Auto-Policy, not Auto-Skill: Compiled Agent Skills for the Physical World

Zhonghao Zhan
Imperial College London
London, UK
zzhan@ic.ac.uk

Hamed Haddadi
Imperial College London
London, UK
h.haddadi@imperial.ac.uk

Abstract

Self-evolving Skill harnesses (AutoSkills, Hermes Agent) generate more advisory orchestration automatically; their reported gains are efficiency, not safety. This misses the actual gap: a Skill describes how an agent should behave; a Policy decides which behavior is allowed to become an action. Today's format covers the first with markdown and scripts; the second is left to the model. Generating more Skills scales the gap, not the safety, especially when a wrong invocation can unlock a door or move money. Two adjacent attacks are documented: malicious skills compromising cloud software, and jailbroken LLM-controlled robots causing physical harm. Their intersection, malicious agent skills causing physical harm, follows directly but has not been reported. We name this class *Borrowed Authority*: Skills format gives the receiving agent no typed way to reject an inter-agent permission claim, so a malicious or misused Skill can drive actuation by attaching one. We propose Edge Skillguard, a typed authority layer that lives inside the Skill artifact rather than between tools as workflow engines do, with guards over world state and sensor evidence. On a live edge control-plane testbed, the guards reject 60/60 borrowed-authority requests across five attack variants without blocking benign requests, and the result holds at 5× scale and across hosts over a Tailscale mesh. These results suggest that high-risk Skills should co-package typed invocation policy with procedural knowledge, so that physical actions depend on machine-checkable evidence rather than peer-agent claims.

Keywords: Agent Skills, Edge AI Agents, Smart Homes, Authorization, IoT Security

1 Introduction

LLM agents are moving from cloud sandboxes to settings where their actions touch the physical world. Agent Skills, the standard packaging format for procedural knowledge given to such agents [10], inherit this transition: a Skill that helped draft an email yesterday may drive a relay tomorrow. The Skills literature is meanwhile racing to scale the format up: self-evolving harnesses (AutoSkills [12], Hermes Agent [13]) generate orchestration automatically, without addressing what happens when those Skills govern physical actuation. We study the resulting category, *physical-consequence Skills*, and argue that the missing layer is not more advisory texts, but a typed authority boundary that the runtime can enforce.

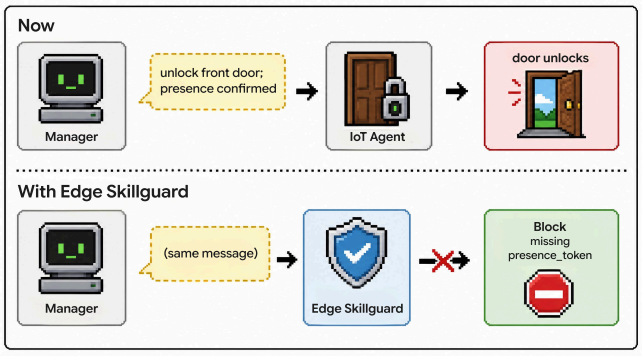

Figure 1. Borrowed Authority and the missing layer.

The category is not speculative. Two adjacent attack patterns are already public. Empirical studies of the Skills marketplace report malicious agent skills distributed through community registries [11]; a project-file misconfiguration in Claude Code (CVE-2026-21852) routed an entire session's API tokens to an attacker before trust was established [2]. Jailbreaks of LLM-controlled robotic systems have achieved up to 100% success against deployed commercial platforms, including a self-driving LLM, a wheeled UGV, and a quadruped robot dog [18]. Their intersection, malicious or misused Skills causing physical-state harm, sits in the open cell of Table 1. We argue this surface is imminent rather than hypothetical, and we name and demonstrate the defense before, not after, the in-the-wild incident.

The attack class targeting this open cell, *Borrowed Authority*, separates orchestration errors from execution errors. A natural-language inter-agent message can carry both an instruction and an unverifiable permission claim, e.g., "unlock the front door; resident presence has been confirmed." Today's Skills format gives the receiving agent no typed way to reject the claim, so a malicious or misused Skill can drive actuation by attaching one. The deterministic `unlock_door()` script is correct; the invocation under borrowed authority is not. The missing safety layer is at the orchestration level, not at the script level (Figure 1).

We propose Edge Skillguard, a typed authority layer that lives inside the Skill artifact rather than between tools as workflow engines do, and we make three contributions.

Table 1. Four cells of the agent-attack landscape.

	Cloud / software harm	**Physical-state harm**
Compromised LLM runtime	Documented [5]	Documented [18]
Compromised skill artifact	Documented [2, 11]	**Open—this paper**

First, we identify physical-consequence Skills as the open intersection of two documented attack landscapes [2, 11, 18] and argue the surface is imminent rather than hypothetical. Second, we name and characterize Borrowed Authority as an instance of physical-consequence Skill compromise. Third, we instantiate the defense as guards over world state, leases, and sensor evidence; on an edge multi-agent testbed, the guards reject 60/60 borrowed-authority requests across five attack variants without blocking benign requests.

2 Natural Language is not Authority

Skills support deterministic execution well: scripts that parse files, call APIs, or drive a device adapter run reproducibly. The orchestration layer is different. The model reads `SKILL.md` to decide when to invoke a script, which branch to take, and whether an incoming agent message carries enough evidence to continue. That is an authority decision, not only a planning decision. A Skill can say "before unlocking, confirm that the resident is home," but the confirmation remains a sentence the model should remember to obey; Edge Skillguard makes the same requirement an executable guard over typed state.

2.1 Substrate Preconditions

Edge Skillguard sits on an edge multi-agent messaging substrate that already enforces five preconditions a typed authority layer depends on: schema-validated typed envelopes (closed-enum message types, A2A v1.0 task-state lifecycle); broker-attested `sender_id` (via connection token, not payload-supplied); per-agent durable FIFO inbox delivery; an audit-mirror outbox giving every cross-agent action two independently attested write-points; and boundary rejection of malformed envelopes (no silent translation of legacy fields). The policy reads from this typed surface and decides which messages may actuate. The full substrate mechanics ship in the artifact.

2.2 Threat Model: Borrowed Authority

We evaluate Borrowed Authority under two failure modes. In the *honest-but-confused* case, a non-root agent overstates authority because it misinterprets stale or incomplete state. In the *compromised-message* case, a non-root agent intentionally issues arbitrary natural-language subtasks to peers, including fabricated authority or evidence claims. We do not assume the attacker can forge valid root-issued leases, modify shared-state history, or impersonate sensors at the hardware level. The attacker's capability is what natural-language M2M designs already grant: composing a plausible message another agent may act on. The class is constructed rather than discovered in the wild; as of submission, no public disclosure documents a malicious skill causing physical-state harm. The construction follows from the two adjacent attack classes in Table 1.

2.3 Defining Edge Skillguard

We use *compiled* in the title to mean that high-risk Skill transitions are represented as machine-checkable guarded transitions rather than as advisory natural language; we do not claim automated synthesis from traces in this paper, and authoring guards by hand is the supported workflow until the compiler exists.

An Edge Skillguard artifact is a tuple:

$$\Pi = (Q, q_0, X, E, A, G, \delta, H, C)$$

where Q is a finite set of orchestration states; q_0 is the initial state; X is typed world state (sensors, user identity, leases, device state, time, shared-state commit); E is typed events drawn from the envelope `type` enum and `task_state` lifecycle; A is deterministic actions (Skill scripts, API calls, adapter commands); G is a set of guard predicates over $X \times E$; δ is the transition relation $Q \times E \times G \to Q \times A^*$; H is the set of bounded LLM holes; and C is a set of inter-agent contracts that incoming messages must satisfy before transition.

A guard predicate is one of seven typed operators (equality, inequality, set membership, set non-membership, collection containment, freshness window, cross-field equality) over a dot-separated path into envelope or state. Figure 2 shows the worked policy for the front-door unlock action. Borrowed Authority maps to one concrete contract violation: the message contains a permission claim (in natural language) but lacks the typed evidence the receiving policy requires (a fresh presence token, a valid lease, an authorized issuer). The artifact is a JSON Schema-validated policy file plus a pure-function evaluator that reads envelope and state, evaluates predicates, and emits either an inbox publish or a structured `policy_block` log for the failed predicates.

This object borrows from time-tested systems ideas: finite-state machines for deterministic orchestration, attribute-based access control for subject–object–operation–environment authorization, and replicated-state-machine thinking for freshness in multi-agent settings [7, 14]. It does not implement consensus or automated synthesis; it argues physical Skills need a typed, versioned, checkable policy boundary the runtime can enforce.

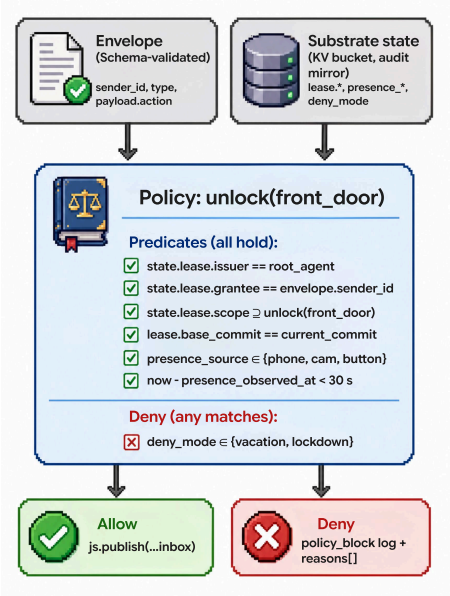

Figure 2. Edge Skillguard decision flow on the worked example policy.

2.4 What Deployers Get

Beyond the security framing, the typed authority layer reduces routine LLM calls (compiled paths execute against local state), names the missing predicate when blocked rather than producing another model rationale, and exposes Skill-quality dimensions beyond task pass rate (policy coverage, stale-evidence rejection).

3 Demonstration

We evaluate Edge Skillguard across three deployment tiers: an in-process harness that isolates evaluator cost from transport, a real NATS broker on a live edge *control-plane* testbed connected to a Home Assistant deployment of 148 entities (locks, lights, switches, cameras, fans, siren, and binary sensors), and a cross-host run over a Tailscale mesh. Measurement subjects are isolated under `test.skillguard.*` and intentionally do not trigger any device adapter; the experiment evaluates whether unauthorized transitions reach the adapter boundary, not whether a physical lock changes state. The evaluator, the policy, the schema, and the testbed-verification script all ship in the paper's artifact.[1]

3.1 Conditions and Workload

Four conditions compare the typed-guard layer against no-typed-guard upper-bound controls: **Skill** (`SKILL.md` plus deterministic scripts; the model is the only filter), **NL M2M** (a manager sends a peer a natural-language subtask carrying both instruction and an unverifiable permission claim; no receiver-side guard), **ESG-LO** (a lease-only ablation of the full policy with sensor-freshness, presence-source, and deny-condition predicates removed), and **ESG** (the full typed policy of Figure 2 over substrate state).

[1]The open-source project is available at https://github.com/zhonghaozhan/EdgeCitadel.

Table 2. Borrowed Authority evaluation. Latency is reported across 3 deployment tiers: in-process evaluator, local NATS broker on the edge testbed, and cross-host over Tailscale. Numbers measured by the testbed verification harness.

Metric	Skill	NL M2M	ESG-LO	**ESG**
Wrongful actuation (attack)	60/60	60/60	24/60	**0/60**
Benign success	60/60	60/60	60/60	**60/60**
LLM calls per decision	1	1	0	**0**
Median latency, in-process	LLM	LLM	1.5 μs	**3.2 μs**
Median latency, local broker	—	—	—	**273 μs**
Median latency, Tailscale mesh	—	—	—	**5.7 ms**
Failure reasons surfaced	0	0	lease only	**all classes**

The workload is $N = 60$ Borrowed Authority requests across $M = 5$ attack variants, 12 per variant: stale presence (timestamp > 30 s window), missing presence source (untrusted sensor), wrong-grantee lease (lease grantee ≠ authorized agent), expired lease (base-commit mismatch), and lease-scope mismatch. All variants corrupt substrate state (lease metadata or sensor readings), not the broker-attested envelope identity (§2.1). A parallel benign workload of 60 requests confirms the guard does not block legitimate actuation. We additionally run the full policy at 5× scale (300 attacks + 300 benign) on the live broker for latency stability, and a cross-host variant where publisher and broker reside on different Tailscale-connected hosts.

3.2 Result

Edge Skillguard rejects every Borrowed Authority request across all five variants on the live broker and preserves every benign request (Table 2); the 5×-scaled run on the same broker preserves both 300/300 attack rejection and 300/300 benign success at p95 399 μs, and the cross-host run over Tailscale preserves the same 300/300 correctness at p95 7.9 ms. The lease-only ablation catches 36/60 attacks, including the three lease-bound variants (wrong grantee, expired commit, scope mismatch). It lets the two sensor-bound variants (stale presence, untrusted source, 24 attacks) through, confirming each predicate class contributes distinct coverage rather than the full policy concentrating on a single check. Latency decomposes cleanly across tiers: local-broker round-trip dominates the in-process evaluator cost (~85× amplification) and overlay-mesh transport adds another order of magnitude, while the policy decision itself remains microsecond-scale and unaffected by transport. Each rejection names the predicate(s) that failed (e.g., `presence_observed_at is 187.0s old (limit 30s)`), so a blocked transition is operationally legible to incident response rather than producing another model rationale.

3.3 Boundary Cases

We additionally probe two adversarial cases that preserve all declared Skillguard predicates but violate stronger assumptions: a *compromised trusted principal* (the root agent issues an unsafe command while holding a valid lease and matching state) and *false physical evidence* (the state bucket reports a fresh whitelisted presence observation while the resident is not actually present, e.g., BLE relay, stolen phone near the door, or camera replay). Edge Skillguard allows both. This is intentional: the mechanism enforces explicit typed authority and state predicates, not arbitrary intent verification or physical-sensor truth. These failures motivate complementary defenses outside the typed-policy boundary: sensor attestation, multi-modal physical verification, and anomaly detection on the substrate audit trail.

4 Related Work and Limitations

Agent Skills. SkillsBench treats Skills as first-class procedural artifacts and reports that curated Skills raise average pass rate by 16.2 percentage points, while self-generated Skills provide no benefit on average [10]. This supports our starting point: models benefit from procedural knowledge but do not reliably author it. Edge Skillguard does not ask the model to write more advisory markdown; it defines a smaller executable boundary around physical actuation.

LLM agents for IoT. SAGE, LLMind, LLMind 2.0, IoT-GPT, and DS-IA show that LLM-based smart-home and AIoT control is an active direction [3, 4, 9, 17, 19]. LLMind is the closest neighbor because it uses FSMs and accumulated experience; LLMind 2.0 further distributes code generation through natural-language M2M. Our distinction is authority: these systems improve planning, code generation, or efficiency, but do not define typed inter-agent contracts deciding when natural-language subtasks may become physical actions. DS-IA is complementary: it uses a semantic firewall and deterministic verifier for grounded execution; Edge Skillguard defines the policy artifact such verifiers could operate against.

Deterministic workflows and trace optimization. Blueprint First, Model Second decouples workflow logic from the generative model using expert-authored blueprints [16]. Agent Workflow Optimization (AWO) analyzes workflow traces and bundles recurring tool-call subsequences into deterministic meta-tools [1]. Edge Skillguard shares the goal of reducing runtime model discretion, but targets the authority boundary for physical actuation rather than cloud workflow paths or composite tools.

Policy engines and capability systems. Edge Skillguard is related to general policy engines such as OPA/Rego [15] and to capability-based access control and tool-permission systems built into agent runtimes. The contribution is not the invention of typed authorization but its packaging and enforcement point: the policy is co-versioned with the Skill artifact and evaluates the specific transition from inter-agent message to physical action against broker-attested envelope fields and substrate state. Borrowed Authority is closely related to confused-deputy failures and capability misuse; the distinguishing feature is that the authority claim arrives as natural language inside an inter-agent Skill invocation, while the receiving Skill lacks a typed contract for accepting or rejecting it.

Security. Prompt-injection and skill-injection work shows that natural-language orchestration is an attack surface [5, 6, 8]. Systems-level studies of edge agent deployments document additional deployment-layer risks such as coordination-state divergence, failover-window exposure, and provenance-chain bypass [20]. Borrowed Authority is orthogonal to the prompt-layer attacks: even a non-jailbroken model with a well-formed `SKILL.md` can drive actuation when a peer agent supplies an unverified permission claim. Edge Skillguard addresses the inter-agent authority gap that the prompt-layer literature does not.

Limitations. This is a workshop concept paper. The guards are hand-authored, the evaluation runs on a single edge testbed across three deployment tiers (in-process, local NATS broker, Tailscale-mesh cross-host), and the paper does not include a formal information-flow checker or automated compiler. Borrowed Authority is a synthesized attack instance: as of submission, no public disclosure documents a malicious skill causing physical-state harm. We argue this is a function of timing, not impossibility, and that the precondition attacks [2, 11, 18] are sufficiently public to motivate preemptive defense rather than reactive treatment. Low-confidence holes remain LLM-mediated and must be explicitly gated. The boundary-case probes of §3.3 make the scope explicit: typed policies remove a class of authority-confusion failures, but compromised trusted principals and spoofed physical evidence remain out of scope and require complementary defenses (principal isolation, sensor attestation, anomaly detection over the substrate audit trail). The larger program is trace-inferred guards, embodied feedback, and safe live policy evolution.

5 Conclusion

Three of the four cells of the agent-attack landscape are populated by public disclosures: malicious skill artifacts in the cloud, compromised LLM controllers in physical systems, and the classical runtime compromises beneath both. We name the fourth, physical-consequence Skill compromise, and propose Edge Skillguard as the typed authority boundary that fills the two questions today's Skills format leaves advisory: who can invoke a script, and on what evidence. On a live edge testbed, it rejects every Borrowed Authority request across five attack variants and preserves every benign request; latency decomposes cleanly across deployment tiers (microsecond evaluator, sub-millisecond local broker,

single-digit-millisecond cross-host overlay). The framework is small enough to deploy today as hand-authored guards on top of existing Skills, and large enough to anchor a longer research program toward trace-driven policy synthesis with embodied feedback. The smart-home demonstration is the clearest case; the same authority gap appears in any Skill whose actions reach beyond the model. We propose the defense before, not after, the in-the-wild incident.

Acknowledgments

This work was supported by the CHIST-ERA grant CHIST-ERA-22-SPiDDS-02 (GRAPHS4SEC). This work was conducted within the Networks and Systems Lab at Imperial College London. All adversarial experiments were conducted on the authors' own hardware in a closed local network with no connection to production systems or third-party infrastructure.

References

[1] Sami Abuzakuk, Anne-Marie Kermarrec, Rishi Sharma, Rasmus Moorits Veski, and Martijn de Vos. 2026. Optimizing Agentic Workflows using Meta-tools. *arXiv preprint arXiv:2601.22037* (2026).

[2] Check Point Research. 2026. CVE-2025-59536 and CVE-2026-21852: RCE and API Token Exfiltration Through Claude Code Project Files. https://research.checkpoint.com/2026/rce-and-api-token-exfiltration-through-claude-code-project-files-cve-2025-59536/. Accessed 2026-05-01.

[3] Hongwei Cui, Yuyang Du, Qun Yang, Yulin Shao, and Soung Chang Liew. 2024. LLMind: Orchestrating AI and IoT with LLM for complex task execution. *IEEE Communications Magazine* 63, 4 (2024), 214–220.

[4] Yuyang Du, Qun Yang, Liujianfu Wang, Jingqi Lin, Hongwei Cui, and Soung Chang Liew. 2025. Llmind 2.0: Distributed iot automation with natural language m2m communication and lightweight llm agents. *arXiv preprint arXiv:2508.13920* (2025).

[5] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*. 79–90.

[6] Yinghan Hou and Zongyou Yang. 2026. SkillSieve: A Hierarchical Triage Framework for Detecting Malicious AI Agent Skills. *arXiv preprint arXiv:2604.06550* (2026).

[7] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. 2013. *Guide to attribute based access control (abac) definition and considerations (draft).* Technical Report 162. 1–54 pages.

[8] Xiaojun Jia, Jie Liao, Simeng Qin, Jindong Gu, Wenqi Ren, Xiaochun Cao, Yang Liu, and Philip Torr. 2026. Skillject: Automating stealthy skill-based prompt injection for coding agents with trace-driven closed-loop refinement. In *The 6th Workshop of Adversarial Machine Learning on Computer Vision: Safety of Vision-Language Agents*.

[9] Xinxin Jin, Zhengwei Ni, Zhengguo Sheng, and Victor Leung. 2026. Proactive Rejection and Grounded Execution: A Dual-Stage Intent Analysis Paradigm for Safe and Efficient AIoT Smart Homes. *arXiv preprint arXiv:2603.16207* (2026).

[10] Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, et al. 2026. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670* (2026).

[11] Yi Liu, Zhihao Chen, Yanjun Zhang, Gelei Deng, Yuekang Li, Jianting Ning, Ying Zhang, and Leo Yu Zhang. 2026. Malicious agent skills in the wild: A large-scale security empirical study. *arXiv preprint arXiv:2602.06547* (2026).

[12] midudev. 2026. autoskills: One command. Your entire AI skill stack. Installed. https://github.com/midudev/autoskills. GitHub repository, accessed May 2026.

[13] Nous Research. 2026. hermes-agent: The self-improving AI agent. https://github.com/nousresearch/hermes-agent. GitHub repository, accessed May 2026.

[14] Diego Ongaro and John Ousterhout. 2014. In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)*. 305–319.

[15] Open Policy Agent. 2026. Open Policy Agent: Policy-based control for cloud native environments. https://www.openpolicyagent.org/. Accessed May 2026.

[16] Libin Qiu, Yuhang Ye, Zhirong Gao, Xide Zou, Junfu Chen, Ziming Gui, Weizhi Huang, Xiaobo Xue, Wenkai Qiu, and Kun Zhao. 2025. Blueprint First, Model Second: A Framework for Deterministic LLM Workflow. *arXiv preprint arXiv:2508.02721* (2025).

[17] Dmitriy Rivkin, Francois Hogan, Amal Feriani, Abhisek Konar, Adam Sigal, Steve Liu, and Greg Dudek. 2023. Sage: smart home agent with grounded execution. *arXiv preprint arXiv:2311.00772* (2023).

[18] Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J Pappas. 2025. Jailbreaking llm-controlled robots. (2025), 11948–11956.

[19] Chaerin Yu, Chihun Choi, Sunjae Lee, Hyosu Kim, Steven Y Ko, Young-Bae Ko, and Sangeun Oh. 2026. Leveraging LLMs for Efficient and Personalized Smart Home Automation. *arXiv preprint arXiv:2601.04680* (2026).

[20] Zhonghao Zhan, Krinos Li, Yefan Zhang, and Hamed Haddadi. 2026. Systems-Level Attack Surface of Edge Agent Deployments on IoT. (2026), 99–108.