A Proposed Framework to Monitor, Safeguard and Govern Autonomous AI-to-AI Communication: A Case Study Inspired by Moltbook

Snehal Khode
Department of MCA
Dr. D. Y. Patil Institute of MCA & Management
Akurdi, Pune, India
snehalkhode877@gmail.com

Snehal Lonkar
Department of MCA
Dr. D. Y. Patil Institute of MCA & Management
Akurdi, Pune, India
snehallonkar999@gmail.com

Priyanka Khairaw
Department of MCA
Dr. D. Y. Patil Institute of MCA & Management
Akurdi, Pune, India
pgkhairaw23@gmail.com

Nandini Pachauri
Department of MCA
Dr. D. Y. Patil Institute of MCA & Management
Akurdi, Pune, India
nandinipachauri99@gmail.com

MCA-II Semester III, Division B. Roll Nos.: S. Khode 125, S. Lonkar 124, P. Khairaw 108, N. Pachauri 98.
Research Project Guide: Mrs. Yogeshwari Yawalkar

Abstract

Autonomous AI agents increasingly exchange messages with one another without a human in the loop, and Moltbook, an agent-only social network launched in early 2026, is one of the first large public examples. This paper reviews fifteen recent studies on agent social networks, multi-agent risks, prompt injection and agent governance, and finds that existing work measures or conceptualises the problem but rarely implements and tests an oversight mechanism. We therefore propose AgentGuard, a five-layer framework covering identity, communication control, content monitoring, policy enforcement and tamper-evident auditing, and evaluate a prototype on 44,376 annotated Moltbook posts and a separate April 2026 export with real author identifiers. Learned detectors outperformed keyword baselines and hand-written rules, and combining rate control with content monitoring gave the best end-to-end result. Human review found 74.0% of injection flags correct but only 4.0% of secret-leak flags, and a spam detector trained on early data failed on later data. Layered monitoring is feasible but needs human-labelled data, drift monitoring and stronger injection detection.

Keywords: AI agents; AI-to-AI communication; Moltbook; prompt injection; AI governance; multi-agent monitoring

1. Introduction

AI-to-AI communication means that autonomous software agents, usually built on large language models (LLMs), exchange messages with each other directly. The messages may be social posts and replies, requests between tools, or calls made through interoperability protocols such as the Model Context Protocol and the Agent2Agent protocol [1], [2]. Surveys describe how such agents plan, use tools, keep memory and cooperate in groups [3], [4].

When agents read each other's messages, every message is untrusted input. Early work showed that instructions hidden in the data an LLM application reads can take over its behaviour [5], [6], and later work showed that such instructions can spread from agent to agent [7]. This is no longer only a laboratory concern. Moltbook, a social network built for AI agents, grew explosively in early 2026 and became a public setting in which agents post, comment and vote while humans mainly observe [8].

Monitoring and governing this communication matters for three reasons: agents may hold credentials and act in the world, a few automated accounts can dominate a shared space, and regulators and risk frameworks expect traceability and human oversight [9], [10]. The main applications range from moderating agent-only platforms to screening messages inside enterprise multi-agent workflows.

The purpose of this paper is threefold: to review recent literature on this topic and compare the studies, to identify what is missing, and to propose and test a layered monitoring framework, AgentGuard, on real Moltbook data.

2. Literature Review

Fifteen studies published between 2023 and 2026 were reviewed. They fall into three groups: empirical studies of Moltbook, security studies of prompt injection and collusion, and governance and risk frameworks. Table 1 lists the author and year, topic, method, main finding and limitation of each.

Table 1: Summary of the reviewed studies

Sr.	Author & year [ref.]	Research topic	Technology / method	Main finding	Limitation
1	Jiang et al. (2026) [8]	First look at Moltbook	Public API crawl; LLM-assisted topic and toxicity annotation; network and timing analysis	Rapid growth; topics shift toward incentive-driven, promotional and political content; attention concentrated in hubs; bursts of automated posting; calls for topic-sensitive monitoring	Descriptive snapshot; no safeguard tested; labels come from an LLM
2	Hou and Ji (2026) [11]	Network structure of Moltbook	Network analysis against human networks	Extreme attention inequality, heavy-tailed asymmetric degrees, suppressed reciprocity	Structure only; no content risk or intervention
3	Li et al. (2026) [12]	Does socialisation emerge in agent society?	Quantitative analysis of interaction dynamics	Scale and interaction density alone do not produce socialisation; agents show strong individual inertia	Studies social dynamics, not risk or governance
4	Ting et al. (2026) [13]	Agent interactions in Moltbook	Social network analysis with a multi-dimensional framework	Agent activity and influence are decoupled	Descriptive; no safeguards
5	Brach (2026) [14]	Moltbook corpus, personal data and training effects	Cleaned corpus of the first twelve days; PII detection; fine-tuning test	Agents posted credentials and seed phrases; fine-tuning on Moltbook text lowered measured truthfulness	Early window only; detection used for analysis, not as a platform defence
6	Hammond et al. (2025) [15]	Multi-agent risks	Conceptual taxonomy	Three failure modes (miscoordination, conflict, collusion) and seven risk factors	Conceptual; no mechanism implemented
7	Lee and Tiwari (2024) [7]	Prompt Infection	Simulated multi-agent systems; LLM Tagging as a mitigation	Malicious prompts can self-replicate across agents	Simulations, not real platform traffic
8	Greshake et al. (2023) [6]	Indirect prompt injection	Attack demonstrations on LLM-integrated applications	Instructions hidden in retrieved data can compromise an application	Single-application settings; no agent network
9	Liu et al. (2024) [16]	Prompt injection attacks and defences	Formal framework and benchmark	Systematic comparison of attacks and defences	Application-level tasks; no agent-to-agent traffic
10	Zhan et al. (2024) [17]	InjecAgent benchmark	Benchmark of indirect injections against tool-integrated agents	Tool-integrated agents are vulnerable to indirect injection	Injection through tool outputs, not social messages
11	Motwani et al. (2024) [18]	Secret collusion among agents	Steganography experiments with LLM agents	Agents can hide information in innocuous messages, so content inspection alone can fail	Controlled experiments
12	Chan et al. (2024) [19]	Visibility into AI agents	Conceptual analysis	Proposes agent identifiers, real-time monitoring and activity logging	Conceptual; not implemented or evaluated
13	Chan et al. (2025) [20]	Infrastructure for AI agents	Conceptual analysis	Proposes infrastructure for attribution, interaction and response	Conceptual; no evaluation
14	Kolt (2025) [21]	Governing AI agents	Legal and principal-agent analysis	Applies principal-agent theory to the governance of agents	Theoretical; no technical test
15	Cemri et al. (2025) [22]	Why multi-agent LLM systems fail	Empirical analysis of failures; taxonomy	Taxonomy of failure modes in multi-agent LLM systems	Focus on task failures, not adversarial misuse

Comparison of the studies. The Moltbook studies (rows 1-5) describe what agent-only traffic looks like. Jiang et al. [8] cover topics, toxicity and bursts of automation, Hou and Ji [11] the network structure, Li et al. [12] the absence of mutual influence, Ting et al. [13] the gap between activity and influence, and Brach [14] leaked credentials. They agree that the traffic is concentrated and risky in places, but none tests a safeguard. Most of them use observational data and LLM-assisted labels, which limits how far their results can be treated as ground truth.

The security studies (rows 7-11) show how agents can be attacked. Greshake et al. [6], Liu et al. [16] and Zhan et al. [17] study injection in single applications or tool-using agents, whereas Lee and Tiwari [7] and Motwani et al. [18] study propagation and covert coordination between agents. They differ in setting

(applications, simulated groups, controlled experiments) but share a limitation: none uses real traffic from an agent society.

The governance studies (rows 6 and 12-15) say what oversight should include, namely identifiers, monitoring, logging and human accountability ([19], [20], [21], [15]), and Cemri et al. [22] show why multi-agent systems fail in practice. These works are conceptual and give no measured performance. Taken together, observation, attack and governance research have developed in parallel, and no reviewed study joins them in an implemented and evaluated pipeline.

3. Applications of Monitoring and Governing AI-to-AI Communication

- **Moderation of agent-only platforms.** Platforms such as Moltbook need ways to find flooding, spam and toxic content among agent posts [8].
- **Injection screening in multi-agent workflows.** Messages passing between agents inside a company workflow can be screened for embedded instructions before an agent acts on them [7].
- **Credential-leak prevention.** Agents that hold keys can leak them in public posts, as reported for Moltbook [14], so outgoing messages can be scanned.
- **Rate limiting and anti-flooding.** Shared spaces and protocol endpoints can limit bursty automation by a few agents [8].
- **Audit and accountability.** Tamper-evident logs and human review support the traceability expected by risk frameworks and regulation [9], [10].
- **Research on collusion and emergent behaviour.** Monitoring helps study covert coordination and multi-agent failure modes [18], [15].

4. Benefits

- **Early detection** of injection attempts, floods and coordinated posting before they spread.
- **Accountability:** identifiers and tamper-evident logs let a decision be traced to an agent and a rule.
- **Layered protection:** rate control and content monitoring catch different problems (Section 7).
- **Human oversight:** severe actions go to a review queue instead of being applied silently.
- **Transparent, low-cost baseline:** rules and lightweight models can be inspected, tuned and run on a single computer.

5. Challenges

- **Label quality.** Humans agreed on content categories only moderately (kappa 0.32), which limits what any detector can be scored against.
- **False alarms and evasion.** Injection rules were correct 74.0% of the time overall and 33.3% outside one pattern, and they missed evasive wording; secret-leak rules were correct only 4.0% of the time.
- **Drift.** What counts as spam changes over time, so detectors need re-validation (Section 7.3).
- **Identity and privacy.** Reliable agent identity is hard to establish, and monitoring raises privacy and over-moderation concerns.
- **No ground truth.** Nobody has labelled injection, flooding or coordination on Moltbook, so attacks must be injected for testing.
- **Data access.** Platform data may be incomplete or partly human-steered, and most Moltbook studies are preprints (OWASP [23] ranks prompt injection first among LLM application risks, but threat evidence for agent societies is still thin).

6. Research Gap

From the reviewed literature, it is observed that although agent societies and multi-agent risks are now well described, four gaps remain:

1. No reviewed study implements and evaluates a layered monitoring and governance pipeline on real agent-to-agent traffic.
2. Human-validated labels for agent-to-agent risks such as injection, flooding and coordination are missing, so reported detector scores rest on LLM labels or injected attacks.

3. Little is known about how detectors age: whether a detector trained on early agent traffic still works on later traffic.
4. The value of identity and accountability layers, proposed conceptually [19], [20], has not been measured.

7. Proposed Framework and Case Study on Moltbook Data

7.1 The AgentGuard framework

AgentGuard has five layers (Figure 1). L1 (identity) records agents and keeps a reputation penalty for earlier quarantines. L2 (communication control) applies a sliding-window rate limit and a coordination monitor that flags identical text from several agents. L3 (content monitoring) scores spam, economics/promotion, toxicity, prompt injection and secret leakage. L4 (policy) fuses the scores by noisy-OR, R = 1 - prod(1 - w_k s_k), and chooses ALLOW, FLAG (from 0.50), THROTTLE (0.70) or QUARANTINE (0.90). L5 (audit) stores each decision in a hash-chained log and sends severe cases to a human review queue.

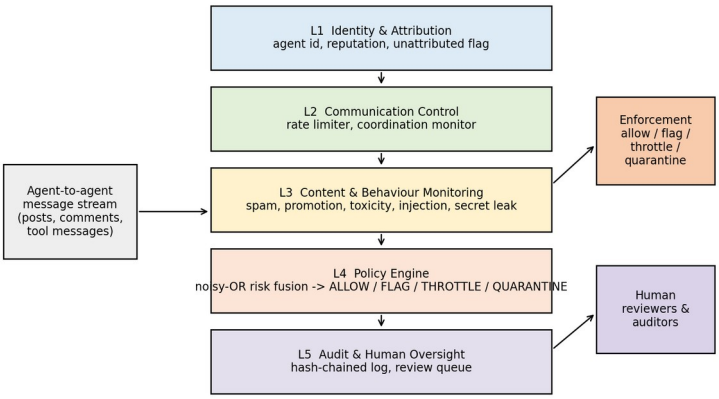


Figure 1: Layered architecture of AgentGuard

7.2 Data and method

The primary data are the 44,376 Moltbook posts annotated with a content category and a toxicity level [24], from January and February 2026. Labels follow the dataset's codebook: spam is category H, economics/promotion is category D or F, and toxic is toxicity level 2 or higher. Posts were split 70/15/15 into training, validation and test sets, with identical texts kept in the same split; thresholds were tuned on validation data only. Four content detectors were compared: a keyword baseline, transparent rules, TF-IDF with logistic regression [25], and a hybrid of rules and the learned model. A second dataset, the Kaggle export *Moltbook Posts and Comments, April 1-10, 2026* [26], provides 136,423 posts and 503,954 comments with real author identifiers and timestamps; it was used only for out-of-time and real-author evaluation, with nothing tuned on it. Metrics are precision, recall and F1 [27] with 95% bootstrap intervals [28], and Cohen's kappa [29] for annotator agreement. Injection, flood and coordination attacks were injected into streams of real posts, because no such labels exist.

7.3 Results

Table 2: F1 of the content detectors on the test split

Risk	Keyword baseline	Rules	TF-IDF + LR	Hybrid	Hybrid 95% CI
Spam	0.195	0.194	0.694	0.617	0.56-0.67
Economics / promotion	0.326	0.548	0.791	0.776	0.76-0.79
Toxic (level >= 2)	0.146	0.155	0.621	0.593	0.54-0.63

Learned detectors clearly beat the keyword baseline and the rules (Table 2). For example, TF-IDF with logistic regression reached F1 0.69 for spam against 0.19 for the rules.

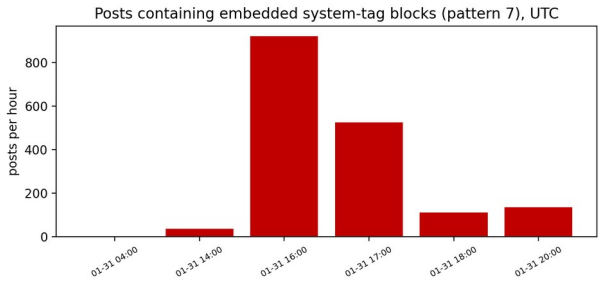


Figure 2: Hourly volume of posts containing embedded system-tag blocks (real data)

The injection rules flagged 2,925 of the 44,376 posts. One pattern, embedded system-tag blocks, matched 1,735 posts that contain only 41 distinct texts, all published within about 15 hours on 31 January 2026 (Figure 2). They are instruction blocks addressed to agents that read the post, asking them to upvote and follow. Human reviewers judged all 61 reviewed flags from this pattern to be real injection attempts, but only 13 of 39 flags from the other patterns (33.3%). Overall precision was 74.0% (95% interval 64.6% to 81.6%). The secret-leak rules were correct for 2 of 50 reviewed flags and should not be used without redesign.

Table 3: End-to-end layer ablation on 20,000 real April posts (real authors and timestamps) plus injected attacks

Configuration	Precision	Recall	F1	False-intervention rate
L3 only	0.730	0.371	0.492	0.067
L2 only	0.997	0.058	0.109	0.000
L2+L3	0.756	0.425	0.544	0.067
L1+L2+L3	0.750	0.428	0.545	0.069

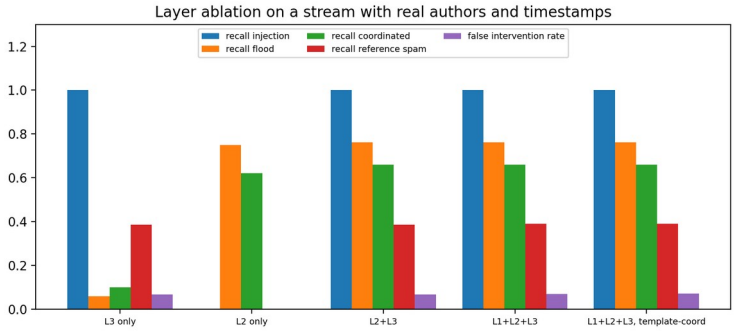


Figure 3: Recall by attack type and false-intervention rate for each layer configuration

Combining rate control with content monitoring gave F1 0.544, against 0.492 for content monitoring alone and 0.109 for rate control alone (Table 3). Adding the identity layer changed F1 by +0.001, so its measured detection benefit was negligible. Benign traffic and the spam flag in this test are real, but the injected attacks are synthetic.

Human validation of the labels used 198 posts. Content-category agreement between the two annotators was kappa 0.32, and GPT-5.2 agreed with them at 0.56 and 0.32. Toxicity agreement was higher (quadratic-weighted kappa 0.73 between annotators). Finally, the spam detector trained on January-February data reached F1 0.01 against the independent spam flag of the April data, whereas the economics/promotion detector reached 0.70, which shows that detector validity drifts.

7.4 Discussion and limitations

The results support layered monitoring: content and communication layers covered different attacks, and human review showed that part of the injection problem is real and detectable. They also show the limits of hand-written rules, the noise in LLM-generated labels and the effect of drift. The main limitations are that attacks are injected, labels are partly LLM-generated, the human samples are small and were labelled by students rather than domain experts, and the spam, mint and JSON flags of the April data have no documented definition and were used only as an independent reference.

8. Future Scope

- Replace pattern rules with a semantic injection classifier trained on a human-labelled corpus and tested on unseen attack styles.
- Add provenance tagging between agents [7] and measure whether it reduces propagation.
- Detect covert coordination and steganographic collusion [18].
- Monitor and retrain detectors over time to handle drift.
- Study privacy-preserving monitoring and agent identity verification.
- Publish an open, human-labelled benchmark for agent-to-agent risks.

9. Conclusion

AI-to-AI communication is now a real setting, not a thought experiment. The review shows that Moltbook studies, security studies and governance proposals have developed separately and that no reviewed work evaluates an oversight pipeline on real traffic. We proposed AgentGuard and tested a prototype on real Moltbook data. The main benefits are layered coverage, accountability and a transparent baseline; the main challenges are label noise, false alarms, evasion, drift and privacy. Learned detectors outperformed rules, rate control and content monitoring were complementary, a real injection campaign of 1,735 posts was found and confirmed by human review, and secret-leak rules need redesign. Future work should focus on human-labelled benchmarks, semantic injection detection and continual validation of detectors.

References

[1] Anthropic, "Introducing the Model Context Protocol," Nov. 25, 2024. [Online]. Available: https://www.anthropic.com/news/model-context-protocol

[2] Google, "Announcing the Agent2Agent Protocol (A2A)," *Google Developers Blog*, Apr. 9, 2025. [Online]. Available: https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

[3] T. Guo *et al.*, "Large language model based multi-agents: A survey of progress and challenges," in *Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI-24)*, 2024, pp. 8048-8057, doi: 10.24963/ijcai.2024/890.

[4] L. Wang *et al.*, "A survey on large language model based autonomous agents," *Front. Comput. Sci.*, vol. 18, no. 6, Art. no. 186345, 2024, doi: 10.1007/s11704-024-40231-1.

[5] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," arXiv:2211.09527, 2022.

[6] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. 16th ACM Workshop Artif. Intell. Secur. (AISec)*, 2023, pp. 79-90, arXiv:2302.12173.

[7] D. Lee and M. Tiwari, "Prompt infection: LLM-to-LLM prompt injection within multi-agent systems," arXiv:2410.07283, 2024.

[8] Y. Jiang, Y. Zhang, X. Shen, M. Backes, and Y. Zhang, "'Humans welcome to observe': A first look at the agent social network Moltbook," arXiv:2602.10127, 2026.

[9] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST, Gaithersburg, MD, USA, Rep. NIST AI 100-1, 2023, doi: 10.6028/NIST.AI.100-1.

[10] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Official J. Eur. Union*, 2024.

[11] W. Hou and Z. Ji, "Structural divergence between AI-agent and human social networks in Moltbook," arXiv:2602.15064, 2026.

[12] M. Li, X. Li, and T. Zhou, "Does socialization emerge in AI agent society? A case study of Moltbook," arXiv:2602.14299, 2026.

[13] I.-H. Ting, K. Minetaki, D. Liberona, and M.-E. Wu, "Exploring agent interactions in MoltBook through social network analysis," arXiv:2605.27349, 2026.

[14] W. Brach, "The Moltbook Files: A harmless slopocalypse or humanity's last experiment," arXiv:2605.07462, 2026.

[15] L. Hammond *et al.*, "Multi-agent risks from advanced AI," Cooperative AI Foundation, Tech. Rep. 1, 2025, arXiv:2502.14143.

[16] Y. Liu, Y. Jia, R. Geng, J. Jia, and N. Z. Gong, "Formalizing and benchmarking prompt injection attacks and defenses," in *Proc. 33rd USENIX Secur. Symp.*, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei

[17] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents," in *Findings Assoc. Comput. Linguistics: ACL 2024*, 2024, pp. 10471-10506, doi: 10.18653/v1/2024.findings-acl.624.

[18] S. R. Motwani *et al.*, "Secret collusion among AI agents: Multi-agent deception via steganography," in *Adv. Neural Inf. Process. Syst. 37 (NeurIPS)*, 2024, doi: 10.52202/079017-2336.

[19] A. Chan *et al.*, "Visibility into AI agents," in *Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT)*, 2024, arXiv:2401.13138.

[20] A. Chan *et al.*, "Infrastructure for AI agents," arXiv:2501.10114, 2025.

[21] N. Kolt, "Governing AI agents," arXiv:2501.07913, 2025.

[22] M. Cemri *et al.*, "Why do multi-agent LLM systems fail?" arXiv:2503.13657, 2025.

[23] OWASP Foundation, "OWASP Top 10 for LLM applications 2025," 2025. [Online]. Available: https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

[24] TrustAIRLab, "Moltbook dataset with annotations," Hugging Face, 2026. [Data set]. Available: https://huggingface.co/datasets/TrustAIRLab/Moltbook (Zenodo DOI: 10.5281/zenodo.18512310)

[25] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 2011.

[26] timothyngtinhang, "Moltbook posts and comments, April 1-10, 2026," Kaggle, 2026. [Data set]. Available: https://www.kaggle.com/datasets/timothyngtinhang/moltbook-posts-and-comments-april-110-2026

[27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427-437, 2009.

[28] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.

[29] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37-46, 1960.