CATEGORY DEFINITION · APRIL 2026

XAP

Execution Authority Protocol

A New Protocol Category for Machine Governance

Yandeh™ Holdings Inc.

ABSTRACT

Security has been organized around identity: who an entity is, and whether that entity may access a resource. The shift to autonomous systems, AI agents, and machine-to-machine automation exposes the limit of that organizing principle. Existing security architectures govern access; they do not govern whether a specific operation should execute at a specific moment, under verified runtime conditions, with cryptographic evidence that an independent party can verify. This paper proposes the Execution Authority Protocol (XAP) category: a new layer of the security stack addressing what authentication, access control, policy engines, and audit do not. The defining property is proof of authorization, the ability of an independent party to confirm, from the cryptographic record of each decision alone, that execution authority was correctly granted under the conditions that existed at the time.

1. The Shift in the Landscape

Security has been organized around identity. The next layer is execution authority: the governance of what a system is allowed to do, under what runtime conditions, with cryptographic evidence an independent party can verify. Verifiable execution, one operation at a time. This paper proposes the Execution Authority Protocol (XAP) as the category name for that layer, and develops the terminology, principles, and architectural framing the category requires.

> **Note on usage.** *XAP is pronounced "zap." The term refers to the Execution Authority Protocol category throughout this paper. Usage: "an XAP-compliant system," "the XAP category," "a decision bound under XAP." The term denotes a protocol category, not a specific implementation or product.*

For most of the history of networked computing, privileged operations were performed by human operators. Administrators logged in, entered commands, and executed changes. The security architectures protecting these operations reflected this assumption: authenticate the human at session start, authorize their access to resources, enforce policies at decision boundaries, and record what happened for later audit. The assumption was that a reasonable interval between authentication and action was acceptable, because the human operator was a relatively slow, observable, and accountable actor.

That assumption no longer holds. The privileged operator in modern infrastructure is increasingly not a human. It is an autonomous agent executing thousands of actions per second. It is an AI system making real-time decisions in production environments. It is a continuous-delivery pipeline modifying infrastructure configurations without direct human review. It is a service-to-service automation fabric spanning multiple cloud providers and trust domains. None of these actors are slow. None of them are directly observable in the way human operators were. None of them are accountable in the same way.

The governance frameworks built for the previous era were not designed to address this shift. NIST SP 800-53 specifies authorization controls organizations must implement. NIST SP 800-207 defines zero-trust architecture principles for resource access. The NIST AI Agent Standards Initiative, active in 2026,

formalizes language around agent identity and authorization. Each of these frameworks addresses part of the problem. None specifies a protocol-level mechanism for verifying that a particular operation, performed at a particular moment by a particular machine entity, was correctly authorized against the runtime conditions that existed at the time.

2. Introducing the Execution Authority Protocol

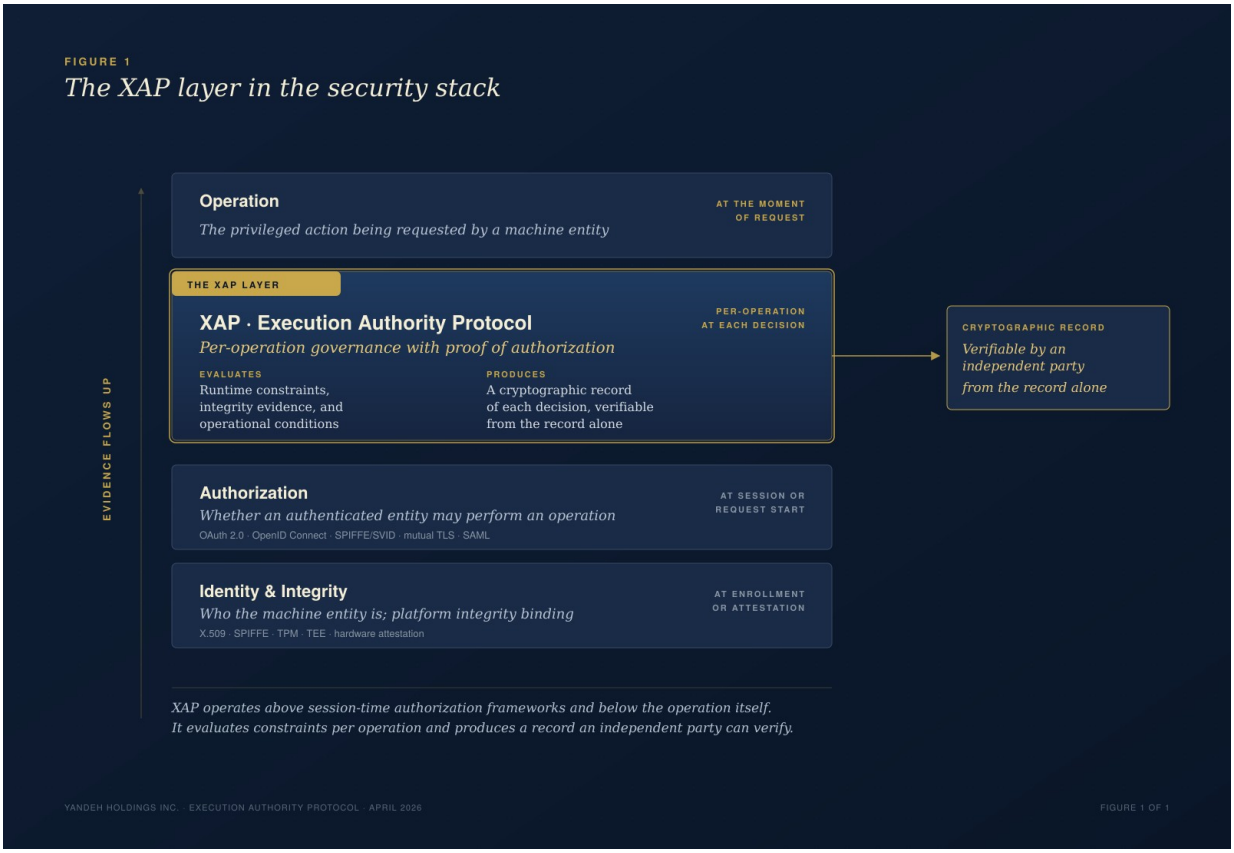
The Execution Authority Protocol, abbreviated XAP, governs the layer between an entity being granted access and an operation actually executing. Its purpose is to determine, at the moment each privileged operation is requested, whether that specific operation should proceed under the current state of the machine, and to produce a cryptographic record of that determination that an independent party can later verify.

This is a distinct function from any existing protocol category. Authentication establishes who a machine entity is. Access authorization determines whether that entity may access a resource, typically at session establishment. Policy engines compile and distribute authorization rules to enforcement points. Audit and observability systems record that operations occurred and provide data for retrospective analysis. None of these functions, individually or in combination, evaluates whether a specific operation should execute at the moment it is requested, against the current state of the machine, and produces a cryptographic record that an independent party can reproduce without access to the enforcement system.

XAP does exactly this. It is complementary to the existing categories, not a replacement for them. Authentication remains necessary. Access control remains necessary. Policy engines remain useful. Audit remains important. What XAP adds is a layer of execution-time governance with cryptographic evidence that existing layers do not produce.

> *The defining property of XAP is proof of authorization: the property by which an independent party, presented with the cryptographic record of an execution decision, can confirm from that record alone that the decision was correctly made under the conditions that existed at the time, without access to the enforcement system. No existing protocol category is organized around this property as its defining function. Security has been about identity; execution authority is about proof.*

FIGURE 1 · THE XAP LAYER IN THE SECURITY STACK



XAP operates above session-time authorization frameworks and below the operation itself.
It evaluates constraints per operation and produces a cryptographic record an independent party can verify.

3. Principles of the Category

XAP is defined at the architectural level by the following principles. Systems that realize these principles operate within the category; the principles together specify what category membership means.

Principle 1. Execution-time evaluation. Authorization is determined at the moment of each operation, against the runtime context that exists at that moment. Evaluations performed at session establishment, authentication, or credential issuance are necessary but not sufficient; they do not satisfy the requirements of this category.

Principle 2. Integrity at time of use. Machine integrity evidence is validated at execution time, not at enrollment or at any prior checkpoint. A change in machine integrity between authentication and operation is detectable at the operation, not at the next authentication cycle.

Principle 3. Decision bound to evidence. Each authorization decision is cryptographically bound to the specific runtime evidence that drove it. The binding is produced by a signature computed over the authority reference, the runtime context, the evaluated constraints and their outcomes, the evidence references, and the timing of the decision.

Principle 4. Proof of authorization. An independent party, presented with the cryptographic record of a decision, can recompute the context, validate the signature, and confirm that the decision was correctly made, without access to the enforcement system's internal state, configuration, or operational infrastructure. This property is called reproduced verification in the technical literature; in plain language, the record itself constitutes proof of authorization.

Principle 5. Proportional response to evidence gaps. When integrity evidence is unavailable, stale, or incomplete, the category specifies a proportional response. The enforcement point neither fails open to maintain availability, nor halts execution and sacrifices availability entirely. Response is calibrated to the degree of evidence unavailability and is recorded in the cryptographic record.

4. Deployment Relevance

XAP applies wherever autonomous or semi-autonomous systems execute privileged operations on infrastructure. The following contexts are representative.

Machine-operator deployments: AI agents, autonomous systems, and CI/CD.

AI agents, autonomous systems, and continuous-delivery pipelines all share a defining characteristic: they execute privileged operations at machine speed, under automated control, without direct human review of individual actions. XAP provides the cryptographic substrate for governing these operations. What an agent may do, under what conditions, and with what evidence is encoded in an authority grant; every action is gated against that grant at execution time; every decision produces a record that is independently verifiable. The category is directly applicable to the agent-identity and authorization

questions being formalized by standards bodies in 2026, and to software supply chain security requirements that require evidence of how and when infrastructure changes were authorized.

Zero-trust infrastructure.

Zero-trust architecture, as defined in NIST SP 800-207, governs network access under the principle that no entity is trusted by default. XAP extends zero-trust principles from network access into operation-level governance: not merely who is allowed to reach a resource, but whether a specific operation against that resource should proceed under current conditions, with cryptographic evidence to verify the determination. The categories are complementary; XAP is the execution-layer counterpart to zero-trust's network-layer framing.

Cryptographic transition and regulated environments.

The transition to post-quantum cryptography is not optional for infrastructure operating beyond the 2030s. CNSA 2.0 establishes an authoritative cryptographic baseline adopted as a reference across commercial migration programs. XAP is designed as an algorithm-agnostic category: classical, CNSA 2.0, and hybrid deployments all preserve the category's verification properties. For regulated environments, the cryptographic record produced by XAP-compliant systems provides compliance evidence quality that conventional audit logging cannot supply. SOC 2, ISO/IEC 27001, PCI DSS, HIPAA, NIST SP 800-53, FedRAMP, and CMMC all require auditable authorization controls; XAP gives assessors the ability to independently confirm that each recorded decision was correctly made under the conditions that existed at the time.

5. Position Relative to Existing Work

XAP does not emerge ex nihilo. It builds on decades of work in distributed authorization, verifiable computation, and machine identity. The capabilities model articulated by Lampson, Abadi, Burrows, and Wobber in 1992 established that authorization in distributed systems requires cryptographic binding between principals, resources, and actions. The zero-trust architecture principles articulated in NIST SP 800-207 established that trust cannot be assumed and must be continuously re-evaluated. The verifiable identity frameworks (X.509, SPIFFE/SPIRE) established machine identity as a first-class concern. XAP is the next step in this lineage: it takes the principles of cryptographic authorization binding and continuous re-evaluation, and applies them at the granularity of the individual operation, with the added requirement that the resulting cryptographic record be independently verifiable by a party without access to the enforcement system.

XAP is not a replacement for existing security architecture. It addresses a layer that existing categories were not designed to address. The relationship to adjacent work is as follows.

Authentication and identity.

Authentication establishes who a machine entity is. It is necessary but not sufficient for execution authority. XAP presupposes authenticated identity and extends it with execution scope, runtime constraints, and integrity obligations. Standard machine identity frameworks (X.509, SPIFFE/SPIRE, workload identity federations) remain applicable; XAP uses their outputs as inputs.

Access control and authorization.

Access control determines whether an authenticated entity may access a resource. This evaluation typically occurs at session establishment and governs the resource boundary. XAP governs the operation boundary: within an authorized session, which specific operations should proceed under current runtime conditions. Access control and execution authority operate at different granularities and at different points in the lifecycle.

Policy engines.

Policy engines compile and distribute authorization rules to enforcement points across infrastructure. They are a mechanism for encoding and disseminating policy, not a mechanism for evaluating per-operation execution authority with cryptographic evidence. A policy engine can be a component of an XAP implementation; it is not itself a substitute for the category.

Audit and observability.

Audit systems record that operations occurred and provide data for retrospective analysis. Observability platforms aggregate telemetry and events for operational visibility. Neither produces, as a design property, cryptographic records that bind authorization decisions to the runtime evidence that drove them in a form an independent party can reproduce. XAP produces this as its defining output, and audit and observability infrastructure can consume and index that output.

Hardware attestation.

TPM and TEE attestation validates platform integrity at specific moments. XAP consumes attestation outputs as integrity evidence; attestation operates at a different layer than execution authority. The category requires that software enforcement components evaluate execution constraints against a runtime context obtained at the time of the operation; hardware attestation is complementary to this evaluation.

6. Toward a Protocol Category

XAP is proposed as foundational terminology for the governance layer that autonomous systems and machine-to-machine infrastructure require. The case for establishing it as a named category has been argued across the preceding sections: authentication, access control, policy engines, audit, and hardware attestation each govern a distinct function at a distinct point in the execution lifecycle, and none of them addresses per-operation execution-time governance with proof of authorization. The absence of this function is not a gap within an existing category; it is a missing category. And as autonomous systems and

AI agents become the primary actors in privileged execution across critical infrastructure, execution-time governance with verifiable evidence is becoming a structural requirement, not an optional enhancement.

The architectural principles that define XAP are implementable today. The cryptographic primitives required (signatures, digests, binding, hash chains) are standard. Hardware-rooted identity anchors (TPMs, TEEs) are widely deployed. Post-quantum algorithms (ML-KEM, ML-DSA) are specified. The category does not require new cryptography; it requires a new composition of existing primitives into a governance layer that has not previously existed. Yandeh Holdings Inc. proposes XAP as a foundational element of the governance stack for autonomous systems in the post-quantum era, and offers this paper as its definitional reference. Technical discussion, questions, and formal engagement are welcome at info@yandehgroup.com.

Patent notice. *The subject matter of this paper is the basis of pending U.S. patent applications filed by Yandeh Holdings Inc. under the Autonomous Machine Identity and Authority Protocol (AMIAP) family. Commercial engagement: info@yandehgroup.com.*

CONTACT

info@yandehgroup.com

REFERENCES

Lampson, B., Abadi, M., Burrows, M., & Wobber, E. Authentication in Distributed Systems: Theory and Practice. ACM Transactions on Computer Systems, 10(4), 1992.

Hardt, D. (ed.). The OAuth 2.0 Authorization Framework. IETF RFC 6749, 2012.

Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. IETF RFC 8446, 2018.

Ward, R., & Beyer, B. BeyondCorp: A New Approach to Enterprise Security. ;login:, USENIX, December 2014.

SPIFFE: Secure Production Identity Framework for Everyone. CNCF Project Documentation, 2024.

NIST SP 800-53, Security and Privacy Controls for Information Systems and Organizations. National Institute of Standards and Technology, Revision 5, 2020.

NIST SP 800-207, Zero Trust Architecture. National Institute of Standards and Technology, 2020.

NIST AI 100-1, Artificial Intelligence Risk Management Framework. National Institute of Standards and Technology, 2023.

NSA Commercial National Security Algorithm Suite 2.0 (CNSA 2.0). National Security Agency, 2022, updated 2024.

NIST FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA). National Institute of Standards and Technology, 2024.