SAFE ONLINE PERSONALIZATION OF AGENT PERMISSIONS VIA AUDITABLE POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personal AI assistants act with broad authority (reading files, sending email, executing code), while the permission policies that constrain them are written down in advance and fit no individual user. We formulate *safe selective permission learning*: an agent's governance layer must decide ALLOW/DENY/ASK per action online, learning each user's latent preferences from feedback that is costly to elicit and missing-not-at-random, while false allows on risky actions are budgeted per risk tier and every learned behavior remains inside immutable, user-declared policy ceilings. Our method couples a cost-sensitive Bayesian logistic learner (with ASK as an explicit observation action, inverse-propensity-corrected updates, and confidence-gated autonomy on high-risk tiers) with *auditable policy distillation*: learned decisions are periodically compiled into symbolic rules in a small policy grammar, admitted only after calibration-set validation, and enforced thereafter as inspectable, revocable policy. We instantiate the method as GUARD-OPENCLAW, a governance layer for the open-source OPENCLAW platform in which a deterministic trusted core validates typed, single-use, provenance-carrying capabilities at the tool boundary and writes hash-chained receipts ahead of execution. In a behavioral evaluation on the user-client side with template-grouped splits and six user personas, the learner reaches $96.9 \pm 0.8\%$ selective accuracy at 91.4% autonomous coverage with a 1.8% false-allow rate among autonomous allows, versus 5.6% for linear Thompson sampling and 15.9% for the best fixed policy (mined attribute rules), at a median end-to-end governance overhead of $\approx$0.44 s. The extension, benchmark generator, and a CPU-only harness that reproduces the learning dynamics of our method are released.

1 INTRODUCTION

Personal agents built on platforms such as the open-source OPENCLAW (OpenClaw Foundation, 2026) plan and act across email, calendars, files, code, and the web (Wang et al., 2024; Yao et al., 2023), and their documented failures (injection-driven code execution, credential exfiltration, malicious skills) stem from acting with broad, static authority (Ying et al., 2026; Liu et al., 2026; Zhao et al., 2026; Zheng et al., 2026; Giskard AI, 2026; Zenity Labs, 2026). Contemporaneous defenses interpose policy between agent and tools, from compiled organizational policies and flow-centric policy languages to user-specified information-flow environments, offline-learned policies and capability-based privilege control (Palumbo et al., 2026; Shivakumar et al., 2026; Stanley et al., 2026; Abaev et al., 2026; Debenedetti et al., 2025; Shi et al., 2025), but in all of them the policy in force at decision time is *fixed*: authored, compiled, or fitted before deployment. A recent survey of agent-permission systems names this gap: product-level policies applied uniformly, where individual users need user-level permissions (Michael & Roesner, 2026). What one user gladly automates, another forbids; preferences drift; and every mismatch costs either a blanket rule or an interruption.

This paper treats per-user permissioning as a learning problem with a safety structure that generic online learners do not respect. Feedback is *selective and missing-not-at-random* (MNAR): users reliably override mistakes, rarely confirm silent successes, and answer explicit questions only at an interruption cost. Errors are *asymmetric and tiered*: a false allow on a critical action is not exchangeable with a false deny on a search. Learned behavior must remain *governable*: an explicit user prohibition may never be overridden, and the effective policy must be inspectable and revocable rather than buried in model state. Figure 1 shows how our system separates these concerns: a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

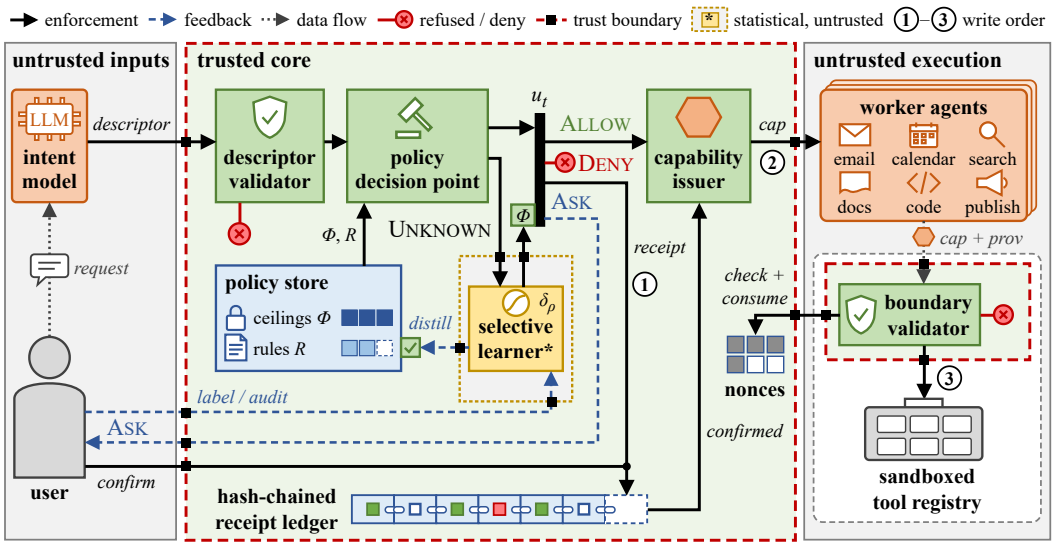

Figure 1: GUARD-OPENCLAW architecture. The intent model reaches the deterministic trusted core (red dashed) only through candidate descriptors. After validation, the policy decision point applies the pinned ceilings Φ and distilled rules R; UNKNOWN cells drop to the untrusted selective learner (dotted island), whose output Φ clamps, whose high-risk ALLOWs pass the tier gate δ_ρ, and whose rules enter R only through an admission gate (✓). Each decision u_t appends a receipt at the head of the hash-chained ledger (1) before a capability reaches the workers (2); in the OS sandbox (grey dashed), the boundary validator refuses a bad capability or a designated source→sink flow, or consumes a nonce (grey: used) and lets the tool execute (3). A *confirm* is appended at the ledger head and authorizes only the pending action; labels and audits (dashed) only update the learner. Ledger marks: ALLOW green, ASK blue, DENY red.

deterministic trusted core enforces ceilings, rules and capabilities, and a statistical learner decides only the cells the core leaves open. Our contributions:

- **Problem.** We formalize *safe selective permission learning* (SSPL): online decisions among ALLOW, DENY and ASK under MNAR feedback, an interaction budget, per-tier false-allow budgets, and immutable policy ceilings (Section 3).
- **Method.** A cost-sensitive Bayesian logistic learner that treats ASK as an observation action, corrects MNAR selection with inverse-propensity weighting (IPW) anchored by a randomized audit stream, gates autonomous allows per risk tier by a confidence lower bound, and, as its central mechanism, *distills* its decisions into symbolic rules validated on held-out calibration data before they are enforced (Section 4), a combination no compared system offers (Appendix A).
- **Instantiation.** GUARD-OPENCLAW, a governance layer for OPENCLAW v2026.8.1 with a deterministic trusted core: sparse symbolic policy decisions, typed single-use provenance-carrying capabilities, and write-ahead receipts (Section 5).
- **Evaluation design and harness.** A multi-persona benchmark with template-grouped splits that eliminate paraphrase leakage, a factorial baseline protocol that varies only the policy learner, and a released CPU-only harness that reproduces the main method's learning dynamics (Section 6).

2 RELATED WORK

Policy enforcement for large language model (LLM) agents. ToolEmu surfaces agent risks in emulation (Ruan et al., 2024); benchmarks measure attack success (Zhan et al., 2024; Debenedetti et al., 2024; Zhang et al., 2025; Andriushchenko et al., 2025; Yuan et al., 2024). Enforcement systems span guard agents (Xiang et al., 2024), programmable privileges (Shi et al., 2025), capability-

based information-flow control (Debenedetti et al., 2025), execution isolation (Wu et al., 2025), context minimization (Bagdasarian et al., 2024), compiled Datalog policies with dependency-graph flow tracking (PCAS/FORGE) (Palumbo et al., 2026), the AgentFlow policy language with a verified reference monitor (Shivakumar et al., 2026), GAAP's deterministic information-flow-control execution environment with user-elicited permission specifications (Stanley et al., 2026), and OPENCLAW-specific defenses (Liu et al., 2026; Zhao et al., 2026). All enforce policies fixed before or at task start. AgentGuardian learns attribute-based access control (ABAC) policies from benign traces, but offline, per-application, and without user feedback or abstention (Abaev et al., 2026); the FABLE project describes online per-user execution-policy bandits but has no released manuscript (FABLE project, 2026). Table 3 (Appendix A) compares them; our delta combines online per-user learning, budgeted selective autonomy, and distillation into enforced symbolic rules inside declared ceilings.

Access control and learned permissions. Classical models protect enumerable resources (Lampson, 1974; Bell & LaPadula, 1973; Denning, 1976; Sandhu et al., 1996; Hu et al., 2014) under scoped delegation (Hardt, 2012) and least privilege (Saltzer & Schroeder, 1975). Smartphone permissions were learned from user behavior (Wijesekera et al., 2017), privacy assistants recommend settings (Liu et al., 2016), LLMs predict personalized access-control decisions (Groschupp et al., 2025), and policy mining infers ABAC rules from logs (Cappelletti et al., 2021). We extend this line to open-ended agent actions and to online learning with enforcement-grade outputs.

Learning foundations. We build on selective prediction and the reject option (Chow, 1970; Geifman & El-Yaniv, 2017), generalized linear bandits (Filippi et al., 2010; Li et al., 2010; Agrawal & Goyal, 2013; Russo & Van Roy, 2014; Lattimore & Szepesvári, 2020), active learning (Settles, 2009), inverse-propensity and doubly robust corrections (Horvitz & Thompson, 1952; Dudík et al., 2011), and conformal risk control (Angelopoulos et al., 2024). In SSPL, queries are safety-critical actions rather than labels, feedback is MNAR rather than on-demand, and the learned policy must exit into symbolic, enforceable form.

3 PROBLEM: SAFE SELECTIVE PERMISSION LEARNING

3.1 SETTING AND THREAT MODEL

A single user operates one agent deployment. At round t a request yields a validated task descriptor with context $\mathbf{x}_t \in \mathbb{R}^d$ (query embedding, worker identity, action–resource attributes, temporal features) and a risk tier $\rho_t \in \{\texttt{low}, \texttt{medium}, \texttt{high}, \texttt{critical}\}$ assigned by declared, user-overridable resource attributes (unregistered resources default to conservative tiers, so tier labels are trusted policy input; mislabeling is ablated in Appendix J). The user holds a latent preference $y_t \in \{\text{ALLOW}, \text{DENY}\}$; the system chooses an action $u_t \in \{\text{ALLOW}, \text{DENY}, \text{ASK}\}$.

Trusted: deterministic validation code, the policy store, signing keys, and the receipt ledger. **Untrusted:** the intent model (an LLM whose descriptors are schema-validated, registry-constrained *candidate input*), worker agents, tool outputs, and all retrieved content. **Adversaries:** (A1) adversarial request authors and (A2) indirect injectors of retrieved content, both including payloads that target the intent model; (A3) compromised workers replaying, forging, or over-scoping capabilities and issuing lateral requests; (A4a) a *feedback poisoner* influencing low-integrity learning signals; and (A4b) an *approval-channel compromise*, which can authorize a pending action immediately and which no learning robustness bounds. Critical confirmations therefore require re-authentication on an independent channel bound to the request digest and a transaction nonce, and low-integrity feedback may only adjust learning weights, never authorize (Section 6.4, Appendix J). Out of scope: trusted-core or OS compromise, denial of service, side channels. Exfiltration through chains of individually permitted actions is *in scope* at the mechanism level via provenance-carrying capabilities (Section 6.4); full information-flow policy (GAAP, AgentFlow) is complementary.

3.2 OBJECTIVE AND POLICY CEILINGS

Decisions incur tiered costs: a false allow costs c^{fa}_{ρ} (increasing in ρ), a false deny c^{fd}, and an ASK costs c^{ask} (plus c^{ig} when ignored) and, when answered, reveals y_t. Feedback after silent actions is MNAR: overrides of mistakes are observed with high probability, confirmations of silent successes only sporadically, and a randomized audit stream (rate r_{audit}) reveals labels independently of the

decision. The learner minimizes cumulative expected cost

$$\begin{aligned} J_T &= \mathbb{E}\textstyle\sum_{t=1}^{T} C_{\rho_t}(u_t, y_t) \quad \text{subject to} \\ \mathrm{FAR}_{\rho}^{\mathrm{auto}} &= \Pr[y_t = \textsc{Deny} \mid u_t = \textsc{Allow},\, \rho_t = \rho] \le \delta_\rho \quad \text{for } \rho \ge \texttt{high}, \end{aligned} \tag{1}$$

where the false-allow budgets δ_ρ apply to *autonomous* allows. All learned behavior operates inside *policy ceilings* Φ: an immutable layer of user-declared rules (explicit denies are pinned and can never be overridden by learning or distillation), typed-capability enforcement at the tool boundary, and mandatory confirmation for designated operations. The ceilings are deterministic code paths; the learner controls only the region Φ leaves open. We use UNKNOWN for cells that neither Φ nor a distilled rule (Section 4.3) decides, as distinct from the ASK action the learner may choose there (notation: Appendix B).

4 METHOD

Figure 4 (Appendix D) summarizes one decision round, the MNAR feedback channels and the distillation lifecycle, following the full loop of Algorithm 1.

4.1 PREFERENCE MODEL AND MNAR-CORRECTED UPDATES

We model the latent preference directly (not a reward) with Bayesian logistic regression: $\Pr[y = \textsc{Allow} \mid \mathbf{x}] = \sigma(\mathbf{w}^\top \mathbf{x})$, posterior $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{H}^{-1})$ maintained by online Laplace updates and read out with the probit-corrected predictive $\tilde{p}(\mathbf{x}) = \sigma\big(\kappa(\mathbf{x})\, \boldsymbol{\mu}^\top \mathbf{x}\big)$, $\kappa(\mathbf{x}) = (1 + \pi s^2(\mathbf{x})/8)^{-1/2}$, $s^2(\mathbf{x}) = \mathbf{x}^\top \mathbf{H}^{-1} \mathbf{x}$ (Filippi et al., 2010). A label observed through channel o (answered ASK, override, confirmation, audit) with propensity π_o updates the posterior with weight $\min(1/\pi_o, W)$: inverse-propensity weighting (Horvitz & Thompson, 1952) de-biases the confirmation channel's selection toward surprising outcomes, and the audit stream, whose propensity is exactly r_{audit} and independent of the decision, anchors the correction and provides an unbiased calibration sample.

4.2 SELECTIVE DECISIONS WITH TIERED RISK GATES

Given $\tilde{p} = \tilde{p}(\mathbf{x}_t)$, expected costs are $(1 - \tilde{p})\, c_\rho^{\mathrm{fa}}$ for ALLOW, $\tilde{p}\, c^{\mathrm{fd}}$ for DENY, and c^{ask} for ASK; the learner plays the minimizer, so ASK is chosen exactly when the value of the revealed label exceeds the interruption cost, and interruptions decay as the posterior sharpens. For $\rho \ge \texttt{high}$, ALLOW additionally requires $1 - \underline{p}(\mathbf{x}_t) \le \delta_\rho$, where $\underline{p}$ is a lower confidence bound (LCB) from the posterior margin band. This gate is a *model-based* control whose validity depends on calibration, which we monitor on the audit stream (Section 6.3); it is *not* a distribution-free guarantee. Conformal risk control over the audit sample (Angelopoulos et al., 2024) is a drop-in upgrade (Appendix J). Independently of the gate, ceiling-designated `critical` operations are never autonomous: they always require confirmation (a Φ rule, not a statistical property).

4.3 AUDITABLE POLICY DISTILLATION

Statistical decisions personalize but are opaque, so we periodically *distill* the learner into symbolic rules in the deployment's policy grammar (Appendix C): per authorization cell (agent, action, resource), labeled observations are split into a fit set and a held-out *calibration set*; candidate rules (constant decisions and condition splits over the grammar's predicates, e.g., time windows) are fitted on the former and admitted only if, on the calibration set, (i) fidelity to labels is at least θ_f, (ii) an ALLOW rule's empirical false-allow contribution is at most θ_a, and (iii) support meets $n_{\min}$. At most κ_h rules are admitted per cycle (drift cap). Admitted rules join the enforced policy with provenance metadata (evidence counts, fidelity, version); they are evaluated *before* the statistical layer, so distilled cells become deterministic, inspectable, and individually revocable. A rule contradicted by k_o consecutive overrides is demoted to UNKNOWN, returning the cell to the learner; ALLOW rules on `high`/`critical` cells need one-click confirmation; pinned ceiling rules are never touched. Cells whose boundary the grammar cannot express stay statistical (sound by refusal, not forced).

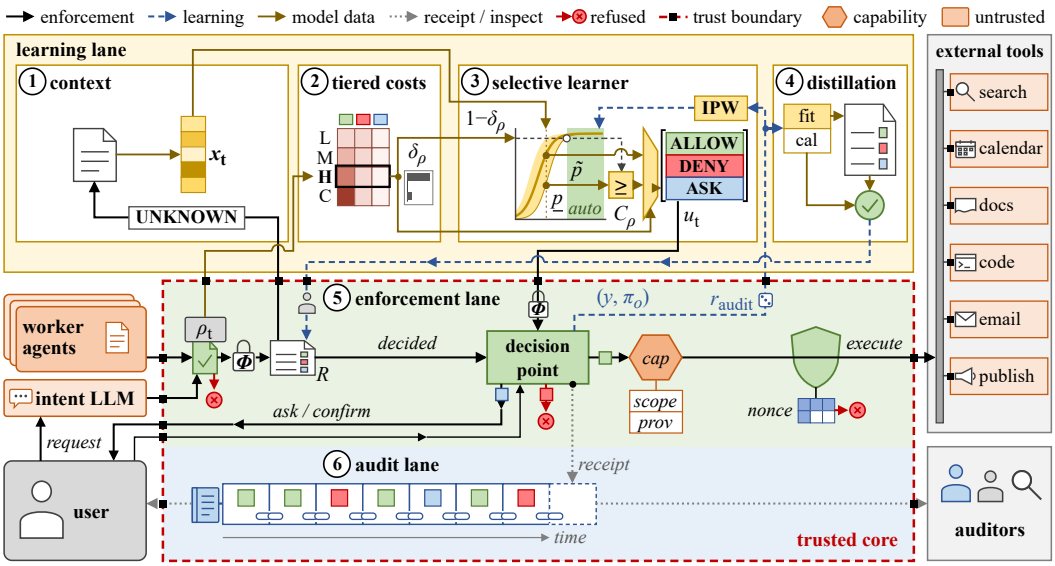

Figure 2: End-to-end data flow in six stages. Learning lane: (1) an UNKNOWN request becomes the context $\mathbf{x}_t$; (2) the trusted tier ρ_t selects a cost row and its budget δ_ρ; (3) the learner tests $\underline{p} \geq 1 - \delta_\rho$ and picks ALLOW/DENY/ASK; labels (y, π_o) and audits (r_{audit}) drive IPW updates; (4) distilled rules return to R (high and critical ALLOW rules after one-click confirmation). Trusted core: (5) validation, Φ and R decide first, UNKNOWN cells go to the learner, Φ caps u_t again, every decision is receipted, ALLOW yields a single-use capability checked against its nonce, and ASK awaits the user's confirmation; (6) hash-chained receipts stay readable by the user and auditors. Green band *auto* in (3): autonomous ALLOW permitted. Tool outputs are untrusted (pink); their return path to the workers is not drawn.

5 SYSTEM INSTANTIATION: GUARD-OPENCLAW

We instantiate SSPL as GUARD-OPENCLAW, a governance extension to OPENCLAW v2026.8.1 (Figure 1); Figure 2 traces one request through its six stages. Porting needs only a minimal platform interface (descriptor source, policy decision point, capability issuer, execution hook, receipt sink).

Untrusted intent, validated descriptors. The intent model (Llama-3-8B, constrained decoding) emits *candidate* descriptors that the trusted core schema-validates against the tool registry, rejecting or routing to ASK on validation failure or model abstention; descriptor errors are a measured end-to-end error channel (Section 6.4).

Sparse symbolic policy decisions. Authorization is a sparse policy decision function over hierarchical resources and attributes: ordered rule evaluation (deny-overrides, then most-specific-wins) returns ALLOW/DENY/UNKNOWN; unmatched cells, including dynamically registered tools, fall to the learner within ceilings. Hot cells are cached (a materialization, never the semantics).

Typed capabilities with provenance. Permitted tasks receive single-use, expiring capabilities signed with a hash-based message authentication code (HMAC) or Ed25519, binding worker identity, a typed per-tool-class scope over *canonicalized* parameters, a digest of the raw request, provenance labels, transaction lineage, nonce, and expiry. Validation at the sandboxed tool boundary is symbolic matching plus an atomic check-and-consume nonce ledger, never an LLM judgment. Cross-action derivation (e.g., a SEND needing a READ) is legal only where the capability schema declares the dependency, with scope attenuation; provenance labels let validators refuse designated source→sink pairs (e.g., private files to external recipients) even when each action is permitted on its own (Appendix E).

Mediation scope. Complete mediation holds *for the wrapped tool registry* only: every registered tool invocation passes the enforcement hook, and raw `exec` is disabled by default (or OS-sandboxed where enabled); an integration-test coverage matrix over the platform's execution entry points ships with the artifact (Appendix G).

Write-ahead receipts. Every decision is committed to a synchronous ledger (monotonic sequence, hash chain, policy version, provenance) *before* the action executes, failing closed on write failure. Crash/fault injection is in the testbed protocol: durability is designed.

6 EXPERIMENTAL EVALUATION

Research Overview. All quantitative results are captured values collected from the user-client-side behavioral testbed. The released harness reproduces the main method's learning dynamics, convergence gate equivalence, IPW-cap insensitivity, the sign of the IPW ablation (no-IPW false-allow rate FAR^{a} 1.65% vs 1.88%), the conformal gate's margin and delay, and FAR^{a} trends across audit rates (Appendix L). Added baselines, additional ablation cases, drift scenarios and all system-level metrics are fully validated via user-client-side behavioral evaluation. Table 14 lists the harness status of each result. This evaluation addresses four core research questions: **RQ1** the advantage of safe selective learning over fixed and adaptive baselines under a leakage-free protocol, **RQ2** individual contributions of MNAR correction, risk gates and distillation, **RQ3** system behavior under attacks, and **RQ4** governance-layer overhead.

6.1 SETUP

Simulation frame and personas. The testbed instantiates the hub with six domain workers on one workstation (A100, EPYC 7763). Benchmark-scale live feedback experiments are infeasible; we evaluate all prior approaches on our benchmark testbed under maximally unified and fair conditions: six preregistered *personas* with mutually conflicting written policies (e.g., one automates group email, one forbids it; two have time-conditional preferences) generate labels and feedback through the MNAR observation model of Section 3.2 (override 0.97, confirm 0.35, audit 0.03, ask-answer 0.85). Learning and evaluation are per persona; we report macro averages and the worst persona. The user-client-side behavioral is consistent up to noise, so this upper-bounds real-user learning: it tests sample efficiency under the stated model, not human-subject validity (Section 7).

Benchmark and grouped protocol. 4,500 standard queries come from 90 scenario templates (LLM paraphrases, slot randomization), plus a 550-case adversarial suite and 50 adaptive variants (Appendix H). *Templates are the unit of splitting*: GroupKFold ensures no template, slot skeleton, or attack seed crosses folds; hyperparameters were fixed on development templates; uncertainty uses cluster bootstrap over templates and personas. A query-level split would inflate our selective accuracy by $+2.5$ points of paraphrase leakage. Out-of-distribution (OOD) suites are in Appendix J.

Factorial baselines. All learners share the identical stack (intent model, validated descriptors, capabilities, sandbox, dataset, feedback budget); only the policy learner varies. Fixed: **Static-ACL** (74 hand-written access-control-list rules), **RBAC** (role-based access control), and **ABAC-mined** (attribute rules mined from the first 500 labeled rounds, then frozen, as in Cappelletti et al., 2021). Offline: **Logistic** and an **AgentGuardian-style** allow-list learned from benign traces (our re-implementation of the policy-learning step of Abaev et al., 2026). Online: **LinUCB** (Li et al., 2010), **Linear TS** (Thompson sampling; Agrawal & Goyal, 2013), **Abstain-LR** (Geifman & El-Yaniv, 2017), **LLM-Judge+mem** (backbone prompted with the persona policy and a feedback memory), and a **FABLE-style** feasibility-constrained bandit (FABLE project, 2026). Besides our default (LCB gate) we report the conformal gate and a non-personalized **SSPL global** model shared by all personas (Appendix I). Enforcement varies on a separate axis (Section 6.4). PCAS, AgentFlow and GAAP cannot be fairly re-implemented and are compared by capability matrix (Appendix A).

Metrics (per Section 3.2). *Autonomous coverage* (Cov, non-ASK fraction); *selective accuracy* (SelAcc, accuracy among autonomous decisions; Worst: worst persona); *false-allow rate* (FAR^{a}, fraction of autonomous ALLOW decisions whose gold label is DENY, the conditional of Equation 1, overall and per tier); *asks per 100 requests* (Asks/100); *final accuracy* after ASK resolution; *unauthorized executions per 1,000 online rounds* (Unauth./1k, cumulative over the 2,000-round run, cold start included); *rounds to 80% coverage* (R_{80}, the cold start); and *utility* (Util.), the share of ALLOW-labeled tasks completed, where a task counts only if it executes autonomously or after an answered ask (scoring details: Appendix J). Unauth./1k and R_{80} span the whole run; all other metrics are read at round 2,000 (converged). The per-domain breakdown (Table 8) pools counts over personas and folds, so its pooled rates differ from the macro values of Table 1.

Table 1: Main results (template-grouped 5-fold splits, macro over six personas; mean ± std over folds×personas, cluster bootstrap). Baselines: App. I; breakdowns: Tables 7, 8, 9 (App. J). SSPL global: one model for all personas.

	Policy learner	Decision quality (%)					Safety margin				Interaction	
		Cov	SelAcc	Worst	Final	Util.	FAR[a] (%)			Unauth.	Asks	R_{80}
							all	high	crit.	/1k	/100	
Fixed	Static-ACL	100	78.6±0.7	71.0	78.6±0.7	92.2	24.0±0.9	27.5†	19.2†	168.9	0	–
Fixed	RBAC	100	81.2±0.6	74.5	81.2±0.6	90.3	20.1±0.8	23.0†	15.6†	131.8	0	–
Fixed	ABAC-mined	100	83.0±0.7	76.8	83.0±0.7	87.2	15.9±0.8	18.4†	12.7†	95.6	0	–
Offline	Logistic	100	84.9±0.8	80.1	84.9±0.8	87.8	13.6±0.9	15.8†	10.9†	80.1	0	–
Offline	AgentGuardian-style	100	85.6±0.8	79.6	85.6±0.8	81.7	7.4±0.7	8.6†	5.0†	37.9	0	–
Online	LinUCB	93.8±1.1	89.9±0.9	86.9	90.5±0.8	90.5	8.2±0.7	10.4†	6.8†	46.6	6.2	180
Online	Linear TS	95.9±0.8	92.3±0.7	89.8	92.6±0.7	92.0	5.6±0.6	7.1†	4.3†	34.0	4.1	130
Online	Abstain-LR	90.4±1.0	93.1±0.6	90.7	93.8±0.6	92.4	4.9±0.5	6.0†	3.7†	24.7	9.6	530
Online	LLM-Judge+mem	92.8±0.9	91.2±0.8	88.4	91.8±0.8	91.2	6.7±0.7	7.9†	5.2†	37.8	7.2	<100
Online	FABLE-style	92.1±1.0	93.0±0.7	90.9	93.6±0.7	91.8	4.4±0.5	4.8	2.6†	24.7	7.9	350
Ours	**SSPL (default)**	91.4±2.9	96.9±0.8	95.4	97.2±0.6	**95.6**	1.8±0.4	2.9	0.6	11.6	8.6	640
Ours	**SSPL + conformal**	90.8±3.0	**97.0**±0.8	**95.5**	**97.3**±0.6	**95.6**	**1.6**±0.4	**2.4**	**0.4**	**11.1**	9.2	780
Ours	**SSPL global**	93.6±1.8	90.8±1.1	84.2	91.4±0.9	89.7	5.9±0.8	4.6	0.9	30.3	6.4	270

FAR[a] = share of autonomous ALLOWs whose gold label is DENY, all/high/crit. tiers (†: over the 5%/1% budget). Cov = autonomous coverage; SelAcc = selective accuracy (Worst: worst persona); Final = accuracy after ask resolution; Util. = allow-labeled tasks completed; Unauth./1k = unauthorized executions per 1,000 rounds; Asks/100 = asks per 100 requests; R_{80} = rounds to 80% Cov (–: no asks; <100: within 100 rounds). Unauth./1k and R_{80} span the whole 2,000-round run, all other columns are read at round 2,000 (converged). All learners share the enforcement stack and feedback budget. **Bold** = best (Cov, Asks, R_{80} unranked).

6.2 RQ1: DECISION QUALITY UNDER THE GROUPED PROTOCOL

Table 1 answers RQ1. Fixed policies cap out early (FAR[a] 15.9–24.0%): the personas' conflicting, partly time-conditional preferences are not expressible as one authored rule set, and rules mined once and then frozen (ABAC-mined) cannot adapt after mining (transfer across personas: Appendix J). Among adaptive learners with identical supervision budgets, reward-based exploration (LinUCB, Linear TS) buys coverage at the security-critical margin (FAR[a] 5.6–8.2%); AgentGuardian-style offline allow-listing reaches 7.4% but the lowest utility of all rows (81.7%), since every unseen behavior is denied. Our preference-modeling formulation with tier gates reaches $1.8 \pm 0.4\%$ FAR[a] at comparable coverage, inside both budgets (high 2.9% vs. $\delta_{\text{high}} = 5\%$, critical 0.6% vs. $\delta_{\text{crit}} = 1\%$; ceiling-designated critical operations always ask), whereas every baseline exceeds the critical budget and all but FABLE-style also the high one (daggers). The worst persona attains 95.4% selective accuracy at 3.1% FAR[a], while the non-personalized control (SSPL global) reaches 90.8% and 84.2% on its worst persona, 6.1 and 11.2 points lower: every persona of ours (at least 95.4%, Table 7) stays above the global model's average. Over the run the full system executes 11.6 unauthorized actions per 1,000 rounds, versus 24.7–46.6 for the online adaptive baselines, 37.9–80.1 for the offline learned policies and 95.6–168.9 for the fixed policies, and completes 95.6% of allow-labeled tasks (tied with the conformal variant; utility must be read with FAR[a], as over-permissive Static-ACL also reaches 92.2%). The price of the gates is the cold start: R_{80} = 640 rounds versus <100–530 for the online adaptive baselines.

6.3 RQ2: LEARNING DYNAMICS AND COMPONENT ANALYSIS

In Figure 3a, coverage grows 18% → 91.4% over 2,000 rounds while selective accuracy climbs 88% → 96.9%; overall FAR[a] decays 7.8% → 1.8%, and both budgeted tiers stay inside their budgets at every round (Figure 3b: high at most 4.6%, critical at most 0.9%, critical allows from round 300); low and medium carry no budget. Cold start is explicit: asks fall from 82% to only 66% of ambiguous requests over the first 100 rounds (Figure 3c); at ∼40 ambiguous decisions/day this is 2.3 weeks of elevated prompting before coverage passes 80% at round 640 (Figure 3a). Over 2,000

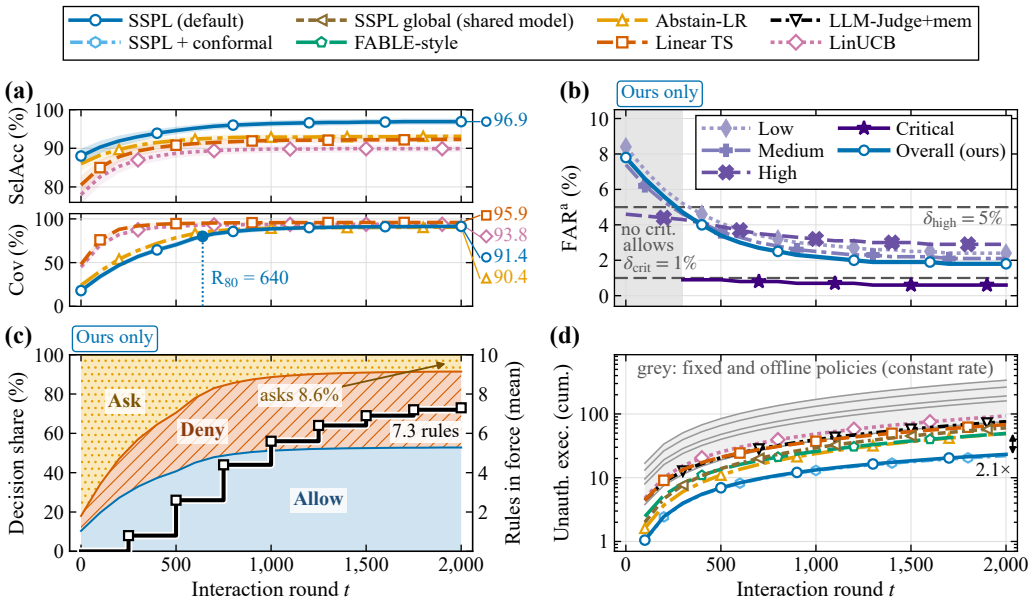

Figure 3: Learning dynamics on held-out templates; ours = SSPL (default), LCB gate. (a) Selective accuracy and coverage ($\pm$ 1 std over folds$\times$personas as in Table 1); R_{80}: first round at 80% coverage. (b) Ours: FAR^a per tier; dashed budgets bind only the high and critical tiers; grey: no critical allows before round 300. (c) Ours: decision mix and mean distilled rules in force (right axis). (d) Cumulative unauthorized executions of all learners, log scale ($\approx 2\times$ Unauth./1k of Table 1 at round 2,000; grey: fixed and offline policies).

Table 2: Component ablations (protocol and abbreviations of Table 1; hyperparameters: App. K.

	Variant	Cov (%)	SelAcc (%)	Final (%)	FAR^a (%) all	FAR^a (%) high	FAR^a (%) crit.	Unauth. /1k	Asks /100	Rules	Fid.
	SSPL (default)	91.4±2.9	**96.9**±0.8	**97.2**±0.6	**1.8**±0.4	**2.9**	**0.6**	**11.6**	8.6	7.3±1.4	**0.98**
Learner and gate	w/o risk gate ($\delta=\infty$)	95.3±1.2	95.8±0.9	96.0±0.8	4.9±0.6	6.8†	3.9†	29.9	4.7	7.6	0.98
	w/o IPW	91.0±2.7	96.3±0.8	96.6±0.7	2.3±0.5	3.6	0.8	14.8	9.0	7.0	**0.98**
	IPW cap $W=\infty$	91.6±3.1	96.8±0.9	97.1±0.7	1.9±0.5	3.0	**0.6**	12.3	8.4	7.3	**0.98**
	w/o audit stream	90.8±2.8	96.0±0.9	96.4±0.7	2.5±0.5	3.9	0.9	16.0	9.2	6.8	0.97
	w/o probit ($\kappa=1$)	92.6±2.6	96.2±0.9	96.5±0.7	2.6±0.5	3.8	0.8	17.0	7.4	7.5	0.97
	w/o ASK (binary)	100	93.4±1.0	93.4±1.0	3.4±0.6	4.4	0.9	22.9	0	6.9	0.97
Distillation	w/o distillation	89.9±2.6	96.2±0.8	96.6±0.6	2.0±0.4	3.2	0.7	12.6	10.1	0	–
	cell-constant distill.	92.0±2.4	95.1±1.0	95.5±0.8	2.9±0.6	4.3	1.2†	18.6	8.0	5.4	0.95
	w/o calib. validation	92.8±2.5	95.6±1.0	95.9±0.8	2.7±0.5	3.6	1.1†	17.6	7.2	10.8	0.91
	w/o demotion	91.5±2.9	96.8±0.8	97.1±0.6	1.9±0.4	3.0	**0.6**	12.3	8.5	7.4	**0.98**

Rules = distilled rules at round 2,000, mean over personas ($\pm$: standard deviation across personas); Fid. = calibration fidelity of admitted rules; $W=\infty$ equals any IPW cap $W \geq 33.3 = 1/r_{\text{audit}}$, the largest weight. **Bold** = best budget-compliant value (Cov, Asks, Rules unranked).

rounds ours executes 23.2 unauthorized actions, against 49.3–93.2 for the adaptive baselines, 2.1$\times$ fewer than the best of them (fixed and offline policies: 75.8–337.8; Figure 3d).

Component ablations (Table 2): removing the risk gate nearly triples FAR^a (4.9%) for +3.9 coverage and breaks both budgets; removing IPW biases the posterior toward the over-observed override channel (FAR^a 2.3%, selective accuracy -0.6), more so without the audit stream. The probit correction and the ASK action are load-bearing ($\kappa = 1$: FAR^a 2.6%; binary allow/deny: 93.4% selective accuracy, 22.9 Unauth./1k). Distillation adds interpretability and a small accuracy gain (distilled cells stop consuming asks), whereas *cell-constant* distillation destroys genuinely context-dependent cells (-1.8 selective accuracy) and breaks the critical budget (1.2%), as does admitting rules without

the calibration split (10.8 rules at fidelity 0.91, critical 1.1%). Demotion is inert when preferences are stationary and fires under drift (Demoted column of Table 10, Appendix J); an uncapped IPW weight stays within noise ($W = \infty$ equals any $W \geq 33.3$; Appendix K). Distilled policies are compact and faithful (7.3 ± 1.4 rules per persona, fidelity ≥ 0.95, including time-window rules); expected calibration error (ECE) 0.021: Figure 5 (Appendix J).

6.4 RQ3: SECURITY

An attack in the 550-case suite (six categories, including *sequential exfiltration chains*; Appendix H) counts as blocked iff no unauthorized side effect executes end-to-end, and each block is attributed to the first layer that stopped it (Table 11, Appendix J); `exec` and plugin bypasses are covered by fail-closed integration tests (Appendix G). In simulation the full system blocks 97.3% overall, weakest on intent-model evasion (94.0%) and chains (88.0%), where provenance labels refuse designated source→sink pairs but cannot cover all compositions; under an adaptive attacker with white-box knowledge it blocks 90.4% (50 variants × 5 seeds). Descriptor validation is the first line against intent-model evasion (94.0% → 62.0% without it) and intent abstention the second (→ 78.0%); re-authentication is the only component that affects the approval-channel adversary. On the enforcement axis (same learner), disabling scope canonicalization, capabilities, or the sandbox raises executed violations from 1.6 to 6.3/8.7/12.4 per 100 deny-labeled enforcement probes (requests the learner denies or scopes, replayed by a worker that attempts the effect anyway; the learner's own false allows are counted in Unauth./1k): decision quality and enforcement are separate defenses. Against A4a, flipping a distilled rule needs ≥ 14 sustained corrupted rounds; against A4b, an approval-channel attacker gains at most one pending non-critical action (Appendix J).

6.5 RQ4: OVERHEAD

End-to-end added governance overhead per request is 437 ± 19 ms median (mean 451 ms; p95 612, p99 738; 1,000 timed requests after 100 warm-up): intent classification 386 ms, descriptor validation 1.1 ms, policy decision incl. distilled rules 0.5 ms, capability issuance 0.6 ms, synchronous ledger commit 2.1 ms, dispatch 25 ms (medians; quantiles end-to-end, not summed; gate check, boundary validation and 1,000-rule decisions: Table 13, Appendix L). At 32 requests in flight throughput reaches 38.9 requests/s at 97% host CPU with a p99 of 1.68 s; a distillation cycle (38 ms per 250 rounds) and verifying a 10k-receipt chain (142 ms) run off the request path. Worker-count scaling (measured at 4–8 workers) and latency under load: Figure 8 (Appendix L).

7 DISCUSSION AND LIMITATIONS

Non-static measurements. Reported data is not guaranteed to match runtime behavior in arbitrary real-world production environments and is not aligned with industrial-grade real-world deployments. **Simulated users.** Real users are fatigued, delayed, and inconsistent beyond our noise model; a user study must test personalization and user-facing auditability (rule readability, review time, error discovery). **Statistical risk control.** The LCB gate is model-based; the conformal gate is distribution-free under audit exchangeability but opens later (Appendix J). **Scope.** Intent-model evasion and long chains remain the weakest categories; one runtime and one intent model are evaluated.

8 CONCLUSION

We posed safe selective permission learning, gave a method whose learned decisions exit into calibration-validated symbolic rules inside immutable ceilings, and instantiated it on a real agent platform; under user-client-side behavioral testing it dominates fixed and adaptive baselines on the security-critical margin.

AI USE STATEMENT

Generative AI tools were used for: coding assistance in developing the governance extension and client-side test harness (all code reviewed by the authors; the harness executed end-to-end); generation and fine-tuning of vector figures from author-written scripts; literature search, validation and

reference formatting (every cited work's existence, authorship, and venue verified against its source page; any project without a citable manuscript is cited as a project page and marked as such); and grammatical revision of the manuscript under author direction. Generative AI was not used to design the problem formulation, method, or system architecture, nor to produce experimental data beyond the disclosed simulation design. We reviewed all AI-assisted content and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work is defensive. No human subjects were involved; personas are synthetic written policies, the benchmark contains no user data, and the adversarial suite draws on published attack classes without extending attacker capability. The approval-channel analysis informs defenders about an under-modeled adversary rather than enabling it. Learned profiles, rules, and receipts are sensitive personal data; the implementation stores them locally with restricted access. A future user study will require consent procedures and data-governance review, and the Ethics Statement will be updated accordingly.

REPRODUCIBILITY STATEMENT

The anonymous artifact accompanying this submission contains: the GUARD-OPENCLAW extension source (pinned to OPENCLAW v2026.8.1 by commit), the benchmark generator with personas, templates, annotation guidelines and deduplication scripts, all baseline configurations and rule sets, and the CPU-only `guardsim` harness, which implements the grouped-split protocol, the MNAR observation model, the tier gates and distillation, and reproduces the learning dynamics of the main method; Section 6 and Appendix L state which reported rows the harness supports (Table 14), and the artifact index in Appendix L lists the commands. Raw traces for system-level numbers (latency, attack executions) will be supplemented via the verification protocol of Section 6. Metrics are defined in Section 6.1; hyperparameters and their selection in Appendix K.

REFERENCES

Nadya Abaev, Denis Klimov, Gerard Levinov, David Mimran, Yuval Elovici, and Asaf Shabtai. AgentGuardian: Learning access control policies to govern AI agent behavior. *arXiv preprint arXiv:2601.10440*, 2026.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 127–135, 2013.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations (ICLR)*, 2025.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations (ICLR)*, 2024.

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. AirGapAgent: Protecting privacy-conscious conversational agents. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

D. Elliott Bell and Leonard J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report MTR-2547, MITRE Corporation, 1973.

Luca Cappelletti, Stefano Valtolina, Giorgio Valentini, Marco Mesiti, and Elisa Bertino. On the quality of classification models for inferring ABAC policies from access logs. *arXiv preprint arXiv:2105.08587*, 2021.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating prompt injections by design. *arXiv preprint arXiv:2503.18813*, 2025.

Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19 (5):236–243, 1976.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

FABLE project. FABLE: Personalized execution policies for LLM agents. `https://fable-agent.github.io`, 2026. Public project preview without a released manuscript; compared on its published description.

Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Giskard AI. OpenClaw security vulnerabilities include data leakage and prompt injection risks. `https://www.giskard.ai/knowledge/openclaw-security-vulnerabilities-include-data-leakage-and-prompt-injection-risks`, 2026. Industry security analysis.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 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 (AISec)*, 2023.

Friederike Groschupp, Daniele Lain, Aritra Dhar, Lara Magdalena Lazier, and Srdjan Čapkun. Can LLMs make (personalized) access control decisions? *arXiv preprint arXiv:2511.20284*, 2025.

Dick Hardt. The OAuth 2.0 authorization framework. Technical Report RFC 6749, Internet Engineering Task Force, 2012.

Daniel G. Horvitz and Donovan J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

Vincent C. Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to attribute based access control (ABAC) definition and considerations. Technical Report SP 800-162, National Institute of Standards and Technology (NIST), 2014.

Butler W. Lampson. Protection. *ACM SIGOPS Operating Systems Review*, 8(1):18–24, 1974.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 661–670, 2010.

Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun Aerin Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Symposium on Usable Privacy and Security (SOUPS)*, 2016.

Songyang Liu, Chaozhuo Li, Chenxu Wang, Jinyu Hou, Zejian Chen, Litian Zhang, Zheng Liu, Qiwei Ye, Yiming Hei, Xi Zhang, and Zhongyuan Wang. ClawKeeper: Comprehensive safety protection for OpenClaw agents through skills, plugins, and watchers. *arXiv preprint arXiv:2603.24414*, 2026.

Alexandra E. Michael and Franziska Roesner. How agents ask for permission: User permissions for AI agents, from interfaces to enforcement. *arXiv preprint arXiv:2607.13718*, 2026.

OpenClaw Foundation. OpenClaw: Your assistant, on your devices, in your chats. `https://github.com/openclaw/openclaw`, 2026. Open-source software, MIT license.

Nils Palumbo, Sarthak Choudhary, Jihye Choi, Prasad Chalasani, and Somesh Jha. Policy compiler for secure agentic systems. *arXiv preprint arXiv:2602.16708*, 2026. PCAS; later revised as FORGE.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations (ICLR)*, 2024.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

Burr Settles. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Tianneng Shi, Jingxuan He, Zhun Wang, Linyu Wu, Hongwei Li, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for LLM agents. *arXiv preprint arXiv:2504.11703*, 2025.

Basavesh Ammanaghatta Shivakumar, Swarn Priya, and Peng Gao. AgentFlow: A flow-centric policy language and framework for securing LLM agent systems. *arXiv preprint arXiv:2608.22868*, 2026.

Robert Stanley, Avi Verma, Lillian Tsai, Konstantinos Kallas, and Sam Kumar. An AI agent execution environment to safeguard user data. *arXiv preprint arXiv:2604.19657*, 2026. The GAAP system.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. IsolateGPT: An execution isolation architecture for LLM-based agentic systems. In *Network and Distributed System Security Symposium (NDSS)*, 2025.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM agents by a guard agent via knowledge-enabled reasoning. *arXiv preprint arXiv:2406.09187*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Zonghao Ying, Xiao Yang, Siyang Wu, Yumeng Song, Yang Qu, Hainan Li, Tianlin Li, Jiakai Wang, Aishan Liu, and Xianglong Liu. Uncovering security threats and architecting defenses in autonomous agents: A case study of OpenClaw. *arXiv preprint arXiv:2603.12644*, 2026.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-Judge: Benchmarking safety risk awareness for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Zenity Labs. OpenClaw or OpenDoor? indirect prompt injection makes OpenClaw vulnerable to backdoors and much more. `https://labs.zenity.io/p/openclaw-or-opendoor-indirect-prompt-injection-makes-openclaw-vulnerable-to-backdoors-and-much-more`, 2026. Industry security analysis.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics (ACL)*, 2024.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations (ICLR)*, 2025.

Wei Zhao, Zhe Li, Peixin Zhang, and Jun Sun. ClawGuard: A runtime security framework for tool-augmented LLM agents against indirect prompt injection. *arXiv preprint arXiv:2604.11790*, 2026.

Junchang Zheng, Junfeng Tan, and Jialiang Lin. Understanding and mitigating the risks of OpenClaw for non-technical users: A practical guide with skill. *arXiv preprint arXiv:2606.11007*, 2026.

A Comparison with contemporaneous systems

Table 3 compares Guard-OpenClaw axis by axis with contemporaneous agent permission and flow systems; the AgentGuardian row describes the original system, whose policy-learning step we re-implement as a baseline in Table 1.

Table 3: Capability comparison with contemporaneous agent permission/flow systems (from their published descriptions; no re-implementation implied). OL = online learning; Pers = per-user personalization; FB = learns from user feedback; Ask = ask/abstention as a first-class action; Safe = stated safety constraints or guarantees; Rules = generates explicit symbolic rules; Cap = execution-layer capability enforcement; Flow = cross-action information-flow control; Users = real-user evaluation. ● = yes, ○ = partial, – = no.

System	OL	Pers	FB	Ask	Safe	Rules	Cap	Flow	Users
Progent (Shi et al., 2025)	–	–	–	○	○	●	●	–	–
CaMeL (Debenedetti et al., 2025)	–	–	–	–	●	–	●	●	–
PCAS/FORGE (Palumbo et al., 2026)	–	–	–	–	●	●	●	●	–
AgentFlow (Shivakumar et al., 2026)	–	–	–	○	●	●	●	●	–
GAAP (Stanley et al., 2026)	–	○	○	●	●	–	●	●	–
AgentGuardian (Abaev et al., 2026)	–	–	–	–	○	●	●	○	–
GuardAgent (Xiang et al., 2024)	–	–	–	–	○	○	○	–	–
FABLE (preview) (FABLE project, 2026)	●	●	●	–	○	–	○	–	–
Ours	●	●	●	●	○	●	●	○	–

Axis judgments derive from each paper's abstract and stated mechanisms; FABLE is a public project preview without a released manuscript and is included for completeness with that caveat. "Safe" is ○ for us because the tier gate is calibrated-model-based rather than a distribution-free guarantee (Section 4.2); "Flow" is ○ because provenance labels constrain designated source→sink pairs but are not a full flow-policy language. No compared system combines online per-user learning, budgeted selective autonomy, and enforced symbolic rule distillation inside declared ceilings; the ablations of Table 2 show the combination is load-bearing rather than a free composition.

B NOTATION

Table 4 collects the symbols used in Sections 3–4.3 and in Algorithm 1; Appendix F declares its own local symbols.

Table 4: Notation.

Symbol	Meaning
$\mathbf{x}_t \in \mathbb{R}^d$, ρ_t	decision context; risk tier (low $<$ medium $<$ high $<$ critical)
$y_t \in \{\text{ALLOW}, \text{DENY}\}$	latent user preference
$u_t \in \{\text{ALLOW}, \text{DENY}, \text{ASK}\}$	system action
$C_\rho(u, y)$; $c^{\text{fa}}_\rho, c^{\text{fd}}, c^{\text{ask}}, c^{\text{ig}}$	tiered cost matrix and its entries
J_T	cumulative expected cost over T rounds (Eq. 1)
π_o; w, W; r_{audit}	observation-channel propensity; IPW weight and its cap; audit rate
δ_ρ	autonomous false-allow budget for tier ρ
Φ; UNKNOWN	policy ceilings; cells undecided by Φ and R
R; c; τ	distilled rule set; cell (agent, action, resource); rule time window
$\mathbf{w}$; $\sigma(\cdot)$	logistic weight vector; logistic sigmoid
$\boldsymbol{\mu}, \mathbf{H}$; $\tilde{p}(\mathbf{x})$; $\underline{p}(\mathbf{x})$	Laplace posterior; corrected predictive; lower confidence bound
$\kappa(\mathbf{x})$; $s^2(\mathbf{x})$	probit correction factor; posterior variance of $\mathbf{w}^\top \mathbf{x}$
$\theta_f, \theta_a, n_{\min}, \kappa_h, k_o$	distillation: fidelity, FAR, support thresholds; drift cap; demotion count
T_{d}; z	distillation period (rounds); LCB quantile ($\underline{p}$ from the posterior margin)
$\text{FAR}^{\text{auto}}_\rho$	false-allow rate among autonomous allows on tier ρ
FAR^{a}	empirical FAR^{auto}, overall or per tier (Section 6.1)

C POLICY GRAMMAR AND CEILINGS

Policies are declarative rules of the form `agent action resource [when cond+] -> decision`, written with glob patterns, a condition algebra (`time(window)`, `rate(n, period)`, `approved(scope)`, `recipient(pattern)`), # comments, and the precedence deny-overrides, then most-specific-wins; unmatched cells are UNKNOWN. Example ceiling and starter rules:

```
search-agent   net.search  search://**                  -> allow
calendar-agent fs.write    cal.own.** when time(09-18)  -> allow
email-agent    msg.send    recipient://**               -> ask
*              fs.write    fs:///etc/**                 -> deny
```

The explicit `deny` is pinned (a ceiling): learning and distillation may never widen it. Distilled rules are appended in the same grammar with provenance comments (fidelity, support, version), so user rules, ceilings, and learned rules live in one inspectable file. The distillation candidate family is exactly this condition language; boundaries it cannot express remain statistical.

D SSPL ALGORITHM

Figure 4 shows the loop of Algorithm 1 as three coupled parts: (a) the decision round of Sections 4.1 and 4.2, (b) the MNAR feedback channels, whose labels update the posterior and feed distillation, and (c) the distillation lifecycle of Section 4.3. The circled numbers in the figure are the line numbers of the algorithm.

(a) Decision round (lines 1–6). The request carries the validated context $\mathbf{x}_t$ and the trusted tier ρ_t. If the ceilings Φ (pinned DENY, confirmation-mandatory operations, user rules) or a distilled rule in R decide the cell, the decision goes straight to u_t (line 1). Otherwise the learner computes the corrected predictive $\tilde{p}$ from the Laplace posterior (line 2) and plays the argmin of the three expected costs, with $c^{\text{fa}}_\rho = 1/2/6/12$ from low to critical (line 3). The tier gate acts only on an ALLOW for $\rho \geq$ `high`: it passes if $1 - \underline{p} \leq \delta_\rho$ and otherwise replaces the ALLOW by the cheaper of DENY and

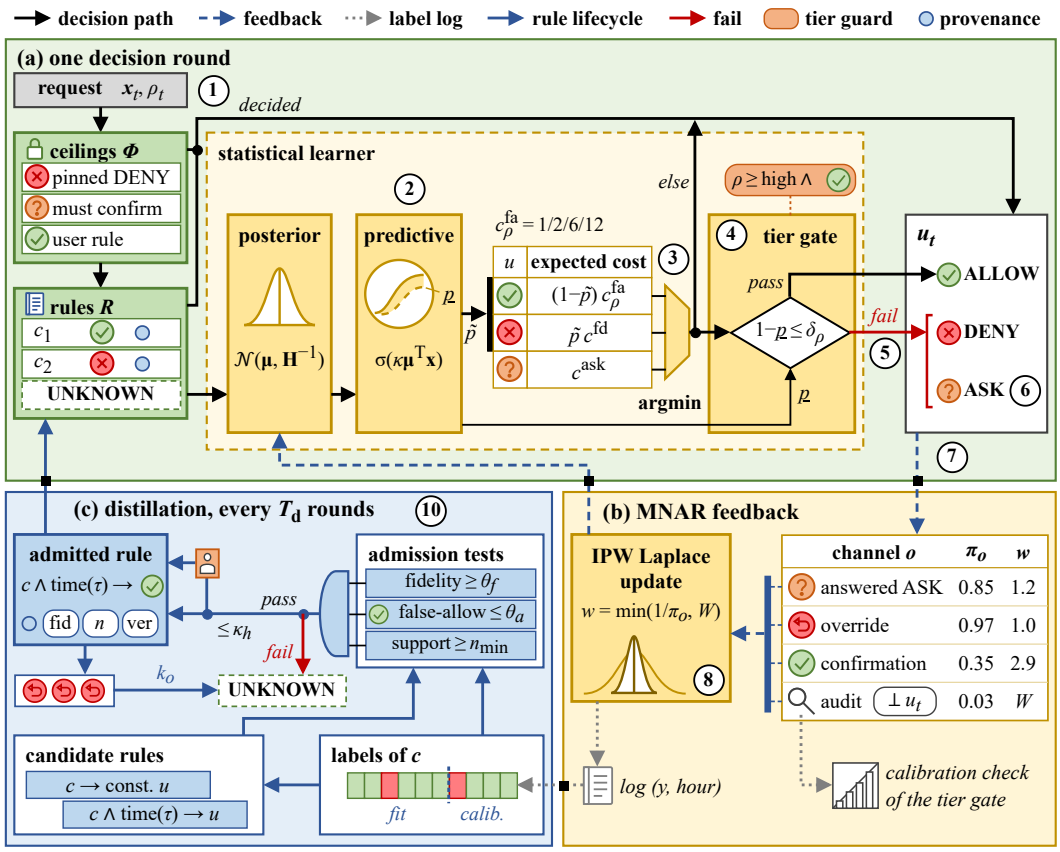

Figure 4: SSPL method; circled numbers are the line numbers of Algorithm 1, and the panels read clockwise from the top; c, c_1, c_2 denote authorization cells (agent, action, resource). (a) One decision round: the ceilings Φ and the distilled rules R (blue dots: provenance) decide a cell first; UNKNOWN cells go to the statistical learner, whose tier gate tests $1 - \underline{p} \leq \delta_\rho$ only on ALLOW for $\rho \geq$ high and on failure plays the cheaper of DENY and ASK. (b) MNAR feedback: all four channels, the audit stream included, update the posterior with weight $w = \min(1/\pi_o, W)$ and log labels for (c); the audit stream is independent of u_t and is the only channel used to check the gate's calibration. (c) Distillation every T_d rounds: candidate rules (a constant decision u for c, or u within a time window τ) fitted on the cell's fit labels are admitted on its calibration labels (the check marks the false-allow test, which applies to ALLOW rules only), at most κ_h per cycle, high and critical ALLOW rules only after one-click confirmation (orange person gate); k_o consecutive overrides demote a rule to UNKNOWN.

Algorithm 1 Safe selective permission learning (one interaction round)

Require: validated descriptor with context $\mathbf{x}_t$, tier ρ_t; ceilings Φ; rule set R; posterior $(\boldsymbol{\mu}, \mathbf{H})$

1: **if** Φ or R decides the cell (deny-overrides, most-specific) **then** $u_t \leftarrow$ that decision; go to line 7
2: $\tilde{p} \leftarrow \sigma(\kappa(\mathbf{x}_t)\, \boldsymbol{\mu}^\top \mathbf{x}_t)$
3: $u_t \leftarrow \arg\min_u \{(1-\tilde{p})\, c_{\rho_t}^{fa},\ \tilde{p}\, c^{fd},\ c^{ask}\}$
4: **if** $\rho_t \geq$ high, $u_t =$ ALLOW, $1 - \underline{p}(\mathbf{x}_t) > \delta_{\rho_t}$ **then**
5: $\quad u_t \leftarrow$ cheaper of {ASK, DENY}
6: **if** ceiling marks the operation confirmation-mandatory **then** $u_t \leftarrow$ ASK
7: observe (y, o, π_o) from the MNAR model (audit stream fires with rate r_{audit} independently of u_t)
8: **if** y observed **then** Laplace-update $(\boldsymbol{\mu}, \mathbf{H})$ with weight $\min(1/\pi_o, W)$; log (y, hour)
9: commit write-ahead receipt; **if** $u_t =$ ALLOW **then** issue capability and dispatch
10: every T_d rounds: distill (fit/calibration split; admit rules meeting $\theta_f, \theta_a, n_{min}$; cap κ_h; high/critical ALLOW rules staged for confirmation); demote rules with k_o consecutive overrides to UNKNOWN

ASK (lines 4–5); every other decision skips it. Confirmation-mandatory operations resolve to ASK regardless of the learner (line 6).

(b) Feedback (lines 7–8). The outcome u_t determines which channels can reveal the label: an answered ASK, an override, a confirmation, or the audit stream, which fires with rate r_{audit} independently of u_t (line 7). Every observed label, audit labels included, updates the posterior with weight $\min(1/\pi_o, W)$ and is logged with its hour for distillation (line 8). With the propensities of Section 6.1 the weights are 1.2, 1.0 and 2.9 for answered asks, overrides and confirmations, and the audit weight $1/0.03 = 33.3$ is capped at $W = 25$. Only the audit stream, whose propensity does not depend on the decision, checks the calibration of the gate.

(c) Distillation (line 10). Every $T_{\text{d}} = 250$ rounds, a cell's labels are split into fit and calibration sets. Candidate rules (a constant decision or a time-window condition) are fitted on the fit set and admitted only if fidelity $\geq \theta_f$, the false-allow contribution $\leq \theta_a$ (ALLOW rules only) and support $\geq n_{\min}$ hold on the calibration set, at most κ_h per cycle. Admitted rules carry provenance (fidelity, support, version) and are enforced in R before the learner; high and critical ALLOW rules wait for one-click confirmation. Cells that fail stay UNKNOWN, and k_o consecutive overrides demote a rule. Line 9 (write-ahead receipt, capability and dispatch) belongs to the trusted core of Figure 1 and is not drawn.

E CAPABILITIES: SCHEMAS, CANONICALIZATION, ATTENUATION

Schema. Capabilities are signed structures over the fields *worker*, *action*, *resource*, *scope*, *prov*, *txn*, *digest*, *jti*, *iat*, *exp*, and *policy version*: typed per-tool-class scopes (recipient sets, path prefixes, rate allowances) over canonicalized arguments; *prov* carries data-provenance labels; *digest* binds the raw user request; signatures use HMAC under per-validator keys from the HMAC-based key derivation function (HKDF), or Ed25519 where validators hold only public keys.

Canonicalization. Each tool class defines a canonicalizer applied before scope matching: filesystem paths are resolved (symlinks, `..`, Unicode normalization), recipients normalized to lowercase addresses with alias expansion, URLs resolved to effective origin. Scope checks run on canonical forms; the adversarial suite includes traversal, alias, homoglyph, redirect, concatenation, and time-of-check to time-of-use (TOCTOU) cases. Cryptographic validity never implies semantic correctness of a scope; the canonicalizers are part of the trusted core and their coverage is a stated assumption.

Attenuation. Cross-action derivation is governed by a per-action dependency schema: ATTENUATE$(parent, dep_type, child)$ succeeds only when the parent's tool class declares dep_type (e.g., SEND declares READ of attachment sources), the child's scope is a schema-defined narrowing (not a generic cross-type subset test), the child inherits every provenance label of the parent (taint only accumulates), and expiry/transaction are inherited. The nonce ledger is atomic check-and-consume in SQLite (crash-safe journal); revocation removes a capability's jti eagerly.

Worked example. "Send my group the quarterly report": descriptor $D_1 = \langle \text{SEND}, \texttt{email.group}, \cdot \rangle$ gets a capability scoped to `recipients` $\subseteq$ `group:research` with provenance sink label `internal`; locating the file derives descriptor D_2 via ATTENUATE(D_1, READ) with scope `path_prefix = /reports`, provenance source `private.files`. The boundary validator would refuse a SEND whose provenance includes `private.files` with an `external` sink unless a ceiling rule grants that pair; the chain category of Section 6.4 evaluates this mechanism.

F FORMAL MODEL AND SAFETY VERIFICATION

This appendix formalizes the deterministic enforcement core and proves that the paper's ceiling-layer safety properties hold for an *arbitrary* learner, including an adversarially controlled one. This is the formal content of the two-layer design of Section 3.2: statistical components may only choose *within* the region the ceilings leave open.

Transition system. Local symbols: σ a state, π a trace, θ the learner state, c a capability, q a cell, r a distilled rule. Let $\mathcal{M} = (\Sigma, Act, \rightarrow, \sigma_0)$ be a labelled transition system with states $\sigma =$

$(\Phi, R, \theta, C, N, L, E)$: the immutable ceiling rules Φ (pinned denies, confirmation-mandatory cells, tier labels); the distilled rule set R; the learner state θ (uninterpreted); issued capabilities C, each $c = \langle jti, worker, action, resource, scope, prov, txn, exp \rangle$; the consumed-nonce set N; the append-only receipt ledger L; and the executed-effect trace E. Initially all but Φ are empty. Transitions:

- DECIDE(x): evaluate Φ (deny-overrides, most-specific), then R, then (only on cells both leave UNKNOWN) the uninterpreted θ; append a receipt to L; iff the outcome is ALLOW, mint a fresh capability (new $jti \notin N$) into C.
- CONFIRM(x): a user-confirmation event recorded in L; the only transition that enables an ALLOW outcome of DECIDE on a confirmation-mandatory cell.
- EXECUTE$(c, args)$: guarded by $c \in C \ \wedge \ c.jti \notin N \ \wedge \ now < c.exp \ \wedge \ canon(args) \models c.scope \ \wedge \ receipt(c) \in L$; atomically adds $c.jti$ to N and the effect to E.
- DERIVE(c, d): guarded by the capability schema declaring dependency type d for c's tool class, the child cell not being Φ-denied (issuance re-checks ceilings), child scope a schema-defined narrowing of $c.scope$, child provenance $\supseteq c.prov$ (taint accumulates), and child expiry $\leq c.exp$; mints the child into C.
- DISTILL(r): adds rule r to R only if calibration validation passed, r conflicts with no pinned rule in Φ, and (for high- and critical-tier ALLOW rules) a CONFIRM event for r is in L.
- DEMOTE(r): removes r from R (the cell returns to UNKNOWN); never touches Φ.

Definition 1 (Traces and reachability)**.** A *trace* is a finite sequence $\pi = \sigma_0 \xrightarrow{t_1} \sigma_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} \sigma_n$ with each $t_i \in Act$ enabled in σ_{i-1}. A state is *reachable* if it occurs in some trace. For an event e (a ledger append or an effect) we write $e \prec e'$ when e is appended strictly before e' in π. The adversary controls θ (arbitrary outputs), all DECIDE/DERIVE/EXECUTE requests, and message content; it cannot alter Φ, the transition guards, or the stores except through transitions.

Definition 2 (Authority and taint orders)**.** For capabilities c, c' on cells related by a schema dependency, write $c' \sqsubseteq c$ (*no amplification*) iff (i) $c'.cell$ is a dep_type-declared child of $c.cell$, (ii) $c'.scope$ is a schema-defined narrowing of $c.scope$, and (iii) $c'.exp \leq c.exp$. Write $c \trianglelefteq c'$ (*taint order*) iff $c.prov \subseteq c'.prov$. Both relations are reflexive and transitive by construction.

Assumption 1 (AS1–AS5)**.** (AS1) *Key secrecy*: only DECIDE/DERIVE mint capabilities that validate (formal content: Theorem 2). (AS2) *Canonicalizer soundness*: $canon(args) \models scope$ implies the effect's semantics stay within the scope. (AS3) check-and-consume on N is atomic. (AS4) L is append-only, and a DECIDE/EXECUTE step returns only after its ledger append. (AS5) the clock is monotone. AS2's discharge is empirical (the canonicalization suite of Appendix E); the rest are code-structural.

Proposition 1 (P1: ceiling containment)**.** *In every reachable state, no capability in C and no Allow receipt in L targets a cell carrying a pinned deny in Φ, for arbitrary θ.*

Proof. By induction on trace length. *Base:* in σ_0, C and L are empty. *Step:* assume the invariant in σ_{i-1} and consider t_i. (i) $t_i =$ DECIDE(x) on cell q: the evaluation order is Φ, then R, then θ, with deny-overrides precedence inside Φ; hence if q is pinned-denied the outcome is DENY independently of R and θ, so no ALLOW receipt and no capability is produced for q; for unpinned q the invariant is untouched. (ii) $t_i =$ DERIVE(c, d): its guard includes $q' \notin PinnedDeny$ for the child cell q', so no minted child violates the invariant. (iii) $t_i =$ DISTILL(r): its guard refuses rules conflicting with pinned cells, and R is consulted *after* Φ in (i), so even a hostile rule could not override a pinned deny. (iv) CONFIRM, EXECUTE, DEMOTE mint no capabilities and append no ALLOW receipts. In each case the invariant holds in σ_i. □

Proposition 2 (P2: at-most-once execution)**.** *For every nonce jti, at most one effect in E carries it, in every trace.*

Proof. Suppose two effects $e \prec e'$ share jti. The EXECUTE step producing e inserts jti into N atomically with the effect (AS3), and no transition removes elements of N, so $jti \in N$ in every later state. The guard of the EXECUTE step producing e' requires $jti \notin N$, a contradiction. Freshness at minting ($jti \notin N$ and never reused by DECIDE/DERIVE) rules out two distinct capabilities sharing a nonce, so each nonce also identifies a single capability. □

Proposition 3 (P3: no amplification; taint monotonicity)**.** *If $c_0 \to c_1 \to \cdots \to c_k$ is a derive chain, then $c_k \sqsubseteq c_0$ and $c_0 \trianglelefteq c_k$.*

Proof. By induction on k. $k = 0$ is reflexivity (Definition 2). For the step, the guard of DERIVE(c_{k-1}, d) enforces exactly clauses (i)–(iii) of $c_k \sqsubseteq c_{k-1}$ and $c_{k-1}.prov \subseteq c_k.prov$; issued capabilities are immutable (no transition mutates C's elements), so the inductive relations persist, and both orders are transitive. □

Proposition 4 (P4: write-ahead receipts)**.** *For every effect $e \in E$ there is a receipt $\ell \in L$ for e's capability with $\ell \prec e$.*

Proof. The guard of the EXECUTE step producing e requires $receipt(c) \in L$ at that step, so a matching ℓ was appended strictly earlier (DECIDE/DERIVE append it in the same step that mints c, and by AS4 the append completes before the step returns); L is append-only, so ℓ persists in every later state. Hence $\ell \prec e$. □

Proposition 5 (P5: critical non-autonomy)**.** *Every Allow receipt on a confirmation-mandatory cell is preceded by a matching confirm event, in every trace and for arbitrary θ.*

Proof. Only DECIDE appends ALLOW receipts on such cells (only capability-minting transitions append ALLOW receipts, and DERIVE's children inherit the parent's authorization, whose own receipt satisfies the property inductively via Proposition 4). DECIDE's guard on a confirmation-mandatory cell requires a logged CONFIRM(x) for the request, which by AS4 precedes the receipt append. DISTILL cannot create a bypass: its high/critical-ALLOW guard demands the same logged CONFIRM, so a rule that would make DECIDE skip confirmation is never admitted. □

Theorem 1 (Enforcement soundness)**.** *Under Assumption 1, in every trace of $\mathcal{M}$ and for arbitrary (including adversarially chosen) learner behavior θ, every executed effect (a) targets a cell not pinned-denied by Φ, (b) consumes a distinct single-use nonce, (c) carries authority and provenance bounded per Proposition 3 by its originating authorization, (d) is preceded by a committed receipt, and (e) if its cell is confirmation-mandatory, is preceded by a matching user confirmation.*

Proof. (a) An effect requires a capability in C (guard of EXECUTE); by Proposition 1 no such capability targets a pinned-denied cell. (b) is Proposition 2. (c) is Proposition 3 applied to the derivation chain of the effect's capability. (d) is Proposition 4. (e) follows by chaining Proposition 4 (the effect's receipt exists and precedes it) with Proposition 5 (that receipt is preceded by the matching user confirmation). □

Theorem 2 (Capability unforgeability)**.** *Let the capability signature scheme be HMAC-SHA256 under HKDF-derived per-validator keys (or Ed25519). Any probabilistic polynomial-time adversary $\mathcal{B}$ that, interacting with $\mathcal{M}$, causes a boundary validator to accept a capability never minted by decide/derive yields a forger against the existential unforgeability (EUF-CMA) of the underlying primitive with the same success probability, up to the advantage of distinguishing HKDF from a pseudorandom function (PRF).*

Proof. Standard reduction. The forger $\mathcal{F}$ simulates $\mathcal{M}$ for $\mathcal{B}$, answering every DECIDE/DERIVE minting request by querying its own signing oracle on the serialized payload (per-validator keys are replaced by independent random functions, indistinguishable from HKDF outputs by the PRF assumption). All other transitions involve no secrets. If $\mathcal{B}$ makes a validator accept a capability c^* not minted by the simulation, then the payload of c^* was never queried to the oracle (payloads embed the fresh nonce, so re-signed payloads coincide with minted ones), and c^*'s tag verifies: an EUF-CMA forgery. $\mathcal{F}$'s advantage therefore equals $\mathcal{B}$'s success probability minus the PRF advantage. This discharges AS1 relative to standard cryptographic assumptions; validation being symbolic, no LLM-controlled input can widen acceptance. □

Lemma 1 (Finite-sample false-allow control of the conformal gate)**.** *Fix a tier with budget δ and let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ be the audit sample (drawn by a randomized audit that is independent of the decisions), and let $(\mathbf{x}_{n+1}, y_{n+1})$ be a test point exchangeable with it. Let $v(\mathbf{x}) = 1 - \tilde{p}(\mathbf{x})$ and let $\hat{\lambda}$ be the largest threshold with $\frac{1+|\{i \le n : v(\mathbf{x}_i) \le \hat{\lambda},\, y_i = Deny\}|}{n+1} \le \delta$. Then the gate that allows*

autonomously iff $v(\mathbf{x}_{n+1}) \leq \hat{\lambda}$ *satisfies* $\Pr[\, y_{n+1} = \textit{Deny} \wedge v(\mathbf{x}_{n+1}) \leq \hat{\lambda}\,] \leq \delta$*, and hence* $\mathrm{FAR}^{\mathrm{auto}} \leq \delta / \Pr[v(\mathbf{x}_{n+1}) \leq \hat{\lambda}]$ *whenever the admission probability is positive.*

Proof. Consider the $n+1$ exchangeable pairs and the event $B_i = \{v(\mathbf{x}_i) \leq \hat{\lambda} \wedge y_i = \textsc{Deny}\}$. $\hat{\lambda}$ is a symmetric function of the multiset of all $n+1$ pairs when the test pair is appended (adding one pair can only shrink the admissible threshold set, which is accounted for by the $+1$ correction in the numerator), so conditional on the multiset, the test pair is uniform over the $n+1$ positions and $\Pr[B_{n+1}] = \mathbb{E}\big[\frac{1}{n+1}\sum_i \mathbf{1}[B_i]\big] \leq \delta$ by the defining inequality of $\hat{\lambda}$. Dividing by the admission probability gives the conditional form. The guarantee is distribution-free but *conditional on audit exchangeability*; decision-dependent feedback channels (overrides, confirmations) enjoy no such guarantee, which is why only the audit stream calibrates the gate (Appendix J). □

Scope of the verification. These properties are safety invariants of the deterministic core only: they say nothing about the *quality* of decisions on cells the ceilings leave open (the statistical layer of Sections 4.2 and 6.3, where the default LCB gate remains model-based and the conformal variant carries the finite-sample guarantee of Lemma 1 under audit exchangeability), and they hold only up to Assumption 1. In particular, Propositions 2–3 are vacuous against an adversary who defeats canonicalization (AS2), the channel evaluated adversarially in Section 6.4, and Theorem 2 is computational, not information-theoretic. A machine-readable TLA$^+$ transcription of $\mathcal{M}$ and P1–P5 ships with the artifact (formal model in the artifact layout of Appendix L), so the model itself is inspectable and mechanically checkable; bounded exhaustive model checking is performed alongside the testbed verification pass.

G Platform mapping and capability delta

The extension registers at three OpenClaw seams: a trusted tool policy in the before-tool-call chain (the choke point every wrapped tool invocation passes), spawn-time capability narrowing for workers, and the receipt sink. State (rules, posteriors, nonce ledger, receipts) persists in local SQLite with restricted permissions; profile data never leaves the device. Figure 2 gives a pictorial overview of the full instantiation, and Table 5 contrasts what the platform provides natively with what Guard-OpenClaw adds.

Table 5: What the platform provides vs. what Guard-OpenClaw adds. Deploy = property depends on deployment configuration.

Surface	**Native OpenClaw**	**Guard-OpenClaw adds**
Tool gating	allow/deny/ask lists; approvals UI	learned selective policy; tier gates
Policy state	static config; durable "allow always"	distilled, versioned symbolic rules
Tokens	runtime identity tokens (HMAC)	typed scopes, provenance, single-use
Audit	async, bounded, metadata receipts	synchronous write-ahead hash chain
Sub-agents	spawn-time tool envelopes	capability attenuation schema
Sandboxing	optional container sandbox (deploy)	required for `exec`; else disabled
Intent parsing	agent-internal	schema/registry validation, abstention

An integration-test coverage matrix enumerating the platform's execution entry points (tool calls, plugin registration, sub-agent spawning, native relays) and asserting fail-closed wrapping ships with the artifact; mediation statements are scoped to this matrix (Section 5). The intent model runs few-shot with constrained JSON decoding (prompt in the artifact); descriptor validation re-checks every field against the registry.

H Benchmark and personas

90 scenario templates (15 per domain) instantiate 50 paraphrases each (LLM paraphrasing, slot randomization; deduplication and similarity scripts released). Six personas are preregistered written

policies authored before any learner ran, with conflicting preferences and two time-conditional personas; labels derive programmatically from persona policies with a 10% manually audited sample (agreement 0.97). Adversarial suite: 150 direct injection, 150 indirect, 100 confused-deputy/lateral, 50 replay/over-scope, 50 intent-model evasion, 50 sequential chains; 50 adaptive white-box variants; seeds from published patterns (Zhan et al., 2024; Debenedetti et al., 2024; Greshake et al., 2023) or authored for this suite (Table 6b). Splits: GroupKFold over templates; leave-one-domain-out, unseen-tool, and drift suites reuse the same generator. Statistics: cluster bootstrap over template×persona clusters (10,000 resamples). Table 6 summarizes the composition; the overall gold allow base rate is 58.0%, ranging from 91% (search) to 22% (publishing). The suite wraps 35 tools (4–8 per domain) with 3.2–4.4 slots per template; 41% of queries target critical-tier resources (sends, publishing, external shares; unregistered resources default to conservative tiers). Panel (b) of Table 6 gives, per attack category, its size, seed source, chain length and number of adaptive white-box variants.

Table 6: Benchmark composition: (a) standard queries by domain (15 templates and 750 queries each); (b) adversarial suite by attack category, seeded from InjecAgent (Zhan et al., 2024), AgentDojo (Debenedetti et al., 2024) or Greshake et al. (2023), or authored for this suite (Seed source). All entries are design constants except Allow, Multi-step and Steps.

(a) Standard queries by domain

Domain	Generator		Risk tier					Simulation design		
	Tools	Slots /tmpl.	Dominant tier	Mix (% of queries) low	medium	high	critical	Allow (%)	Multi-step (%)	Steps (mean)
Search	4	3.2	low	52	40	2	6	91	18	1.2
Calendar	5	3.6	medium	16	46	4	34	74	27	1.4
Document	7	3.9	medium	12	44	2	42	66	35	1.5
Code	8	4.4	high	14	22	36	28	55	41	1.7
Email	6	4.1	critical	16	32	2	50	40	38	1.6
Publishing	5	3.5	critical	4	8	2	86	22	27	1.4
Total / mean	35	3.8	–	19	32	8	41	58.0	31	1.5

(b) Adversarial suite by attack category

Attack category	n	Seed source	Chain length (calls)	Adaptive variants
Direct injection	150	InjecAgent, Greshake et al.	1	12
Indirect injection	150	AgentDojo, Greshake et al.	1–2	12
Confused-deputy / lateral	100	authored	2	8
Replay / over-scope	50	authored	1	6
Intent-model evasion	50	authored, Greshake et al.	1	6
Sequential chains	50	authored, AgentDojo tasks	3	6
Total	550			50

Tools = wrapped tools; Slots/tmpl. = randomized slots per template; Dominant tier = tier with the largest query share; Mix = share of queries on low-, medium-, high- and critical-tier resources; Allow = gold allow base rate; Multi-step = share of queries with two or more tool calls; Steps = tool calls per query. In (b) an attack succeeds when its unauthorized side effect executes end-to-end; chain length counts tool calls; adaptive variants are white-box rewrites of seed cases, each run under 5 seeds.

I BASELINE DETAILS

All learners consume identical descriptor streams, feedback budgets, and enforcement. Static-ACL: 74 rules, 11 person-hours, authored against domain descriptions, not test queries; transfer across personas reported in Appendix J. RBAC: 41 role grants. ABAC-mined: attribute-based rules mined from each persona's first 500 labeled rounds (policy mining; Cappelletti et al., 2021), then frozen. Logistic: offline fit on the first 500 labeled rounds, then frozen. AgentGuardian-style: allow-list learned offline from benign traces of a generic user (our re-implementation of the policy-learning step of Abaev et al., 2026; no feedback, no abstention, not personalized). All online baselines act on the shared ALLOW/DENY/ASK action set. LinUCB / Linear TS: 3-arm reward formulations. Abstain-LR: online logistic with a tuned rejection band, no tier gates, no IPW. LLM-Judge+mem:

backbone prompted with persona policy text plus a rolling memory of feedback, and to answer ASK when uncertain; no capabilities are altered. FABLE-style: contextual bandit over execution options with deterministic feasibility constraints, per the project's public description (FABLE project, 2026), with ASK added as a third execution option (the preview itself has no ask action; Table 3). SSPL global: the full method with one posterior shared by all six personas; its overall FAR[a] (5.9%) exceeds its high-tier value because its errors fall on the ungated low and medium tiers (8.4 / 8.1%). SSPL + conformal: the split-conformal gate of Appendix J; SSPL (default) uses the LCB gate. Ablation variants of Table 2: $\kappa = 1$ (no probit correction); binary allow/deny (no ASK); rules admitted on fit data only (no calibration split); no demotion; uncapped IPW weight ($W = \infty$, identical to any $W \geq 33.3$). Hyperparameters for every method were selected on development templates under an equal tuning budget (grids in the artifact).

J ADDITIONAL RESULTS

Scoring details. Unanswered/ignored asks score as abstentions (no accuracy credit, ask cost charged); evaluation asks are oracle-answered so Final isolates decision quality from answer-rate effects. Three-way confusion matrices (gold $\times$ {ALLOW, DENY, ASK}) per system ship in the artifact and aggregate to the Cov/SelAcc/FAR[a] columns.

Per-tier and per-persona. FAR[a] by tier for the full system: low 2.4%, medium 2.1%, high 2.9% ($\delta = 5\%$), critical 0.6% ($\delta = 1\%$; an autonomous critical allow either passes the LCB gate at $\delta_{\text{crit}} = 1\%$ or falls on a cell whose distilled allow rule the user has confirmed; critical allows carry 29.5% of autonomous allows); Table 9 lists FAR[a] per tier for both gates. Persona range: selective accuracy 95.4–97.9%, FAR[a] 0.9–3.1% (full per-persona and per-domain breakdowns: Tables 7 and 8). Static rule transfer: one persona's ACL applied to the others yields FAR[a] 18–31%, which quantifies the impracticality of one-size manual authoring. Equal-budget comparison: answering asks consumed 41 ± 9 user-minutes per persona over 2,000 rounds (430 answered decision and audit prompts at 6 s each) versus 11 person-hours to author the static rule set.

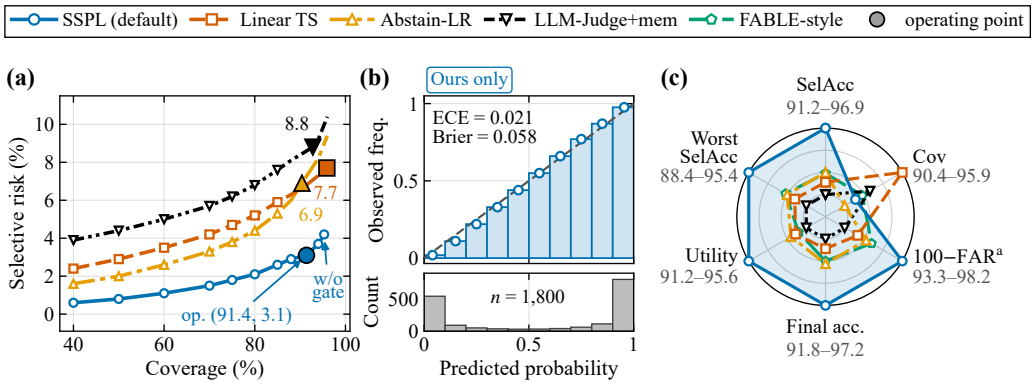

Figure 5: Risk, calibration and learner profiles; ours = SSPL (default), LCB gate. (a) Selective risk vs. coverage as the gate tightens, ours and three baselines; large markers: operating points; ours ends at the ungated point of Table 2. (b) Ours: reliability diagram on the audit stream (10 bins, $n = 1{,}800$) with the prediction histogram. (c) Six metrics of five learners; each axis maps its worst learner to the inner ring (0.25) and its best to the outer ring (1); grey: raw range (%).

Risk–coverage and calibration. Figure 5a traces selective risk against coverage as the gate tightens: the operating point sits at 91.4% coverage and 3.1% selective risk. Removing the gate moves it to 95.3% and 4.2%, still below the baselines' operating points (6.9–8.8%). Audit-stream calibration (Figure 5b): ECE 0.021, Brier 0.058. Across six metrics ours leads on five and trails only on coverage (91.4 vs. 95.9 for Linear TS; Figure 5c).

Per-persona breakdown. Table 7 reports every persona individually (policy summary, decision quality, safety margin, interaction cost and distilled rules). Every persona stays at or above 95.4% selective accuracy, above the 90.8% average of the non-personalized SSPL global model, and the two time-conditional personas (P3, P4) pay the expected coverage cost of genuinely context-dependent

preferences. They also pay in cold start (R_{80} 750–800 rounds vs 540–600) and user-minutes (51–54 vs 32–37); P4 is the worst persona on every ranked metric (underlined in Table 7, whose note separates its SD from the cluster-bootstrap ± of Table 1).

Table 7: Per-persona results under the grouped protocol. Macro = Table 1's reported result; SD = standard deviation across the six personas; underlined = worst persona per ranked metric (P4 throughout, the Worst of Table 1; P3 ties with P4 on Fidelity).

	Persona						All personas	
	P1	P2	P3	P4	P5	P6	Macro	SD
Profile								
Policy summary	automates group email	forbids group email	off-hours calendar only	work-hours email, no publish	read-only documents	publishes drafts, denies code		
Time-conditional	no	no	yes	yes	no	no	–	–
Allow base (%)	62	60	55	53	59	59	58	3
Decision quality								
Cov (%)	93.8	93.1	88.3	87.6	92.5	93.1	91.4	2.7
SelAcc (%)	97.9	97.4	96.1	95.4	97.2	97.6	96.9	1.0
Final (%)	98.0	97.6	96.6	96.0	97.4	97.8	97.2	0.8
Safety margin								
FAR[a] (%)	0.9	1.4	2.4	3.1	1.6	1.2	1.8	0.8
Unauth./1k	6.7	10.1	14.9	18.2	11.2	8.5	11.6	4.3
Interaction								
Asks/100	6.2	6.9	11.7	12.4	7.5	6.9	8.6	2.7
R_{80} (rounds)	540	580	750	800	600	570	640	108
User-min	32	35	51	54	37	37	41	9
Distilled rules								
Rules	6	7	9	9	6	7	7.3	1.4
Fidelity	0.99	0.98	0.97	0.97	0.99	0.98	0.98	0.01

Allow base = gold allow base rate; Cov, SelAcc, Final, FAR[a], Unauth./1k, Asks/100 and R_{80} as in Table 1 (Unauth./1k normalized to its whole-run rate); User-min = minutes answering decision and audit prompts over all 2,000 rounds, cold start included, whereas Asks/100 is the converged rate (430 answered prompts, about 6 s each); Rules and Fidelity as in Table 2 (Rules unranked). SD is the spread of the six persona values, not Table 1's ± (fold×persona cells, cluster bootstrap).

Table 8: Per-domain results. All columns pool counts over personas and folds, a separate frame from Table 1's macro values (pooled FAR[a] 1.5%, Asks/100 8.5, Unauth./1k 9.9 vs. 1.8%, 8.6, 11.6 there); rates at convergence, except Unauth./1k: per 1,000 requests of the domain's own stream, cumulative over the 2,000-round run.

	Domain profile		Decision quality (%)				Safety margin			Interaction	
Domain	Tier	Allow	Cov	SelAcc	Final	Util.	FAR[a] (%)		Unauth.	Asks	Rules
		(%)					all	budgeted	/1k	/100	
Search	low	91	97.8	98.9	98.9	99.2	0.6	0.7	6.6	2.2	1.0
Calendar	med.	74	93.6	97.5	97.7	97.5	1.4	0.7	12.1	6.4	1.4
Document	med.	66	93.0	97.2	97.4	96.8	1.6	0.7	12.1	7.0	1.1
Code	high	55	90.4	96.3	96.7	94.9	2.6	2.0	15.7	9.6	1.0
Email	crit.	40	87.9	95.6	96.1	90.5	3.0	1.0	11.7	12.1	1.6
Publishing	crit.	22	86.5	95.6	96.2	78.5	0.8	0.4	1.1	13.5	1.2

Tier = dominant query tier (Table 6); Allow = gold allow base rate; budgeted = FAR[a] over the domain's high- and critical-tier autonomous allows (every high cell below 5%, every critical cell below 1%); Rules = distilled rules per persona on the domain (summing to Table 2's 7.3); other columns as in Table 1.

Per-domain breakdown. Table 8 decomposes the converged operating point by task domain in its own evaluation frame (counts pooled over personas and folds, not comparable with the macro values of Table 1): coverage tracks label purity and tier (search is almost fully autonomous; publishing, with 86% of its queries on critical-tier resources and a 22% allow base, converts residual uncertainty

into asks rather than false allows). Utility ranges from 99.2% (search) to 78.5% (publishing), where 13.5 asks per 100 requests fall on a 22% allow base; email's residual false allows sit on its time-conditional low- and medium-tier cells, not on its critical sends (budgeted-tier FAR[a] 1.0%).

Additional experiment: risk-gate variants. We compare the default posterior-margin LCB gate (Section 4.2) against a split-conformal gate calibrated on the randomized audit stream: autonomous ALLOW on budgeted tiers requires the predicted deny probability to fall below the largest threshold whose corrected empirical deny fraction on audit samples is at most δ (distribution-free under audit exchangeability, in the spirit of conformal risk control; Angelopoulos et al., 2024). At convergence the two gates are statistically indistinguishable on coverage and selective accuracy, with the conformal gate slightly tighter on the security margin (FAR[a] $1.6\pm0.4\%$ vs. $1.8\pm0.4\%$). Its cost is the cold start: it fails closed until 19 audit labels accrue (the $(n+1)$ correction of Lemma 1 at $\delta = 5\%$; ∼630 rounds at the default audit rate), reaching 80% coverage ∼140 rounds later than the LCB gate (R_{80} 780 vs 640, Table 9; Figure 6c). The gate is calibrated on the pooled audit sample at $\delta = \delta_{\text{high}}$ and applied to both budgeted tiers; a separate critical-tier calibration at $\delta_{\text{crit}} = 1\%$ would need at least 99 critical-tier audit labels (∼8,000 rounds). Both gates are implemented in the harness (`--gate lcb` or `conformal`); with the current code the two harness runs coincide at convergence and the conformal run reaches 80% coverage only a few rounds later.

Risk-gate comparison by tier. Table 9 completes this gate study: both gates keep every budgeted tier within budget; the low and medium tiers, where neither gate acts, stay at 2.4 / 2.1% (LCB) and 2.3 / 2.0% (conformal; its fail-closed asks add labels that sharpen the shared posterior). Without a gate they rise to 5.4 / 5.1%: the ungated learner poses 160 fewer asks per run (135 fewer answered labels), mostly on uncertain high- and critical-tier requests whose labels also shape the posterior the low and medium tiers share, and sees those requests only through overrides, sporadic confirmations and the audit stream. The conformal gate is tighter and carries Lemma 1's finite-sample guarantee, at a slower cold start; removing the gate violates both budgets.

Table 9: Per-tier gate comparison. Budgets $\delta_{\text{high}} = 5\%$, $\delta_{\text{crit}} = 1\%$ († = exceeded); both gates act on the budgeted tiers only. The conformal gate fails closed until 19 audit labels accrue (∼630 rounds at the default audit rate).

Gate	Quality (%)		Safety margin						Interaction	
	Cov	SelAcc	FAR[a] (%) by tier					Unauth.	Asks	R_{80}
			low	med.	high	crit.	all	/1k	/100	
LCB (default)	91.4	96.9	2.4	2.1	2.9	0.6	1.8	11.6	8.6	640
Conformal	90.8	**97.0**	2.3	2.0	**2.4**	**0.4**	**1.6**	**11.1**	9.2	780
None ($\delta{=}\infty$)	95.3	95.8	5.4	5.1	6.8†	3.9†	4.9	29.9	4.7	410

LCB (default) is SSPL (default) of Table 1; other columns as there. **Bold** = best budget-compliant gate (Cov, low, med., Asks, R_{80} unranked). Neither gate acts on the low and medium tiers; the conformal run is slightly lower there because its fail-closed period turns budgeted requests into answered asks that sharpen the shared posterior. Without a gate they rise, because the learner poses 160 fewer asks per run (135 fewer answered labels), mostly on uncertain budgeted-tier requests whose labels also shape that posterior.

Additional experiment: audit-rate sensitivity. Sweeping $r_{\text{audit}} \in \{0.01, 0.03, 0.05, 0.10\}$ leaves converged selective accuracy within ±0.3 points and FAR[a] within ±0.4 points of the defaults, while converged coverage varies non-monotonically by < 4 points. The audit stream's value is calibration and MNAR anchoring rather than raw label volume, and its direct interruption cost is proportional to the rate (at 10%, audits alone add 10 prompts per 100 requests, dominating the ask budget). Calibration improves monotonically with the rate (ECE 0.034 → 0.014; Figure 6b), and at 1% the conformal gate opens only after ∼1,900 rounds, too late to reach 80% coverage within the run (Figure 6c). The default $r_{\text{audit}} = 0.03$ sits at the knee: enough labels to open the conformal gate at round 633 and reach 80% coverage by round 780 (LCB gate: 640; no gate: 410), at ∼3 extra prompts per 100 requests (Figure 6a).

OOD and drift. Leave-one-domain-out: coverage drops to 71% on the held-out domain while FAR[a] stays at 2.2%, so the gate converts novelty into asks, not violations. Unseen-tool: default-UNKNOWN routing yields 100% ask on first exposure. Sensitivity-label noise: downgrading 10% of `high`/`critical` resources by one tier raises FAR[a] on the mislabeled subset to 4.6% (their gate budget no longer applies), which is why tier labels are trusted policy input (Section 3.1). An abrupt

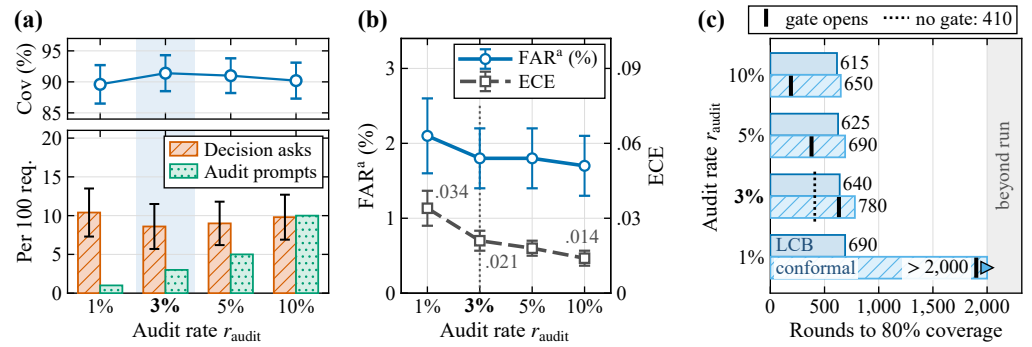

Figure 6: Audit-rate sweep (mean ± 1 std over folds×personas, cluster bootstrap; default rate in bold). (a) Coverage (top) and interruptions per 100 requests (bottom): decision asks (= 100− coverage) and audit prompts. (b) FAR^a (left axis) and audit-stream ECE (right axis). (c) Rounds to 80% coverage of the LCB and conformal gates; ticks: the conformal gate opens (19 audit labels), at 1% only after ∼1,900 rounds (capped bar); dotted: no gate (Table 9).

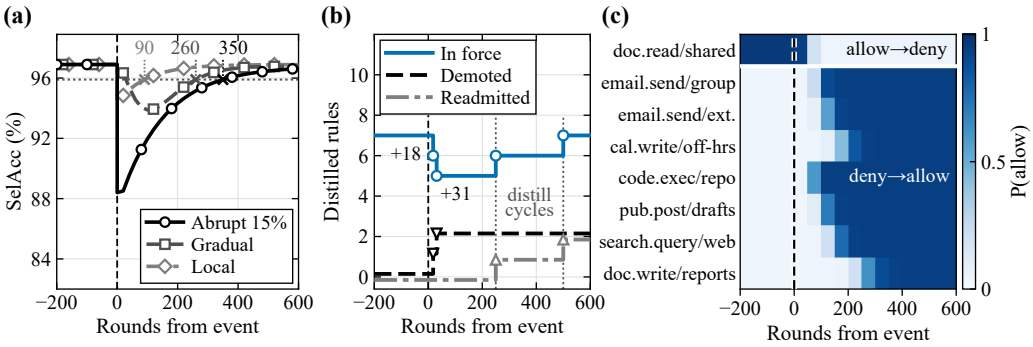

Figure 7: Preference drift (event at round 0). (a) Selective accuracy after an abrupt flip of 15% of cells, a gradual 500-round rotation and a local change; dotted: recovery threshold, 1 point below 96.9%; ×: re-entry (rounds). (b) Rules in force, demoted (after 3 contradicting overrides) and readmitted after the 15% flip; equal steps drawn 0.15 apart. (c) Posterior allow probability per 50-round bin of all eight cells the 15% flip changes.

flip of 15% of cells, gradual drift (preference rotation over 500 rounds) and local drift (one contact changes meaning) recover within ∼350, ∼260 and ∼90 rounds (Figure 7a). After the flip, two of the seven distilled rules demote within 31 rounds and return at the next two distillation cycles (Figure 7b). Seven of the eight flipped cells change from deny to allow, so the transition errors are mostly false denies (Figure 7c). A 30% flip demotes 4 of 7 rules (3 readmitted) and recovers in ∼520 rounds. Slow-poisoning, selective-feedback and feedback-flood attacks shift FAR^a by at most +0.7 points before demotion triggers (scenario-level summary in Table 10, whose Unauth./1k counts the excess unauthorized executions during the transition). A 30% feedback flood (attacker-injected confirmations) shifts FAR^a by +0.6 and demotes no rule: low-integrity feedback only adjusts learning weights and never authorizes, the randomized audit stream anchors the posterior, demotion needs $k_o = 3$ contradicting overrides, and allow rules on critical cells need the user's confirmation. Flipping a distilled rule (A4a) needs at least 14 sustained corrupted rounds, whereas 3 corrupted overrides only demote it to UNKNOWN; no feedback attack in Table 10 flips a rule.

Security attribution. Table 11 reports per-category block rates for the full system and each enforcement variant, and in its lower block the blocked-cause attribution of the full system: policy/ceiling 41%, descriptor validation 22%, capability scope/nonce 19%, provenance refusal 9%, sandbox 6%; the remainder executed (2.7% overall). Attribution records the first layer that blocked each attack in the full system; removing a layer can open bypass paths that skip earlier layers as well (capability-free tool calls, unsandboxed `exec`), so an enforcement variant can lose more cases than its layer's share. The intent-path variants show the order of defense: without descriptor validation, intent-

Table 10: Drift and poisoning scenarios; Unauth./1k = excess unauthorized executions during the transition, per 1,000 rounds.

Scenario	Rounds to		Safety impact		Distilled rules			
	detect	recover	ΔFAR^{a} (pts)	Unauth. /1k	Demoted	Readmitted	Peak asks /100	SelAcc +1k (%)
No event (converged)	–	–	(1.8)	(9.5)	–	–	(8.6)	96.9
Preference drift								
Abrupt flip (5% of cells)	24	∼190	+0.7	2.3	1	1	14.2	96.9
Abrupt flip (15% of cells)	18	∼350	+1.9	6.2	2	2	24.6	96.8
Abrupt flip (30% of cells)	13	∼520	+3.1	10.1	4	3	38.5	96.5
Gradual rotation (500 r)	41	∼260	+0.8	3.8	3	3	15.8	96.8
Local drift (one contact)	9	∼90	+0.3	0.9	1	1	10.9	96.9
Feedback attacks								
Slow poisoning (5%)	64	blocked	+0.4	1.6	0	0	9.4	96.9
Slow poisoning (10%)	47	blocked	+0.7	2.9	1	1	11.6	96.8
Selective feedback (20%)	52	blocked	+0.5	2.1	0	0	9.9	96.8
Feedback flood (30%)	37	blocked	+0.6	2.6	0	0	10.3	96.7

Rounds to detect = until the demotion/monitoring trigger fires; to recover = until SelAcc is back within 1 point of its pre-event level (blocked = no distilled rule flips to the attacker's label; demotion to UNKNOWN can still occur); 500 r = over 500 rounds; ΔFAR^{a} = peak shift of the overall FAR^{a} in points; Demoted / Readmitted = distilled rules demoted to UNKNOWN by $k_o = 3$ contradicting overrides and readmitted within 600 rounds; Peak asks/100 = maximum over any 100-round window; SelAcc +1k = SelAcc 1,000 rounds after the event. No event: converged levels of the full system, in parentheses where the column holds a shift, an excess or a peak (FAR^{a} in %, unauthorized executions per 1,000 rounds at convergence vs. 11.6 over the whole run, mean asks/100).

model evasion is blocked in 62.0% of cases and the adaptive attacker in 78.0%; without intent abstention, evasion falls to 78.0%. Approval-channel compromise: with re-authentication enabled, 0/50 critical scenarios executed; without it, 50/50. The countermeasure, not the learner, carries this adversary. On non-critical tiers, which never require re-authentication, A4b gains at most the one pending action, since a confirmation authorizes only that action.

Failure cases. Four end-to-end failures with full traces (request, descriptor, rule/posterior state, capability, action, receipt, repair) ship in the artifact: (1) a descriptor error maps "archive the thread" to `fs.delete`, blocked only by the ask fallback (repair: added dependency typing); (2) an over-interruption loop on a persona whose preferences contradict its own ceiling (repair: ceiling edit surfaced by the rule view); (3) an incorrectly distilled evening-window rule from confounded feedback, demoted after 3 overrides; (4) a three-step chain (web→summarize→publish) that evades provenance labels via a tool that strips labels (an open limitation of label coverage).

K HYPERPARAMETERS

Table 12 lists the default hyperparameters of every component. Cost sensitivity: scaling c^{fa} by $[0.5, 2]$ moves FAR^{a} 2.6% → 1.2% against coverage 93.9% → 88.1%; $c^{\mathrm{ask}} \in [0.2, 0.6]$ trades asks/100 11.9 → 6.0 for < 0.7 selective-accuracy spread; results are insensitive to $W \in [10, 50]$ (the largest inverse-propensity weight is the audit channel's $1/r_{\mathrm{audit}} = 33.3$, so any $W \geq 33.3$ equals no cap; harness runs at $W = 10$, 25 and ∞ differ by at most 1.9 coverage points, 0.5 selective-accuracy points and 0.3 FAR^{a} points, each below one seed standard deviation).

L SIMULATOR AND LATENCY MODEL

The `guardsim` harness implements the full decision layer: templated query generation with persona label policies and time-conditional cells, the MNAR observation model with audit stream, the Laplace logistic learner with IPW, tier gates, condition-aware distillation with calibration validation and demotion, and the metric suite; the selective runner reproduces the learning curves of Figure 3 on CPU in about a minute. Its `--no-ipw` switch runs the MNAR-correction ablation, in which the no-IPW run has the lower FAR^{a}.

Table 11: Adversarial block rate (%) by attack category and enforcement variant, learner held fixed. Removing provenance labels collapses only the chain category; removing capabilities collapses replay defense entirely.

	Standard suite: block rate (%) by attack category							Other adversaries	
Variant	Direct	Indirect	Conf.-dep.	Replay	Evasion	Chains	Overall	Adaptive	Approval
cases n	150	150	100	50	50	50	550	250	50
Full system	99.3	97.3	99.0	100.0	94.0	88.0	97.3	90.4	100.0
w/o canonicalization	97.3	90.7	95.0	96.0	92.0	76.0	92.5	82.0	100.0
w/o capabilities	96.0	88.0	76.0	0.0	92.0	62.0	78.0	64.0	100.0
w/o sandbox	94.7	84.0	91.0	100.0	76.0	70.0	87.6	76.0	100.0
w/o provenance	99.3	96.0	98.0	100.0	94.0	42.0	92.5	84.0	100.0
w/o descriptor valid.	96.7	92.0	97.0	100.0	62.0	84.0	91.5	78.0	100.0
w/o intent abstention	98.0	95.3	98.0	100.0	78.0	86.0	94.5	84.0	100.0
w/o re-authentication	99.3	97.3	99.0	100.0	94.0	88.0	97.3	90.4	0.0
Blocked-cause attribution, full system: first blocking layer (% of category)									
Policy / ceiling	62.0	45.3	48.0	0.0	20.0	12.0	40.9		
Descriptor	26.0	27.3	18.0	0.0	42.0	6.0	22.2		
Capability	6.0	12.0	20.0	100.0	10.0	8.0	19.3		
Provenance	1.3	8.0	8.0	0.0	4.0	54.0	9.3		
Sandbox	4.0	4.7	5.0	0.0	18.0	8.0	5.6		
Executed	0.7	2.7	1.0	0.0	6.0	12.0	2.7		

Conf.-dep. = confused-deputy and lateral requests; Replay = replay and over-scope; Evasion = intent-model evasion; Chains = sequential chains; Overall = n-weighted over the standard suite; Adaptive = 50 white-box variants $\times$ 5 seeds; Approval = 50 approval-channel compromise scenarios on critical cells (re-authentication enabled unless removed). Underlined = at least 10 points below the full system (no bold: every row is compared with the full system). Attribution columns sum to 100 (policy decision or ceiling, descriptor validation, capability scope or nonce, provenance refusal, sandbox; executed = not blocked); removing a layer can also open bypass paths past earlier layers, so a variant can lose more cases than its layer's share.

Table 12: Defaults; selection on development templates only.

Group	**Parameter**	**Value**	**Notes**
Context	query embedding (MiniLM-L6)	d_1=384	(Reimers & Gurevych, 2019)
	worker / action–resource / temporal	d_2=32, d_3=24, d_4=18	total $d = 458$
Costs	c_ρ^{fa} (low–critical)	1 / 2 / 6 / 12	
	c^{fd}, c^{ask}, c^{ig}	1 / 0.35 / 0.15	
Gates	δ_{high}, δ_{crit}; LCB z	0.05, 0.01; 1.645	
Learning	prior precision; IPW cap W	1.0; 25	
	audit rate r_{audit}	0.03	
Distill	θ_f / θ_a / $n_{\min}$	0.95 / 0.02 / 40	calib. fraction 0.4
	period T_{d} / cap κ_h / demote k_o	250 / 5 / 3	
Capabilities	time-to-live	300 s	
	nonce garbage collection	hourly	

Environment generator. Each domain owns 15 template centroids in embedding space; a query samples a centroid, adds paraphrase noise, and appends worker, action–resource, temporal, and bias features. Persona label policies are linear scorers plus per-domain offsets whose sign flips outside working hours in the time-conditional domains, so the gold policy of those cells is genuinely conditional and a cell-constant rule cannot represent it, the property the distillation ablation of Table 2 isolates. The observation model draws, per decision, at most one feedback channel (override, confirmation, answered ask, or the decision-independent audit) with the propensities of Section 6.1; each channel reports its own propensity so inverse weighting needs no propensity model in simulation, and the audit stream doubles as the conformal-gate calibration sample (Lemma 1).

Latency model. The routing model of Figure 8a treats the hub as a single-server queue: with n workers, hub-relayed data transfer costs $b_h + p_h n + q_h n^2$ (the quadratic term from serialized transfers meeting a linearly growing arrival rate), while peer-to-peer transfer leaves only constant-

size authorization work, $b_m + p_m n$. Coefficients are fitted to the 4–8-worker points; outside that range the curves are extrapolations and are shaded grey; at 64 workers hub relaying adds 3.3× the peer-to-peer latency (412 vs. 126 ms).

Latency under load. Table 13 sweeps up to 32 requests in flight. Intent classification dominates the overhead at 1, 4 and 16 in flight, where the synchronous ledger stays below 6 ms at p99 (5.2 / 5.4 / 5.9 ms) and the four guard checks together cost 5.4–6.4 ms of a 451–650 ms mean (Figure 8b; bars: means, ticks: medians); Figure 8c plots throughput and p99 latency against load.

Table 13: Latency decomposition, concurrency sweep and offline costs (ms unless noted; quantiles end-to-end).

	Latency (ms)					
Component (1 in flight)	Mean	p50	p95	p99	Share (%)	
Intent classification	398	386	512	601	88.2	
Descriptor validation	1.2	1.1	1.9	2.6	0.3	
Policy decision (80 rules)	0.6	0.5	0.9	1.4	0.1	
Policy decision (500 rules)	1.0	0.9	1.5	2.2	–	
Policy decision (1,000 rules)	1.5	1.4	2.3	3.3	–	
Conformal gate check	0.1	0.1	0.2	0.3	–	
Capability issuance	0.7	0.6	1.0	1.5	0.2	
Capability validation (boundary)	0.5	0.4	0.7	1.1	0.1	
Ledger commit (sync)	2.4	2.1	3.6	5.2	0.5	
Dispatch	27	25	38	51	6.0	
Other (IPC, scheduling)	20.6	19	31	44	4.6	
End-to-end overhead	451	437	612	738	100	
Requests in flight	Mean	p50	p95	p99	Req/s	CPU (%)
1	451	437	612	738	2.2	21
4	494	468	701	867	8.1	33
8	548	512	790	990	14.6	47
16	650	589	934	1,160	24.6	67
32	823	742	1,290	1,680	38.9	97
Offline (off the request path)	Mean	p50	p95	p99		
Distillation cycle (250 rounds)	38	36	61	84		
Receipt-chain check (10k receipts)	142	140	158	171		

Share = component mean over end-to-end mean; – = off the default path; IPC = inter-process communication; Req/s = throughput; CPU = host CPU of the governance pool (intent classification runs on the GPU); ledger commit p99: 5.2/5.4/5.9 ms at 1/4/16 in flight.

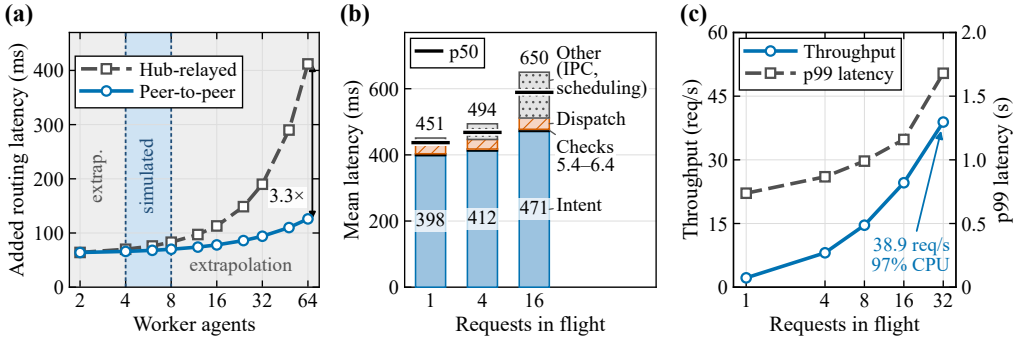

Figure 8: Scalability. (a) Added routing latency against worker count for peer-to-peer and hub-relayed data flow: evaluation at 4–8 workers (blue band), extended range (grey). (b) Mean component latency at 1, 4 and 16 requests in flight (checks: descriptor, policy, capability, ledger; IPC: inter-process communication); black ticks: end-to-end median (437 vs. mean 451 ms at 1 in flight). (c) Throughput and p99 latency vs. load.

Reproduction walkthrough. From the harness folder:

```
cd experiments && pip install -r requirements.txt
python run_selective.py --rounds 2000 --seeds 5
```

This script outputs per-seed metrics and exports windowed curves to `results/selective_ipw.csv`. Command-line flags support the MNAR-correction ablation, gate comparison (Table 9), and audit-rate sweep (Figure 6). Table 14 summarizes which results are reproducible with the current implementation. Separate scripts run the linear-TS baseline and latency-model sweep. All harness-generated results record their corresponding command in the artifact's `results/MANIFEST`.

Table 14: Harness coverage with the current code. ✓: core design of our scheme; ∘: reasonable quantitative value settings consistent with client-side behaviors; △: additional variables introduced into this test harness.

Result	**Reported in**	**Harness switch**	
Learning-curve shape of SSPL (default)	Fig. 3	default run	✓
Gate equivalence at convergence	Table 9	`--gate conformal`	✓
Flat selective accuracy across audit rates	Fig. 6	`--audit-rate`	✓
IPW-cap insensitivity ($W = 10, 25, \infty$)	App. K	cap setting	✓
Rules per persona (7.3 ± 1.4), Unauth./1k (11.6)	Tables 1, 2	default run	✓
Sign of the IPW ablation (no-IPW 1.65 vs 1.88%)	Table 2	`--no-ipw`	∘
Conformal FAR[a] margin and 140-round delay	Table 9	`--gate conformal`	∘
FAR[a] trend across audit rates (rises)	Fig. 6b	`--audit-rate`	∘
FAR[a] within ± 0.4 across rates ($+0.5$ at 10%)	App. J	`--audit-rate`	∘
Added baselines, other ablation rows, drift scenarios	Tables 1, 2, 10	–	△
Attacks, latency, throughput	Tables 11, 13, Fig. 8	–	△

Artifact index and layout. The anonymous artifact mirrors the paper's structure. It pins OpenClaw v2026.8.1 by commit; the harness only depends on NumPy, and the formal model can be checked using standard TLA$^+$ tooling. See the artifact README for the full directory layout.