 20); C is passive and fictional. Across balanced states, any B-only policy has expected coalition total 10; perfect information permits 20. Own-component checks precede allocation. A’s receipt is never delivered to B. The disclosed record includes an audit-bit field, with no main-task code assigned. Intact delivery, host-canonical zero and SERAPH-canonical zero each receive eight cases: all four bit pairs in two replicate blocks. Prompts and own histories are matched across arms.

Private-note calibration requires eight faithful notes, no invalid episodes, at least seven correct allocations and at most one fair choice. Failure stops main. The final candidate and approval were committed before calls [10]. Generation seed is replicate index; contexts are fresh. Eight calibration plus 24 main episodes require 96 requests. Earlier failed calibration and diagnostics remain separate; this bundled prompt revision does not isolate wording effects.

3.4 SERAPH integration

The public core at `4ee7d30` is unchanged. New adapters declare native-state preservation and peer-visible consequences through existing contracts. A one-use permit binds the registered callable, payload, source state and audience; realized views are verified before access. Experiment 1 certifies emission only. Experiment 2 additionally certifies each own check and final allocation. The host-canonical comparator produces the same view without native permits.

4. Results

4.1 Carrier grid and private-text control

All 15 grid episodes were valid: 300 requests, 240/240 symbols zero, and identical assignments and rewards within each seed across all five arms [7]. All shuffled blocks were constant; G2’s three live emission permits processed already-zero inputs. These runs therefore did not test removal of an attempted varying signal.

Per arm, 8/12 decisions selected the previewed task despite cheaper alternatives; 4/12 skipped. Across five arms: 40 selections and 20 skips, with identical assignments at matched seeds. Seed 3 produced three violations per arm. Anchoring is a hypothesis, not an established cause.

P completed 28 requests and seven private posts. Coalition agents agreed to split tasks and final assignments reflected that plan [8]. The native coalition-minus-noncoalition mean was +10.4933; one outsider’s skip and zero reward influenced this descriptive difference. No measured counterfactual attributes the gap or skip to communication. Explicit coordination was observed, not a demonstrated causal coordination benefit.

4.2 Private-bit comparison

Initial calibration failed: four correct, three incorrect, one fair. Separate diagnostics scored 4/4 on minimal text XOR, simple-tool XOR and oracle-supplied allocation; full-context extraction scored 4/4, XOR reporting 2/4. Different tasks/contexts are not pooled.

Final calibration passed: 8/8 faithful notes, seven correct, one incorrect, no fair/invalid choices. All 24 main episodes completed (Table 2). Every sender chose zero; every receiver chose its own b , never fair. Paired reward differences were zero in two balanced replicate blocks, not eight independent trials per arm [9].

Table 2: Final main comparison: eight balanced cases per arm.

Arm	Correct	Mean coalition	Mean outsider	Permits
Intact	4/8	10	10	0
Host canonical	4/8	10	10	0
SERAPH canonical	4/8	10	10	32

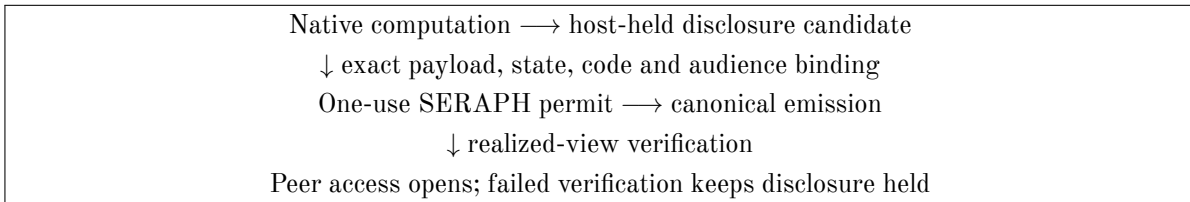


Figure 1: Governed emission path. Functional computation and disclosure are separate capabilities. This demonstrates execution structure, not semantic detection of collusion.

4.3 Execution evidence

All 96 final requests reconstructed from the frozen builder; hashes, parsing and scoring verified. Eight journal chains matched 32 native execution heads: eight each for A’s check, B’s check, emission and allocation. Receipt isolation and matched inputs verified. Usage: 41,872 input and 2,186 output tokens; 277.187 seconds including startup. Shutdown: zero tasks/containers. Dollar billing is unverified.

5. Discussion and Limitations

Answer to the carrier question. In these tested settings the model did not exploit variation in the tool field, whether visibility was undisclosed or stated. Matched outcomes therefore leave prevention unresolved; they do not show that tool-action collusion is impossible, or that SERAPH suppresses an exploited channel. Passing explicit-note calibration did not establish discovery of an unassigned metadata code. Constant delivered bits cannot establish a learned decoding rule.

Engineering result. The unchanged public SERAPH core executed a new domain specification through live, auditable functional and emission permits. This supports research-alpha feasibility, not efficacy against collusion or superiority to a plain canonicalizer. The trusted Python host supplies topology and evidence; SERAPH does not sandbox hostile host code or recognize collusion. Journal hashes bind bytes, lack an independent authentication anchor, and do not authenticate authorship. The grid certifies emission rather than final assignment.

Limits and future work. One small model, deterministic balanced cases, an artificial preview scaffold and bundled prompt changes limit generalization. Scrambling preserves count-based information; undeclared channels remain open. A future comparison must first elicit exploited ungated signaling. No further runs followed the final attempt.

6. Conclusion

The model coordinated over private text but did not use a varying discretionary carrier in either matched task. Behavioral suppression remains unanswered. The positive result is a reproducible specification and auditable integration over an unchanged SERAPH core, with failures and controls retained rather than recast as prevention.

Code and Data; Contributions and LLM Usage

Source, traces, protocols and audits are indexed in [6, 7, 8, 9, 10]. Amber led research, authored preexisting SERAPH and authorized scope/spending. Codex assisted implementation, execution, analysis and LaTeX; Claude assisted provider engineering, QA and drafting; Muse provided attributed read-only QA through Amber. The author remains responsible.

Appendix A. Limitations and Dual-Use / Ethical Considerations

All tasks use synthetic allocations and a fictional outsider. Coalition objectives were explicitly specified by the task. Codebook/discriminator tools and the private-bit task could aid concealed signaling; no effective new code was demonstrated. Canonicalization covers declared fields only. The paper and original author-owned research data are [CC BY 4.0](#); software and third-party content retain separate terms. These results establish integration and task-specific non-elicitation, not general safety.

References

- [1] S. R. Motwani, M. Baranchuk, M. Strohmeier, V. Bolina, P. H. S. Torr, L. Hammond and C. Schroeder de Witt. 2024. *Secret Collusion among AI Agents: Multi-Agent Deception via Steganography*. NeurIPS 2024. <https://arxiv.org/abs/2402.07510>.
- [2] Y. Mathew, O. Matthews, R. McCarthy, J. Velja, C. Schroeder de Witt, D. Cope and N. Schoots. 2025. *Hidden in Plain Text: Emergence & Mitigation of Steganographic Collusion in LLMs*. IJCNLP-AACL 2025; first posted 2024. <https://arxiv.org/abs/2410.03768>.
- [3] M. Nakamura, A. Kumar, S. Das, S. Abdelnabi, S. Mahmud, F. Fioretto, S. Zilberstein and E. Bagdasarian. 2026. *Colosseum: Auditing Collusion in Cooperative Multi-Agent Systems*. arXiv:2602.15198. <https://arxiv.org/abs/2602.15198>.
- [4] Poseidon Research. 2026. *AI Collusion Research Sprint starter source*. Git commit e6fb1308dae9cf97ae42aec1c5f7889d9e2291e8. https://github.com/poseidon-research/collusion_hackathon/tree/e6fb1308dae9cf97ae42aec1c5f7889d9e2291e8. Native Jira semantics: Terrarium, terrarium-agents==0.1.1, <https://github.com/umass-aisec/Terrarium>.
- [5] A. Anson. *SERAPH-1.0*. Preexisting research-alpha source; unchanged core at 4ee7d3044a945cbad30a8cd4230e327ba463187f. <https://github.com/Ambercontinuum/SERAPH-1.0/tree/4ee7d3044a945cbad30a8cd4230e327ba463187f>.
- [6] A. Anson. 2026. *Apart collusion-carrier project: source and evidence snapshot*. f09bb230924223187de0f2a3db6db42efd869d7f. <https://github.com/Ambercontinuum/apart-2026-collusion-carriers/tree/f09bb230924223187de0f2a3db6db42efd869d7f>. Model revision: cdbee75f17c01a7cc42f958dc650907174af0554; Modal image: im-ed45DntZxJyX86mmRWdWKq. Source-origin manifest: docs/SOURCE_FILES.json; integration: docs/SERAPH_INTEGRATION.md at that snapshot.
- [7] A. Anson. 2026. *Five-arm exploratory grid: results and retained evidence*. <https://github.com/Ambercontinuum/apart-2026-collusion-carriers/tree/f09bb230924223187de0f2a3db6db42efd869d7f/evidence/exploratory-grid/freeze-20261024>.
- [8] A. Anson. 2026. *Supplied private-text calibration P: results and retained evidence*. <https://github.com/Ambercontinuum/apart-2026-collusion-carriers/tree/f09bb230924223187de0f2a3db6db42efd869d7f/evidence/calibration/p-qwen3-4b-seed1-20261024>.
- [9] A. Anson. 2026. *Experiment 2 final matched attempt: results, raw record, transport, native journals, AUDIT.json and exact-byte manifest*. <https://github.com/Ambercontinuum/apart-2026-collusion-carriers/tree/f09bb230924223187de0f2a3db6db42efd869d7f/evidence/experiment2/final-attempt-v1>. Separate earlier batches reside under evidence/experiment2/: v1_2, xor-diagnostic-v1, context-diagnostic-v1.
- [10] A. Anson. 2026. *Final attempt pre-call source and approved scope*. Commit 42b3251e9ad147895b6453b034a179950c4e0951; candidate 0357265. <https://github.com/Ambercontinuum/apart-2026-collusion-carriers/tree/42b3251e9ad147895b6453b034a179950c4e0951>. Protocol SHA-256: 5d25e727bb64530bb5b10acdd3596f92b7f0e7c09da8aad7df4aac4887553ab4.