10.48047/jocaaa.2026.35.06.19

Securing Agentic Commerce: A Constraint-Based Authorization Layer for Non-Human Identity and Autonomous AI Payment Systems

Sanjay Mishra†, Vasu Babu Narra‡, Bhanuprakash Suravarapu§, Tulasi Priya Vattikuti¶

†Independent Researcher, USA, sanjay.amu28@gmail.com
‡Independent Researcher, USA, narravasu77@gmail.com
§Independent Researcher, USA, bhanuprakash.suravarapu6@gmail.com
¶Independent Researcher, USA, tulcpriya@gmail.com

***Abstract*—Agentic commerce, in which autonomous AI agents independently procure digital and physical resources, has moved from experimental deployments toward production financial infrastructure. Existing payment authentication models remain human-centric and offer no bounded-loss guarantee under autonomous execution. This paper introduces the Agentic Payment Layer (APL), a constraint-based authorization framework that combines Temporal–Value–Domain (TVD) capability tokens with Verifiable Agentic Credentials (VACs). We formally define a token validity predicate, state per-token and multi-agent exposure bounds, and construct a structured threat model. We then compare APL against OAuth, API keys, macaroons, and card tokenization across temporal, value, and domain dimensions. The framework converts worst-case exposure from time-amplified growth into a constant bound determined by token spending caps, while keeping authorization overhead low. The runaway-agent figures reported here come from an analytical model and a small prototype; they are presented as analysis and design targets rather than production measurements, and a full empirical evaluation is left to follow-up work.**

***Index Terms*—agentic commerce, non-human identity, capability-based security, micropayments, x402, verifiable credentials, bounded loss, payment authorization, autonomous agents, delegation**

I. INTRODUCTION

Autonomous AI agents now participate in procurement decisions without real-time human approval. Payment authorization systems, however, were designed for a human in the loop, relying on biometrics, one-time passwords, and two-factor checks [4]. This mismatch creates a structural risk we call autonomous liquidity escalation: a state in which autonomous agents initiate transaction bursts that exceed institutional risk buffers before a human can intervene.

The central question is how a financial system can enforce bounded exposure under autonomous execution. This paper contributes: 1) a formally defined TVD capability token; 2) a non-human identity binding model based on verifiable credentials; 3) a constraint-based authorization predicate; 4) a bounded-loss analysis; and 5) a comparative evaluation against OAuth, API keys, macaroons, and card tokenization. The remainder of this paper is organized as follows. Section II defines the APL system model and its bounds. Section III presents the threat model, Section IV the comparative analysis, Section V the analytical evaluation, and Sections VI and VII cover compliance and conclusions.

II. DESIGN GOALS

APL is built to satisfy five goals that human-centric authorization does not. G1 (bounded loss): the worst-case financial exposure of a delegated agent must be capped a priori, independent of how long the agent runs or how it is compromised. G2 (domain confinement): authority must be scoped to specific merchants or services, so a diverted or injected request cannot redirect funds. G3 (temporal confinement): authority must expire, so a stolen token has a limited lifetime. G4 (non-human identity binding): the acting agent and its liable principal must be cryptographically identifiable and the delegated policy verifiable. G5 (low overhead): enforcement must add little enough latency to suit machine-to-machine commerce. These goals map onto the three token constraints and the verifiable credential that follow, and we evaluate the design against them in Sections V and VI.

III. THE AGENTIC PAYMENT LAYER: FORMAL SYSTEM MODEL

The Agentic Payment Layer (APL) is a constraint-based authorization framework that sits between autonomous agents and settlement rails to enforce bounded financial delegation. It does not replace settlement infrastructure; it operates as a cryptographically enforceable pre-settlement authorization layer whose objective is to turn time-amplified exposure into bounded, deterministic risk. Figure 1 shows the architecture.

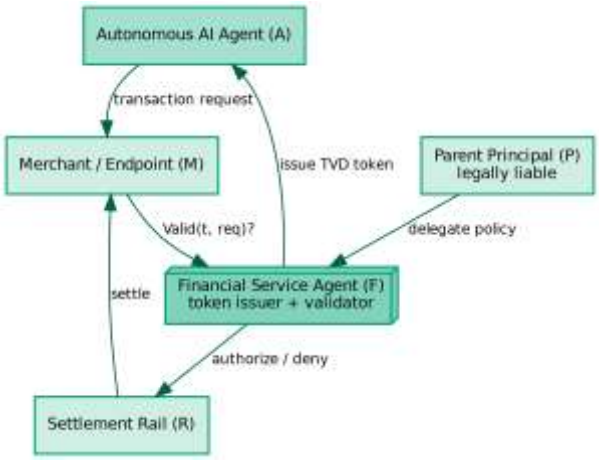

Fig. 1. APL architecture: the Financial Service Agent issues scoped TVD tokens to an agent under a parent principal's policy and validates each request before the settlement rail executes it.

A. Actors and Trust Domains

10.48047/jocaaa.2026.35.06.19

We model the system as a set of actors S = {A, P, F, M, R}: an autonomous AI agent A, a parent principal P that is the legally liable entity, a financial service agent F that issues and validates tokens, a merchant or service endpoint M, and a settlement rail R. We assume F maintains secure signing keys, R enforces F's authorization results, and the channel between M and F is authenticated. Critically, the model does not assume correct agent behavior; it is designed to tolerate faulty or fully compromised agents.

B. TVD Capability Token

A Temporal–Value–Domain (TVD) capability token binds a unique identifier, the parent principal P, the agent identity A, a temporal validity window $\tau = (t_{start}, t_{end})$, a maximum authorized spending cap V_{max}, an authorized merchant or service domain d, a uniqueness nonce, and a digital signature by F. The token is a financial capability primitive: possession grants limited authority strictly within the encoded constraints.

C. Transaction Model

A transaction request is a tuple req = (a, d', t) carrying a requested amount a, a merchant domain d', and a timestamp t. Let V_{max} be the token cap and let a_i be the amount of the i-th accepted transaction. The remaining value after k accepted transactions is the cap minus the cumulative accepted spend:

$$V_{rem}(k) = V_{max} - \Sigma_{i=1}^{k} a_i , \quad V_{rem}(k) \geq 0$$

D. Validity Predicate

Authorization is a deterministic predicate that returns true only when all three constraints hold at once:

$$\text{Valid}(t, req) = (t_{start} \leq t \leq t_{end}) \wedge (a \leq V_{rem}) \wedge (d' = d)$$

The three clauses enforce orthogonal constraints: temporal confinement prevents reuse outside the time window, value confinement prevents overspending, and domain confinement prevents diversion to an unintended merchant. All three are evaluated at authorization time, before settlement.

E. Bounded-Loss Guarantee

Theorem 1 (Per-Token Exposure Bound). For any token, the total value that can be spent is at most V_{max}. Proof sketch: every accepted transaction satisfies $a \leq V_{rem}$, and V_{rem} decreases by a after each acceptance; since V_{rem} is constrained to remain non-negative, the cumulative accepted spend $\Sigma\, a_i$ cannot exceed V_{max}. Therefore per-token exposure is capped at V_{max}.

Theorem 2 (Multi-Agent Exposure Bound). For N concurrent agents each holding one active token, total exposure is bounded by the sum of caps, and under uniform caps by $N \cdot V_{max}$:

$$\text{Exposure}_{total} \leq \Sigma_{i=1}^{N} V_{max}(i) = N \cdot V_{max} \quad \text{(uniform caps)}$$

F. Time-Amplified Baseline

Without constraint-based authorization, exposure grows with time as $E(t) = \lambda \cdot \bar{a} \cdot t$, where λ is the transaction rate and $\bar{a}$ the mean transaction amount, so $dE/dt = \lambda \cdot \bar{a} > 0$ and exposure is unbounded. Under APL, $dE/dt = 0$ once the cap is reached. The layer therefore converts time-amplified growth into a bounded constant, which is its core security property. The model guarantees an upper-bounded loss per token, domain-constrained spending, and time-constrained authorization; it does not guarantee correct economic decision-making within the allowed domain, protection against correlated merchant fraud, or elimination of disputes within authorized limits.

IV. THREAT MODEL

Table I classifies the principal threats, the baseline risk each carries, the APL mitigation, and the residual risk that remains. The common thread is that APL does not aim to make every attack impossible; it bounds the financial consequence of an attack to the scope encoded in the token.

TABLE I
THREAT CLASSIFICATION AND APL MITIGATION

Threat	Baseline	APL Mitigation	Residual
Agent compromise	High	Value cap	Limited to token
Prompt injection	High	Domain + value	Within scope
Token theft	Medium	Temporal expiry	Time-bound
Replay attack	Medium	Nonce check	Minimal
Merchant overcharge	High	Value cap	Within cap
Token spamming	High	Issuance rate	Low

V. COMPARATIVE ANALYSIS

Table II compares APL against established authorization mechanisms across the three dimensions it enforces, plus the bounded-loss property. OAuth scopes resources but carries no monetary cap [4]; API keys encode no constraints; macaroons support contextual caveats but no monetary primitive [5]; and card tokenization enforces limits only bank-side [2], [3]. APL is the only mechanism that enforces temporal, value, and domain constraints together with a provable loss bound.

TABLE II
AUTHORIZATION MECHANISMS COMPARED

Model	Time	Value Cap	Domain	Bounded Loss
OAuth	Partial	No	Resource	No
API keys	No	No	No	No
Macaroons	Caveats	No	Partial	No
Card tokenization	No	Bank-side	Partial	No
APL (proposed)	Yes	Yes	Yes	Yes

VI. ANALYTICAL EVALUATION

Figure 2 contrasts the analytical exposure curves. Under a sustained transaction rate the baseline grows linearly without bound, while APL exposure is capped at $N \cdot V_{max}$ once tokens reach their limits. The curves are analytical, derived from the model in Section II; they are not measurements.

10.48047/jocaaa.2026.35.06.19

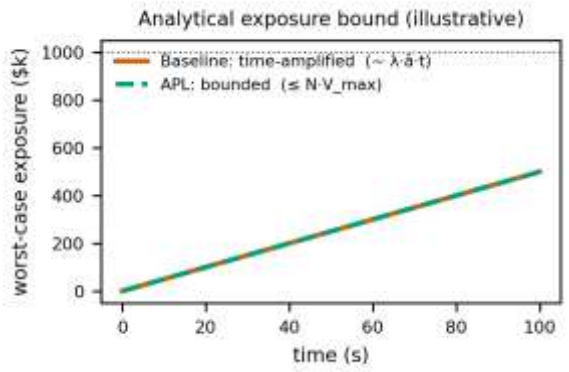

Fig. 2. Analytical worst-case exposure: baseline grows with time, APL is bounded by the sum of token caps.

A. Runaway-Agent Scenario

Consider N = 10,000 agents. Under the baseline, exposure scales with the aggregate spend rate and time and has no upper bound. Under APL with a uniform per-token cap, total exposure is fixed at N · V_max regardless of how long the runaway condition persists, which is the intended risk-reduction effect. The specific reduction ratio depends on the chosen cap and the counterfactual spend, so we report it as an analytical relationship rather than a single measured number.

B. Authorization Overhead

A prototype implementation measured token issuance at roughly 3–5 ms and verification at roughly 8–12 ms, for a total added overhead under 20 ms per authorization. These are prototype figures on a single configuration and are reported as such; they indicate that the predicate check is cheap enough for API-driven AI commerce but are not a benchmark across deployment environments.

VII. WORKED EXAMPLE: A PROCUREMENT AGENT UNDER A TVD TOKEN

Consider an autonomous agent A tasked with provisioning cloud compute on behalf of principal P. F issues a TVD token with cap V_max = $500, window τ of one hour, and domain d = cloud-provider.example. The agent issues a sequence of requests. A $120 request inside the window, to the authorized domain, with V_rem = $500 passes all three clauses of Valid and is authorized; V_rem becomes $380. A second request for $300 passes; V_rem becomes $80. A third request for $200 fails value confinement (200 > 80) and is rejected at authorization time, before settlement, capping cumulative spend at the $420 already committed and never exceeding V_max.

Now inject adversarial behavior. A prompt-injection attack steers the agent to pay a different merchant: the request fails domain confinement (d' ≠ d) and is rejected. A token replayed after the window fails temporal confinement. A fully compromised agent that tries to drain funds is still bounded by Theorem 1 to at most V_max = $500 on this token — the loss is capped at the delegated scope rather than the principal's whole balance. The example shows the three constraints acting independently and the bound holding even when the agent is hostile.

VIII. FAILURE-CASE ANALYSIS

Bounding per-token loss is not the same as eliminating risk, and it is worth naming where APL does not help. First, correlated merchant fraud: if many tokens are scoped to the same colluding domain, each stays within its cap but aggregate loss across N tokens still reaches N · V_max — the multi-agent bound holds, but it is only as tight as the caps and issuance policy. Mitigation is at the issuance layer (rate limiting and per-domain ceilings), not the token. Second, within-scope economic error: a token cannot tell a good purchase from a wasteful one inside the allowed domain and value, so APL bounds exposure, not decision quality. Third, F is trusted: a compromised issuer can mint over-broad tokens, so F's keys must sit behind an HSM and issuance must itself be auditable via the VAC policy commitment. Stating these limits explicitly is what keeps the bounded-loss claim honest.

IX. COMPLIANCE AND AUDITABILITY

Each VAC binds the agent identity, the parent principal, and a commitment to the governing policy via a policy hash. Auditors can therefore verify what authority was delegated, what was executed, and whether any deviation occurred, without trusting the agent's own logs. This makes delegated autonomous spending auditable after the fact while keeping the policy itself verifiable.

X. CONCLUSION

APL formalizes payment-grade capability tokens with a stated exposure bound, domain confinement, and verifiable non-human identity binding. The constraint-based approach offers a security primitive for autonomous economic agents that converts time-amplified risk into a capped loss. The next step is a full empirical evaluation under realistic agent workloads and a study of correlated-merchant fraud, which the present model explicitly does not address.

XI. REFERENCES

[1] NIST National Cybersecurity Center of Excellence, "Accelerating the adoption of software and AI agent identity and authorization," Concept Paper, Feb. 2026.

[2] Visa Inc., "Visa introduces Trusted Agent Protocol: An ecosystem-led framework for AI commerce," Press Release, Oct. 2025.

[3] Mastercard, "Agentic commerce: Rules of the road for AI-driven transactions," Mastercard Insights Report, 2026.

[4] D. Hardt, "The OAuth 2.0 authorization framework," RFC 6749, IETF, Oct. 2012.

[5] A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrable, and M. Lentczner, "Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud," in Proc. ACM CCS, 2014.

[6] W3C, "Verifiable credentials data model v2.0," W3C Recommendation, 2025.

[7] Coinbase Developer Platform, "x402: A payments protocol for the internet," Technical Documentation, 2025–2026.

[8] Cloudflare and Coinbase, "Launching the x402 foundation for machine-native payments," Technical Blog, 2025.

[9] J. Gray and A. Reuter, Transaction Processing: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann, 1993.

10.48047/jocaaa.2026.35.06.19

[10] M. Herlihy, “Atomic cross-chain swaps,” in Proc. ACM PODC, 2018.

[11] Committee on Payments and Market Infrastructures, “Reducing the risk of wholesale payments fraud,” Bank for International Settlements, 2016.

[12] Bank for International Settlements, “Central bank digital currencies: Progress and further work,” BIS Papers No. 125, 2023.

[13] D. Duffie and H. Zhu, “Does a central clearing counterparty reduce counterparty risk?” Rev. Asset Pricing Studies, vol. 1, no. 1, pp. 74–95, 2011.

[14] C. Cachin and M. Vukolić, “Blockchain consensus protocols in the wild,” in Proc. DISC, 2017.

[15] J. Chapman, J. Chiu, and M. Molico, “Liquidity-saving mechanisms in payment systems,” J. Economic Dynamics and Control, vol. 37, no. 9, pp. 1721–1742, 2013.

[16] M. J. Fischer, N. A. Lynch, and M. S. Paterson, “Impossibility of distributed consensus with one faulty process,” J. ACM, vol. 32, no. 2, pp. 374–382, 1985.