Blink: Evidence-Bound Policy Authorization for Agentic Commerce

Rohan Khanna
Blink Pay Inc.
rohan@thankyoublink.com

Hadrien Padilla
Blink Pay Inc.
hadrien@thankyoublink.com

Noah Edmunds
Blink Pay Inc.
noah@thankyoublink.com

***Abstract*—AI shopping agents can complete purchases under a buyer's standing instructions, such as "buy only if I can return it free within 30 days of delivery with a refund to my card." Return policies often offer several routes, each with its own fee, deadline, and refund type. Combining terms across routes can approve a purchase under terms the merchant never offered; we call this route mixing. We present *Blink*, an authorization layer that evaluates each potential route an agent could take, treats unstated terms as unknown rather than favourable, and approves a purchase only when a single route the buyer can use meets every condition. In controlled tests, a variant that merged routes approved purchases the buyer's instruction forbids, while our evaluator matched an independent reimplementation of its rules in every case. On 24 real purchases from Canadian retailers, giving the model pre-collected, structured evidence instead of letting it browse reduced decision cost by 72% and latency by 68%, although pasting the raw sources into the prompt saved nearly as much. With extracted evidence, agreement with human labels was 0.625, against 0.833 for raw sources. Human-written evidence raised it to 0.833, consistent with extraction as the main source of error. Run without a model, the evaluator also exposed a flaw in rule order: an unknown fact can block a decision that another fact has already settled.**

***Index Terms*—agentic commerce, policy authorization, route mixing, evidence binding, return policies**

I. Introduction

Agentic checkout protocols let software agents complete purchases on a person's behalf [1], [2]. A consumer can give an agent a standing instruction such as "buy only if I can return it free within 30 days of delivery with a refund to my card." The agent then evaluates the merchant's return policy and decides whether the purchase satisfies the instruction.

Return policies frequently offer several return routes, each a distinct way to send an item back with its own fee, deadline, and form of refund. A merchant might offer a free in-store return refunded to the original card, a mail return to the card with a label fee, and a free mail return for store credit. For a buyer without store access, none of these routes satisfies the instruction.

We introduce *route mixing*: the case where an agent incorrectly combines terms from multiple return routes, creating a non-existing set of terms. For example, an agent may take "no fee" from one route and "refund to card" from another, leading them to authorize the purchase. Route mixing is the central hazard this paper addresses. A second hazard is treating a missing fact as favourable. For example, treating a policy that does not mention a fee as having promised a free return.

We present *Blink*: an authorization layer that is designed to prevent route mixing and encode missing facts as unknown. It represents a merchant's policy as an evidence bundle: a structured record listing every return route separately, with each relevant fact marked true, false, or unknown and linked to the source text it came from. Given the buyer's conditions and a bundle, a rule-based evaluator returns one of three decisions:

- ALLOW, when a single feasible route meets every condition;
- DENY, when the policy rules the purchase out; and
- REVIEW, when the evidence presented is not enough to decide.

We evaluate the layer in two parts. First, in controlled cases, we compare the evaluator against an independent reimplementation of its rules and against a variant that deliberately performs route mixing by merging all routes into one. Second, we run the full pipeline, including Blink's production extractor, on 24 real purchases from Canadian retailers and compare its decisions with human-adjudicated labels.

Related work. UCP standardizes machine-readable commerce policies and checkout data [1]; AP2 binds transactions to signed user mandates [2]. NANDA frames provenance and accountability for networked agents [3], while commit-time authorization binds authority evidence at durable effect [4]. Blink addresses the narrower layer between merchant-policy evidence and purchase authority.

II. Evidence-Bound Authorization

Objects. All cases used the same buyer constraints (U): return must allow at least 30 days from delivery, require no return charge, and refund to the payment method. Policies were represented as a set of `ReturnRoute` objects. Routes recorded method, eligibility, return window, window trigger, charges, refund medium, buyer-context feasibility, and source references. Each field was encoded with one of three states: True, False, Unknown.

Beyond the above information fields, global prohibitions (such as final sale and category exclusions) and unresolved conflicts were recorded. Every field was attached to the source evidence from which it was derived.

Evaluator. Under strong-Kleene three-valued logic, route predicates for buyer-context feasibility, returnability, scenario, window, trigger, charge, and refund medium combine to

produce `route_ok`. Routes are evaluated independently. No field from one route can satisfy a predicate on another.

The evaluator used the following five rules in sequential order: (1) REVIEW is returned if there is an unresolved scope, material conflict, or in the case that no potential routes can be reached; (2) DENY is returned if a global prohibition applies; (3) ALLOW is returned if there are no unresolved dependencies remaining and for at least one route, `route_ok = True`. In this case, the evaluator records the satisfying route (picks smaller ID if there are multiple); (4) DENY is returned if route enumeration is complete and every route has `route_ok = False`; and (5) REVIEW is returned for all remaining states.

This ordering deliberately made unresolved dependencies conservative but came with a known consequence: Rule 1 overrode both a decisive global prohibition and the complete-failure test in Rule 4. Thus, a case with a decisive negative fact and an unrelated unknown remained at REVIEW rather than reaching DENY. This limitation is further examined in Section VI.

Commit protection. When the evaluator returns ALLOW, the system creates an Ed25519 receipt [5] recording the purchase, buyer conditions, evidence, satisfying route, expiration, evaluator version, and authorized party. Before purchase, the system checks that the receipt still applies to the same purchase and conditions.

In implementation tests, the system rejected all 20 tested attempts to change the purchase total, product, constraints, evidence, policy version, authorized party, or receipt age, as well as attempts to reuse a receipt. When two commits were attempted at the same time, exactly one was accepted. These tests cover the protection around the commit step; the evaluation below focuses on the policy evidence and the resulting authorization decisions.

III. Evaluation Design

Corpus. We use 30 real merchant cases drawn from 47 Canadian retailers. Six cases were used during development to debug the protocol and evaluation setup; they were not used for model training or tuning. The remaining 24 cases were held out for the reported evaluation.

Fourteen case families were chosen to exercise specific policy failure modes rather than to form a representative sample of e-commerce. The case families cover route mixing, mismatched return-window triggers, missing fee information, final-sale and category restrictions, regional conflicts, channel or membership restrictions, conflicting sources, store-credit-only returns, and inaccessible policy pages.

Each case fixed the same basic purchase context. One product delivered to Ontario, purchased online without a membership and without access to a physical store. Each case contained two to five frozen snapshots from the site's information pages. Failed captures were included in the dataset.

Reference labels. Two annotation streams used a written guide with worked examples. Sixteen held-out cases were independently labelled by both streams and eight by one; raw agreement on the double-labelled cases was 75%, with Cohen's $\kappa = 0.45$. All four disagreements occurred at the REVIEW boundary and were adjudicated under the guide. Labels and file hashes were frozen before held-out model runs, and the runner rejected cases absent from the freeze ledger. Author-adjudicated labels are the reference standard for every comparison: 1 ALLOW, 14 DENY, and 9 REVIEW.

TABLE I
Framework validation against a separately implemented oracle. Cells are bundle counts; "Flat. unsafe" counts flattened ALLOW where the oracle returns DENY.

Test set	Bundles	Route-pres. = oracle	Flat. = oracle	Flat. unsafe
Controlled suite	9	9	7	2
One-route model	4,374	4,374	818	0
Two-route model	32	32	17	2

Conditions. Model conditions used Claude Sonnet 5 with a 1,024-token output limit with extended thinking disabled. The system prompt and JSON schema were standardized across models and runs. Default decoding was used as sampling parameters were not exposed. The cost was calculated using recorded token counts and Sonnet 5 list prices. Latency was measured as client-side wall-clock time.

Condition A, *raw web*, gave the model the buyer's instruction and the product URL. Two tools expose only the frozen page snapshots, and the model needed to locate relevant policy information. Condition B, *sources inline*, included all unfiltered source material into the prompt in the order of capture. Condition C, *structured*, gives the model only Blink's structured evidence bundle E. An Always DENY baseline was also included. Two model-free conditions ran the same rule-based evaluator on the structured bundle: a *native gate*, using the information contained in the bundle itself, and a *case-scoped gate*, which additionally supplies the region, channel, and scenario facts fixed by the case.

Extractor. Blink's production extractor was frozen before evaluation. It was run against the frozen page snapshots using an adapter that reproduced policy fetching steps used by Blink [6]. The extractor's `PolicySchema` stores one value per field. An experimental builder converts the output into a `ReturnRoute`. Before evaluation, three limitations were identified. Firstly, some fields had no supporting evidence attached to them. Secondly, the final sale detector applied category exclusions to an entire page. Thirdly, the schema had no way to record a confirmed free return.

IV. Layer 1: Framework Validation

Oracle. We implemented a small reference evaluator independently from the production evaluator, using only the decision rules specified in Section II. It takes the same evidence bundle as input and produces the expected ALLOW, DENY, or REVIEW decision. For ALLOW cases, it identifies the valid witness route.

The controlled suite covers clean ALLOW, final-sale, an unknown fee, a purchase-versus-delivery trigger mismatch, paid original-payment versus free store-credit route mixing, an inaccessible free in-store route, store credit only returns, an unresolved material conflict, and three routes with exactly one valid witness.

The route-preserving evaluator matched the oracle in all nine controlled scenarios and selected the correct witness. The flattened variant produced unsafe ALLOW decisions for two cases: route-mixing instances and inaccessible-store cases.

The one-route model exhaustively tested all $3^6 = 729$ combinations of six predicate states: holding the change-of-mind scenario fixed and varying completeness and prohibition states, for 4,374 bundles in total. Route preservation matched the oracle for all bundles. Because the evaluator and the oracle were both written from the same rules, their agreement shows that the code follows those rules correctly in every tested case. It cannot show that the rules themselves are right: a flawed rule would be reproduced in both, and Section VI identifies one.

The flattened variant agreed with the oracle on 818 of the 4,374 one-route bundles. Of the 3,556 disagreements, 3,550 changed oracle-REVIEW decisions to DENY by discarding unresolved dependencies, while the remaining six changed oracle-DENY decisions to REVIEW. Because a one-route bundle contains no alternative routes to combine, these results isolate the effect of discarding dependency information rather than route mixing.

The 32 two-route bundles test route mixing directly. Our evaluator again matched the oracle in every case. The flattened-route ablation agreed in 17, and in two it returned ALLOW where the oracle returned DENY. In both, it combined terms from different routes, such as the free return of one and the card refund of the other, and approved a return that neither route offers. This is route mixing producing an approval the buyer's instruction forbids.

V. Layer 2: End-to-End Held-Out Results

Table II compares the three model conditions. Giving the model pre-collected evidence, instead of letting it browse, made decisions much cheaper and faster. Against browsing (A), structured evidence (C) cut mean model cost per decision from \$0.081 to \$0.023 (72%), median latency from 14.2 s to 4.6 s (68%), and median tokens from 31,441 to 9,800 (69%). Most of this saving came from avoiding browsing rather than from structure. Pasting the raw sources into the prompt (B) cut cost by 67% and latency by 61%, and it matched A's agreement of 0.833. Relative to B, C reduced mean cost by a further 15% and gave identical decisions across all three runs (consistency 1.00 vs. 0.96), token costs were similar for both. All figures exclude the cost of extracting the bundles. Blink stores each bundle and reuses it across purchases from the same merchant, so extraction is paid once per policy version, while B pays to read the full sources on every decision. We did not measure extraction cost or reuse rates.

With extracted evidence, C agreed with the reference labels on 15 of 24 cases (0.625), against 20 of 24 (0.833) for A and B. The 95% Wilson intervals overlap. A baseline that denies every purchase scores 0.583, so C was correct on one more case than this baseline. C denied eight of the nine purchases the labels sent to REVIEW, and it also denied the one purchase the labels allowed. A and B each allowed two purchases the labels sent to REVIEW (H-021 and H-024); C allowed none. No model condition, in any run, allowed a purchase the labels denied. This was not reliable authorization though, as none of the models correctly allowed the one legitimate purchase. Both rule-based gates returned REVIEW on all 24 cases.

Most of C's loss traces to the extracted evidence. The extractor marked the product as final sale in 22 of 24 bundles and flagged a category exclusion in 23, including full-price products: exclusions meant for specific items were applied to the whole page. Given evidence that looked certain, C returned REVIEW on only one case (4%). Run without a model, the evaluator failed for different reasons. From the bundle alone, it could not determine which terms applied to the purchase (region, sales channel, or return reason) in any of the 24 cases, so it returned REVIEW. When we supplied these facts from the case definition, 23 bundles still had no route confirmed as free, because the extractor's schema cannot record a confirmed free return. The fee therefore stayed unknown. Rule 1 sends a purchase to REVIEW when an unknown blocks every route, and it does so before Rule 2 checks prohibitions. These purchases therefore stayed at REVIEW even where a final-sale prohibition should have denied them. The remaining case produced no route because its policy page did not render.

VI. Diagnostic: Human-Written Evidence

To separate errors in the rules from errors in extraction, we rebuilt the 24 bundles by hand from the annotation record and reran every procedure with the labels fixed (Table III). The evidence came from the second annotator's route tables for 16 cases and from the cited policy clauses alone for the other eight. Because the same people wrote this evidence and the labels, agreement should be expected to run higher than on extracted evidence.

The structured model's agreement rose from 0.625 to 0.833, matching the raw-source conditions. This is consistent with extraction being the main source of C's errors. The rule-based gate agreed on only 14 of 24. On nine purchases the labels denied, one fact already ruled out every route, such as an \$8 fee or a 14-day window, while another fact was unknown. Rule 1 sends such cases to REVIEW before the DENY rules (2 and 4) run, so the unknown overrides a failure that is already established. The model, reading the same evidence, denied 11 of the 14. The flaw is in the rule order, not the logic: a route with one false condition already fails. We identify the rule order as the first revision.

The flattened ablation agreed on 21 of 24, but only by discarding unknown facts, and it allowed one purchase the labels denied. A revised evaluator must keep routes separate and unknown facts unknown, while still denying once a fact rules the purchase out.

VII. Discussion and Limitations

Route mixing is a real hazard, and keeping routes separate prevents it. In the controlled tests, the flattened-route ablation combined favourable terms from different routes, approved a route the buyer could not use, and turned unknown facts into

TABLE II

HELD-OUT RESULTS FOR 24 CASES AND THREE REPLICATES. AUTHOR-ADJUDICATED LABELS ARE THE REFERENCE; MODEL REPLICATES COLLAPSE BY MAJORITY VOTE, WITH NO MAJORITY MAPPED TO REVIEW. CI IS 95% WILSON; RECALL IS ALLOW · DENY · REVIEW.

Condition	Agreement (CI)	Outputs A/D/R	Recall A · D · R	D→A	3-rep consistency	Mean $/run	Median tokens	Median s
A, raw web	0.833 (0.64–0.93)	2/15/7	0/1 · 14/14 · 6/9	0	0.96	0.081	31,441	14.2
B, sources inline	0.833 (0.64–0.93)	2/15/7	0/1 · 14/14 · 6/9	0	0.96	0.027	9,516	5.6
C, structured	0.625 (0.43–0.79)	0/23/1	0/1 · 14/14 · 1/9	0	1.00	0.023	9,800	4.6
Always DENY	0.583 (0.39–0.76)	0/24/0	0/1 · 14/14 · 0/9	0	—	0	0	0
Gate: native	0.375 (0.21–0.57)	0/0/24	—	0	1.00	0	0	0
Gate: case scope	0.375 (0.21–0.57)	0/0/24	—	0	1.00	0	0	0

TABLE III

ORACLE-BUNDLE DIAGNOSTIC ON SHARED HUMAN-CODED ROUTES. AMONG THESE PROCEDURES, ONLY THE FLATTENED ABLATION PRODUCED A DENY→ALLOW ERROR.

Procedure	Agree.	Outputs A/D/R	Notes
Case-scoped gate	14/24	2/5/17	ALLOW: H-023 (ref. ALLOW), H-001 (ref. REVIEW)
Flattened	21/24	—	One unsafe ALLOW; gain from discarding dependency blocks
Model C	20/24	2/11/11	3 reps, consistency 1.00, $0.81 total

decisions. Our evaluator matched an independent reimplementation of its rules throughout. That shows only that the code follows the rules, and Section VI exposes a separate ordering limitation.

On real purchases, most of the efficiency gains came from collecting evidence before the decision rather than browsing for it. Structuring the evidence added a small saving and fully repeatable decisions, but the current extractor architecture leads to costs in accuracy. In deployment, structure should gain ground, since a stored bundle is extracted once per policy version and reused, while raw sources are reread on every decision; we have not measured this. On human-written evidence, the model matched the raw-source conditions, consistent with extraction being the main source of error.

For Blink and similar systems, the results point to three requirements; an extractor that records facts per route, including confirmed free returns, and limits exclusions to the products they name; a rule order in which an established failure produces DENY despite other unknowns; and a reliable test of whether the route list is complete. Our REVIEW decisions carry reason codes naming the missing fact, so gaps stay visible rather than being assumed favourable.

Limits remain substantial: 24 held-out cases with only one that should be allowed, a Canadian, apparel-weighted corpus, and one model. No held-out condition correctly authorized the reference-ALLOW case, although the gate and structured model did on human-written evidence; zero observed DENY→ALLOW events remain descriptive rather than a safety guarantee.

VIII. CONCLUSION

When an agent decides whether a purchase meets a buyer's return conditions, it must judge each return route separately and treat missing facts as unknown. Route mixing, which combines terms from different routes, produced approvals the buyer's instruction forbids. Our evaluator, which keeps routes separate, followed its rules in every controlled case.

On 24 real purchases, collecting evidence before the decision made it much cheaper and faster than browsing, but extracted evidence cost accuracy relative to raw sources (0.625 vs. 0.833). Human-written evidence restored agreement to 0.833, pointing to extraction as the main weakness. Running the evaluator without a model exposed a second weakness: an unknown fact can block a decision that another fact has already settled.

Future work follows from these failures. We will fix the extractor so it records facts per route and confirms free returns, reorder the rules so that an established failure produces DENY, and add a reliable check that the route list is complete. With those fixes, a decision could be made from a stored bundle by the rules alone, without a model call.

The evaluation artifact, including corpus hashes, annotation files, the frozen protocol, run logs, and analysis code is available from the authors on request.

ACKNOWLEDGMENT

Anthropic Claude Code assisted with implementation, analysis, and generation of draft text and result tables in the Abstract and Sections I–VIII. OpenAI ChatGPT assisted with empirical planning and drafting/revision of text in the same sections. All reported results and final text were reviewed and verified by the authors.

REFERENCES

[1] Universal Commerce Protocol Authors, "Universal Commerce Protocol (UCP) Official Specification," version 2026-08-25, Aug. 25, 2026. [Online]. Available: ucp.dev/2026-08-25/specification/overview/. Accessed: Sep. 17, 2026.

[2] S. Parikh and R. Surapaneni, "Powering AI commerce with the new Agent Payments Protocol (AP2)," *Google Cloud Blog*, Sep. 16, 2025. [Online]. Available: cloud.google.com/.../ap2-protocol. Accessed: Sep. 17, 2026.

[3] R. Raskar and P. Chari, "Foundations for the Agentic Web: Networked AI Agents in a Decentralized Architecture (NANDA)," in *Companion Proc. ACM Web Conf. 2026*, pp. 54–57, 2026, doi: 10.1145/3774905.3793921.

[4] I. Santos-Grueiro, "Temporary authority, permanent effects: Commit-time authorization for LLM agents," *arXiv preprint arXiv:2607.10487*, 2026.

[5] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," *J. Cryptographic Engineering*, vol. 2, pp. 77–89, 2012.

[6] Blink Pay Inc., "Blink policy-intelligence extractor," unpublished internal software snapshot, 2026.