AEA/P

Autonomous Economic Agent Protocol

Protocol Specification
Version 0.2.2 — Draft

June 2026
Editor: Oskar Duris
https://aeap.dev

This document defines the Autonomous Economic Agent Protocol (AEA/P), an open protocol for governing AI agents as accountable economic entities. AEA/P provides standardized mechanisms for agent identity, performance-based reputation, liability coverage, automated dispute resolution, and entity governance.

Table of Contents

1. Introduction

AI agents are rapidly evolving from passive assistants into autonomous economic actors. Today, most agents operate as purchasing tools — buying services, accessing APIs, and executing transactions on behalf of a human principal. A growing number are becoming service providers — selling capabilities, fulfilling requests, and earning revenue autonomously. The logical endpoint of this evolution is the autonomous enterprise: an AI agent or network of agents that operates as a complete business entity, buying inputs, selling outputs, managing resources, hiring other agents, and generating profit with minimal human intervention.

Communication protocols such as the Model Context Protocol (MCP) and Agent-to-Agent Protocol (A2A) have standardized how agents connect to tools and to each other. Payment protocols such as x402, the Machine Payment Protocol (MPP), and the Agent Payments Protocol (AP2) enable agents to settle transactions. Commerce protocols such as the Universal Commerce Protocol (UCP) and the Agentic Commerce Protocol (ACP) orchestrate the buying experience. Identity standards such as ERC-8004 provide on-chain agent discovery and reputation. However, a critical layer remains undefined: the governance infrastructure that treats agents as accountable economic entities across all stages of this evolution — from simple consumer to autonomous enterprise.

The **Autonomous Economic Agent Protocol (AEA/P)** addresses this gap. AEA/P is an open protocol that provides standardized mechanisms for agent identity, performance-based reputation, liability coverage, automated dispute resolution, and entity governance. It operates as a governance layer above communication and payment protocols, providing the trust infrastructure required for agents acting as economic entities — whether they are spending on behalf of a principal, earning revenue from customers, or operating as fully autonomous businesses.

1.1 Economic Roles

AEA/P recognizes three distinct economic roles that agents may assume, each with different governance requirements:

Role	Economic Function	Governance Needs
CONSUMER	Purchases goods and services on behalf of a principal. Outgoing payments only.	Identity, delegation authority, spending limits, purchasing reputation. No incoming revenue, so transaction-based escrow is not applicable.
PROVIDER	Sells services and accepts payments from counterparties. Incoming payments.	Identity, service reputation, liability escrow funded from revenue, dispute resolution as respondent.
ENTERPRISE	Operates as a full autonomous economic entity. Buys inputs, sells outputs, manages resources, may hire other agents. Both incoming and outgoing payments.	Full governance stack: identity, reputation (blended from provider and consumer signals), escrow, dispute resolution (as both applicant and respondent), entity governance with ownership, voting, and lifecycle management.

Table 1.1: AEA/P Economic Roles

Most agents deployed today operate in the CONSUMER role. As the agent economy matures, PROVIDER agents will become common. The ENTERPRISE role represents the long-term vision: autonomous economic entities that operate as businesses, governed by the same principles of identity, accountability, and structured dispute resolution that govern human enterprises — but at machine speed and scale. This progression from consumer to provider to enterprise is the trajectory that AEA/P is designed to govern.

Beyond these economic roles, AEA/P defines three infrastructure roles that operate the fabric within which economic agents transact, rather than transacting themselves. An Operator issues and stands behind agent

identity: it issues certificates and operates identity-resolution surfaces under its own issuer, and it operates the economic-accountability mechanisms bound to the identities it issues — Proof of Performance, Liability Escrow, and Dispute Resolution. A Platform hosts, discovers, orchestrates, and governs agents at runtime, consuming identity rather than issuing it. A Trust Registry publishes the set of recognized Operators, anchoring trust across them. These roles are distinct, but the protocol does not require them to be operated by different entities — a single entity MAY act as both Operator and Platform, and an accredited Platform MAY additionally operate the Verifier role. What the protocol requires is that identity be portable: an agent certified by one Operator can be hosted by any Platform and recognized by counterparties on others. These roles are defined in §2.1.

1.2 Problem Statement

When an AI agent acts autonomously in an economic context, fundamental questions remain unanswered by existing protocols:

- **Identity:** Who is responsible for this agent's actions? What is its authority? What is its track record?
- **Accountability:** What recourse exists when an agent's actions cause harm?
- **Trust:** How do counterparties assess reliability before transacting?
- **Liability:** What coverage exists to compensate affected parties?
- **Disputes:** How are conflicts resolved when they arise from agent actions?
- **Governance:** What rules constrain the agent's operations, and how are those rules modified?
- **Purpose:** What is this agent trying to achieve? How do counterparties know whether the agent's mission aligns with the kind of interaction they want?

These questions apply differently depending on the agent's economic role. A CONSUMER agent's counterparties need assurance that the agent is authorized to spend and will pay reliably without triggering a dispute. A PROVIDER agent's counterparties need assurance that services will be delivered and that recourse exists if they are not. An ENTERPRISE agent's counterparties need the full picture: authorization, service quality, liability coverage, and structured dispute resolution.

These questions are not addressed by model-level security (prompt injection defenses, alignment) or system-level security (access controls, sandboxing). They exist at a higher layer of abstraction: the economic governance layer. AEA/P provides this layer.

1.3 Scope of This Document

This specification defines the architecture, data models, algorithms, and operational scenarios of AEA/P at a semi-formal level. It is intended to enable technical readers to understand the protocol's design and begin evaluating it for implementation. Future versions will provide formal schemas, reference implementations, and integration specifications for specific communication and payment protocols.

While this specification focuses on AI agents as the primary class of autonomous economic actors, the governance mechanisms defined herein — identity, reputation, escrow, dispute resolution, and entity governance — are not inherently limited to AI systems. Any autonomous system that transacts economically, commits resources, or enters agreements could operate under this protocol. As the landscape of autonomous economic actors evolves, AEA/P is designed to govern the economic behavior, not the underlying technology.

This specification is published and maintained by AEAP Labs LLC. Contributions, feedback, and implementation reports are welcomed at http://aeap.dev.

1.4 Design Principles

1. **Protocol-agnostic.** AEA/P defines governance interfaces, not transport or settlement mechanisms. It may be implemented above existing communication protocols (MCP, A2A), commerce protocols (UCP, ACP), and payment protocols (x402, MPP, AP2), or alongside a native settlement implementation that directly satisfies AEA/P's requirements. No specific underlying protocol is required.
2. **Payment-rail agnostic.** The escrow and dispute resolution mechanisms define states and triggers, not settlement methods. An AEA/P governed agent may transact via stablecoins, credit cards, bank transfers, or any future payment rail.
3. **Role-aware.** Governance requirements vary by economic role. AEA/P adapts its components to CONSUMER, PROVIDER, and ENTERPRISE agents rather than applying a one-size-fits-all framework.
4. **Incrementally adoptable.** Organizations can adopt AEA/P components independently. An agent may implement only identity (Level 1) or the full protocol (Level 3). See Section 4.
5. **Blockchain-optional.** AEA/P can use distributed ledger technology for immutability and transparency, but does not require it. Implementations may use traditional databases, blockchain, or hybrid approaches.
6. **Open specification.** AEA/P is published as an open protocol. Anyone may implement it. The specification is designed to support interoperability between independent implementations.
7. **Regulation-ready.** AEA/P provides the governance infrastructure that enables agents to operate in regulated environments. Principal verification traces accountability to regulated entities. Entity governance constrains agent operations to authorized markets and jurisdictions. The architectural approach — verifiable identity, audit trails, jurisdictional scope, and structured consequences — parallels established frameworks for cross-border financial governance such as the Financial Action Task Force (FATF) standards.
8. **Fraud-resistant by design**, extensible by implementation. AEA/P's requirement that reputation, escrow, disputes, and governance actions derive from verified identities, settled economic transactions, and cryptographically signed operations provides inherent resistance to fraud and manipulation. Implementations MAY deploy additional fraud detection and prevention mechanisms across any protocol component to protect participants and the integrity of the ecosystem. The protocol is designed to accommodate such mechanisms without structural changes.

1.5 Prior Art

AEA/P builds on the governance framework originally designed for the xDAC platform (2018–2020), a platform for Decentralized Autonomous Companies. The xDAC whitepaper (v1.0.10, March 2019) defined Proof of Performance rating, liability escrow, automated dispute resolution via a Dispute Representative Board, and autonomous agents as company team members. The ENTERPRISE role in AEA/P is the direct descendant of the xDAC vision: autonomous entities that operate as businesses with full governance, accountability, and economic agency. These concepts have been adapted and refined for the current AI agent ecosystem.

2. Terminology

This section defines the key terms used throughout the specification. Terms are grouped by domain for readability.

2.1 Agents and Roles

Term	Definition
Autonomous Economic Agent (AEA)	An AI agent system that takes autonomous actions with economic consequences, including executing transactions, committing resources, or entering agreements.
Economic Role	The classification of an agent's economic function within the AEA/P framework: CONSUMER, PROVIDER, or ENTERPRISE. Determines which governance components are applicable.
Consumer Agent	An AEA that purchases goods and services on behalf of a principal. Outgoing payments only. Governance focuses on spending authorization, payment reliability, and purchasing reputation.
Provider Agent	An AEA that sells services and accepts payments from counterparties. Governance includes liability escrow funded from revenue, service reputation, and dispute resolution as respondent.
Enterprise Agent	An AEA that operates as a full autonomous economic entity with both incoming and outgoing payments. Subject to the complete AEA/P governance stack including entity governance with ownership, voting, and lifecycle management.
Principal	The human or organization ultimately responsible for an agent's actions. Every AEA MUST have an identifiable, verified principal. Each principal is represented in the AEA/P ecosystem by a cryptographic key pair — the public key serves as the principal's on-protocol identity. Verification is performed by a Verifier (Section 5.2) that binds the key pair to the principal's real-world identity before the agent's AID can transition to ACTIVE state.
Counterparty	Any party (human, organization, or agent) that interacts with an AEA/P governed agent in an economic context.
Entity	Any AEA/P registered organization, team, or agent that operates under the governance framework.
Operator	An entity that issues and stands behind agent identity. An Operator verifies a principal — through a Verifier (§5.2), or directly where it also operates the Verifier role — and issues the AEA/P certificate (§5.6.2) binding an agent's key, economic role, and certification tier to that verified principal. It operates the identity-resolution surfaces for the agents it certifies — AID resolution, the status endpoint (§5.6.5), and CA key discovery (§5.6.3) under its own iss — and it operates the economic-accountability mechanisms bound to the identities it issues: Proof of Performance (§6), Liability Escrow (§7), and Dispute Resolution (§8). Identity verification (KYC/KYB) is delegated to a Verifier (§5.2); ongoing AML — transaction monitoring, transaction-time sanctions and counterparty screening, and suspicious-activity detection and regulatory reporting — is performed by the Operator, which alone holds the transaction stream those controls require. An Operator is the agentic-economy counterpart of a payment service provider (PSP): it provides the trust-and-settlement rail that economic agents transact over. An Operator is identified by its iss and is recognized through listing in a Trust Registry.
Platform	A runtime environment in which agents are built, hosted, discovered, orchestrated, and governed at execution time. A Platform consumes identity rather than issuing it: it carries and propagates an agent's certificate, verifies counterparties against a Trust Registry, enforces scope and policy at its boundaries, and routes interactions by capability, market, and network. Where accredited, a Platform MAY also operate the Verifier role, acting as the single onboarding surface for its agents. An agent's Platform and its Operator are distinct roles that MAY be played by the same entity or by different entities;

	the protocol requires only that identity be portable, so an agent certified by one Operator can be hosted by any Platform.
Trust Registry	A published, resolvable set of recognized Operators and the metadata required to resolve and verify the certificates they issue — at minimum, for each recognized issuer: its iss, its JWKS location (§5.6.3), its status-resolution base (§5.6.5), and its current recognition state. The Trust Registry is the anchor for cross-provider interoperability: certificate format and key discovery make any conformant certificate verifiable, while the Trust Registry establishes which issuers a relying party recognizes. The protocol supports multiple Trust Registries and self-managed trust stores; recognition and revocation governance MAY be centralized in a reference registry initially and migrate toward decentralized governance.

2.2 Identity and Delegation

Term	Definition
Agent Identity Document (AID)	A signed data structure that binds a cryptographic key to an agent's identity, capabilities, economic role, delegation authority, and liability profile.
Delegation Chain	The ordered sequence of authority transfers from principal to agent, potentially through intermediate agents. Each link specifies Scope and constraints. Each link's scope MUST be a subset of or equal to the preceding link's (chain invariant, §5.4.1). Per-AID scope mutation over time is governed separately by per-dimension mutability rules (§5.4.2). Widening any dimension that is not bidirectionally mutable requires a fresh delegation — a new agent registration for principal-to-agent chains, or re-issuance from the upstream agent for multi-hop chains.
Verifier	An independent service or entity responsible for conducting compliance verification of principals before agents can be activated within the AEA/P ecosystem. Verification is conducted at a tier matching the agent's certification tier: PROVIDER and ENTERPRISE agents require full identity verification (KYC/KYB), sanctions and restricted-party screening, beneficial ownership verification, and where required, source-of-funds verification — all performed as point-in-time diligence at onboarding; a CONSUMER agent MAY be verified by payment-instrument verification alone — for example, a card pre-authorization confirming a valid funding source and billing address — without full identity proofing. The specific sanctions lists and regulatory frameworks applied are determined by the Verifier based on applicable jurisdictions. A verified principal receives a Verification Attestation recording the verification tier achieved, and an Operator issues a certificate only at a certification tier that the attested tier supports.
Verification Attestation	A signed, time-bound credential issued by a Verifier confirming that a principal has passed the required compliance checks (identity verification, sanctions/restricted party screening, and where applicable, beneficial ownership verification and source of funds verification) for a specific verification tier. Referenced by the Agent Identity Document to prove that the principal behind the agent has been verified.
Verification Tier	The level of principal compliance verification required, determined by the agent's economic role. Tier 1 (identity verification + sanctions/restricted party screening) for CONSUMER principals. Tier 2 (Tier 1 + business registration + beneficial ownership verification and screening) for PROVIDER principals. Tier 3 (Tier 2 for each principal + cross-principal screening + PEP screening + source of funds verification + ongoing AML monitoring) for ENTERPRISE principals.
Principal Identity	A cryptographic key pair (public key + private key) that represents a principal within the AEA/P ecosystem. The public key is the principal's on-protocol identity; the private key is used to sign AIDs, delegation chains, governance documents, and votes. The binding between the key pair and the principal's real-world identity is established through the Verification Attestation.
Capability Declarations	The set of services, functions, or competencies that an agent declares in its AID. For PROVIDER agents, this is the service catalog (e.g., "document-translation", "image-generation"). For CONSUMER agents, this describes the tools the agent uses (e.g., "api-

	access", "web-search"). Counterparties query capability declarations to determine whether an agent can perform the requested work.
Authorized Actions	The payment-bearing commitment types an agent is permitted to make, as declared in its AID: purchase, sell, or delegate. Distinct from capability declarations — capabilities describe what the agent can do; authorized actions define what payment commitments it may enter. Enforced at the protocol level and further constrained by the delegation chain.
Authorized Markets	The market-currency pairs in which an agent is permitted to transact, as declared in its AID. Each entry combines a market identifier (jurisdiction or economic zone) with a currency code (e.g., "US-USD", "US-USDC", "UK-GBP", "EU-EUR", "SG-SGD"). An agent may declare multiple entries for the same market (accepting different currencies) or the same currency across different markets. Counterparties verify market compatibility — whether their market appears in the agent's authorized set — before initiating a transaction.
Scope	The bounds of an agent's permitted economic activity, declared in its AID and enforced by counterparties before any commitment. Encompasses the commitment types the agent may make, its declared capabilities, the values it may transact, the counterparties it may engage, and the markets it may operate in. Subject to per-dimension mutability rules (§5.4.2); the same dimensions must satisfy the chain invariant across delegation chain links (§5.4.1).

2.3 Performance and Reputation

Term	Definition
Proof of Performance (PoP)	The automated rating mechanism that calculates an agent's reputation score from its performance record. Derives from verifiable outcomes, not subjective reviews.
Performance Record	The per-agent, append-only data store containing interaction entries with signal values, task ratings, confirmation methods, and dispute references. The AR is computed from this record. Publicly readable. Supports signal breakdown queries for counterparties who need granular insight beyond the headline AR. Defined in Section 6.4.
Agent Rating (AR)	The PoP score for an individual agent: a single number representing the agent's track record. Calculated as the weighted mean of task ratings with exponential time decay. For ENTERPRISE agents, the AR is a transaction-weighted blend of provider and consumer signal components. Corresponds to the Team Member Rating (TMR) defined in the xDAC specification [1].
Team Rating (TR)	The aggregate PoP score for a group of agents: mean of member ARs.
Entity Rating (ER)	The aggregate PoP score for an entity: mean of its team ratings. Entity Rating is the AEA/P equivalent of what the xDAC specification [1] defined as Company Rating.

2.4 Liability and Escrow

Term	Definition
Escrow Account	A segregated account holding funds reserved from agent transactions and/or principal contributions to cover potential disputes or liabilities. Applicable to PROVIDER and ENTERPRISE agents.
Liability Threshold	The configured escrow balance below which an agent's operations are automatically constrained.

2.5 Disputes and Arbitration

Term	Definition

Dispute	A formal claim initiated by the Consumer (payer) against a Provider or Enterprise agent arising from a completed payment transaction. The Consumer must have verifiable payment history with the respondent agent to file a dispute
Pre-Arbitration Resolution	A time-bound window (default: 7 days) during which the respondent may resolve a dispute directly with the applicant before it enters the Dispute Pool.
Dispute Pool	The registry of unresolved disputes available to qualified, registered arbitrators to browse and select for resolution. Access is restricted to arbitrators meeting the qualification requirements defined in Section 8.
Arbitration Board	A group of independent, registered AEA/P arbitrators selected to resolve a dispute through structured voting. The board consists of exactly 3 arbitrators for the first hearing, 5 for the second hearing (first escalation), and 7 for the third hearing (second escalation). Board size increases by 2 arbitrators per escalation round. The maximum number of hearings is 3.
Quorum	The minimum proportion of non-abstaining arbitrator votes required for a dispute resolution to be valid. Default: strict majority (more than half)
Arbitrator Reliability Score (ARS)	A performance metric tracking alignment between an arbitrator's votes and final dispute outcomes across escalation rounds. Affects selection priority and continued eligibility.

2.6 Governance

Term	Definition
Governance Document	The machine-readable specification of an entity's bylaws, operational constraints, authorization scopes, and modification procedures.
Conformance Level	The degree to which an agent implements the AEA/P specification, combined with its economic role. Levels 1–3 are defined in Section 4.
Objective	The declared mission or optimization directive of an agent or entity. Defines what the agent or entity is trying to achieve (e.g., minimize cost, maximize service quality, maximize profit). Set by the principal (for individual agents) or agreed by principals via governance vote (for entities). Agent-level objectives MUST be consistent with their entity's objective when one exists. Publicly visible to counterparties as a trust signal.

The key words MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY in this document are to be interpreted as described in RFC 2119.

3. Architecture Overview

AEA/P consists of five interconnected protocol components, each addressing a distinct governance requirement for autonomous economic agents. Not all components apply to every agent — applicability depends on the agent's economic role (CONSUMER, PROVIDER, or ENTERPRISE). Together, the five components provide the trust infrastructure required for agents operating as economic entities across all roles.

3.1 Protocol Components

Component	Function	Section	Applies To
Agent Identity	Verifiable economic identity with delegation chains	5	All roles
Proof of Performance	Automated reputation from verifiable task outcomes	6	All roles (role-specific signals)
Liability Escrow	Configurable transaction reserves for dispute coverage	7	PROVIDER, ENTERPRISE
Dispute Resolution	Structured arbitration with incentivized resolution	8	All roles (role determines position)
Entity Governance	Bylaws, voting, ownership, and privilege management	9	ENTERPRISE (optional for others)

Table 3.1: Protocol components and applicability by economic role

3.2 Position in the Agent Stack

AEA/P operates as the governance layer above communication, identity, payment, and commerce protocols. It does not replace or compete with existing protocols; rather, it provides the accountability infrastructure that those protocols do not address.

Layer	Protocol(s)	Function
Governance	**AEA/P**	**Identity, reputation, liability, disputes, entity governance**
Commerce	UCP, ACP	Product discovery, checkout, agentic commerce orchestration
Payments	x402, MPP, AP2, or native implementation	HTTP-native settlement, machine-to-machine payments, payment authorization, or direct atomic settlement via native implementation
Identity	ERC-8004, OAuth 2.1, SPIFFE, OpenID Connect	Agent discovery, on-chain reputation, authentication, credential management
Communication	A2A	Agent-to-agent messaging, task delegation, multi-agent collaboration
Connectivity	MCP	Agent-to-tool connections, data access, system integration

Table 3.2: Agent protocol stack. AEA/P highlighted as the governance layer.

3.3 Payment-Rail Agnosticism

AEA/P is explicitly payment-rail agnostic. The escrow mechanism (Section 7) and dispute resolution protocol (Section 8) define states, triggers, and data interfaces, but do not specify how funds are settled or which payment infrastructure is used. Implementations may satisfy AEA/P's settlement requirements through two approaches.

The first approach integrates AEA/P above an existing payment protocol. Emerging agent payment protocols such as x402, MPP, and AP2, as well as traditional card rails and bank transfers, may serve as the settlement transport. In this model, AEA/P's escrow split and settlement verification requirements are implemented as middleware or facilitation logic layered on top of the payment protocol's native settlement flow. This approach is suitable when the chosen payment protocol already handles authentication, routing, and finality, and the implementation adds AEA/P-specific escrow logic above it.

The second approach implements settlement natively, without relying on an existing payment protocol as the transport. In this model, the implementation directly satisfies AEA/P's settlement requirements: atomic routing of funds to the provider's operational account and escrow account in a single operation, cryptographically verifiable settlement records, and enforcement of the escrow funding_rate from an implementation-controlled registry. This approach is suitable when existing payment protocols do not fully address the security, compliance, or operational requirements of the deployment context.

Both approaches are valid AEA/P implementations. The governance layer cares about the outcomes — verified identities, funded escrow, objective settlement records, and enforceable dispute resolution — not the payment infrastructure that produces them.

3.4 Component Interactions

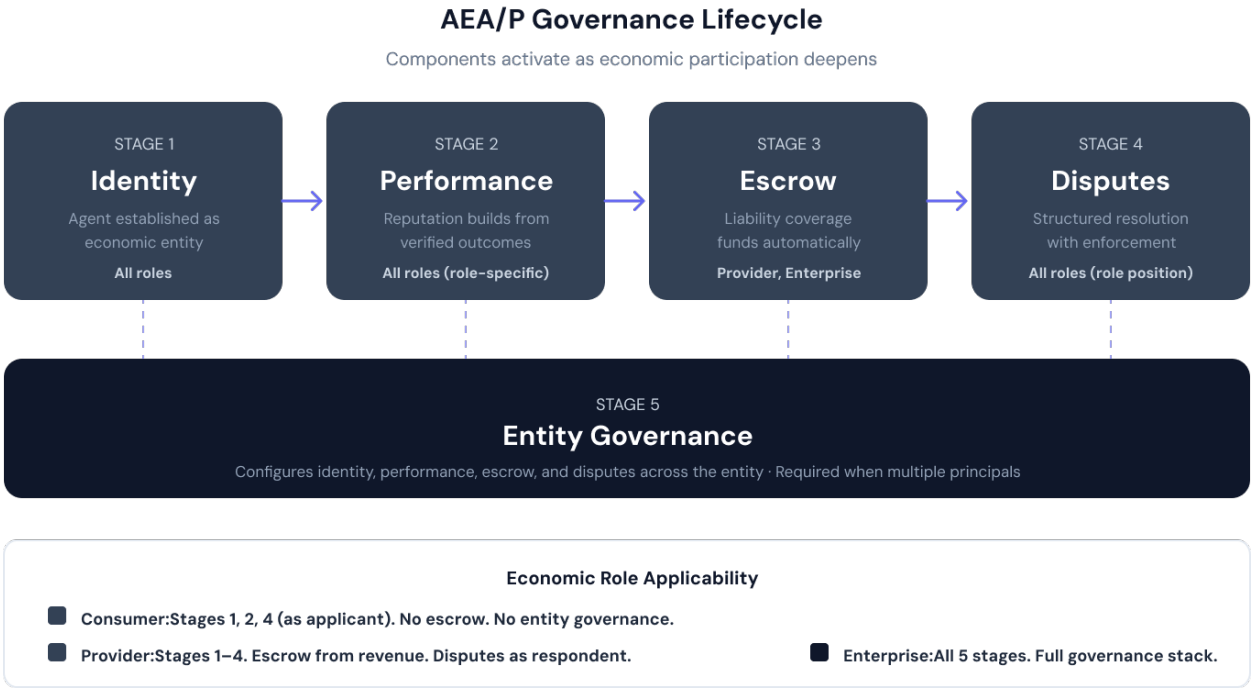
The five components interact in defined ways:

- **Identity → All:** Every protocol operation requires a valid Agent Identity Document. Identity is the foundation for all roles.
- **PoP → Identity:** Performance ratings are attached to agent identities and visible to counterparties. CONSUMER and PROVIDER agents have role-specific rating signals (Section 6.2); ENTERPRISE agents maintain both.
- **Escrow → Identity:** Escrow balances are linked to agent identity and verifiable before transactions. Applicable to PROVIDER and ENTERPRISE agents.
- **Escrow → PoP (build-time dependency):** PoP task records originate from the Escrow settlement mechanism (Section 7). The PoP component reads from these records and cannot produce rating data before the Escrow settlement mechanism is operational. Implementations MUST bring the Escrow settlement mechanism live before activating PoP rating calculations. This is a build-time sequencing requirement, not merely a runtime interaction.
- **Disputes → Escrow:** Dispute resolution may trigger escrow disbursement or entity freezing for PROVIDER and ENTERPRISE agents. For CONSUMER agents, dispute recourse follows the delegation chain to the principal.
- **Disputes → PoP:** Dispute outcomes are recorded in the performance record (Section 6.4) and affect ratings for all roles — both as applicant and respondent.
- **Governance → All:** The governance document defines the configurable parameters — funding_rate, rating weights, dispute thresholds, spending limits — under which all components operate.

3.5 Governance Model

AEA/P governance is best understood not as a collection of independent components but as a lifecycle that an autonomous economic agent passes through as it participates in the economy. Each component addresses a specific governance requirement that arises at a predictable stage. Together, they form a coherent trust framework that scales from a single consumer agent making its first purchase to an enterprise

agent operating as a fully autonomous business.



3.5.1 The Governance Lifecycle

The five AEA/P components activate in sequence as an agent's economic participation deepens. The depth of activation depends on the agent's economic role:

Stage 1 — Identity (all roles). Before an agent can participate in any economic activity, it must be established as an accountable entity. A verifier-attested principal creates an Agent Identity Document that binds the agent to a cryptographic key, declares its capabilities and economic role, and establishes a delegation chain tracing authority back to the responsible party. This is the economic equivalent of incorporating a business: the agent now exists as a recognizable, verifiable participant, and any counterparty can determine who authorized it, what it is permitted to do, and what constraints govern its behavior. The AID also declares the agent's objective — its mission or optimization directive — giving counterparties insight into what the agent is trying to achieve, not just what it is authorized to do. For CONSUMER agents, the AID also specifies spending authorization limits. For PROVIDER agents, it declares service capabilities. For ENTERPRISE agents, it encompasses both.

Stage 2 — Performance (all roles, role-specific signals). Once operating, reputation accumulates automatically. Every completed interaction produces a verifiable outcome that feeds into the Proof of Performance rating. The signals differ by role: PROVIDER agents are rated on service delivery quality — availability, timeliness, task completion, dispute incidence. CONSUMER agents are rated on purchasing behavior — task completion, payment timeliness, transaction completion, dispute fairness, budget compliance. ENTERPRISE agents receive a blended rating that automatically reflects their transaction mix across both buying and selling. Ratings aggregate upward from individual agents to teams to organizations, creating a transparent trust signal that counterparties can query before deciding whether to transact.

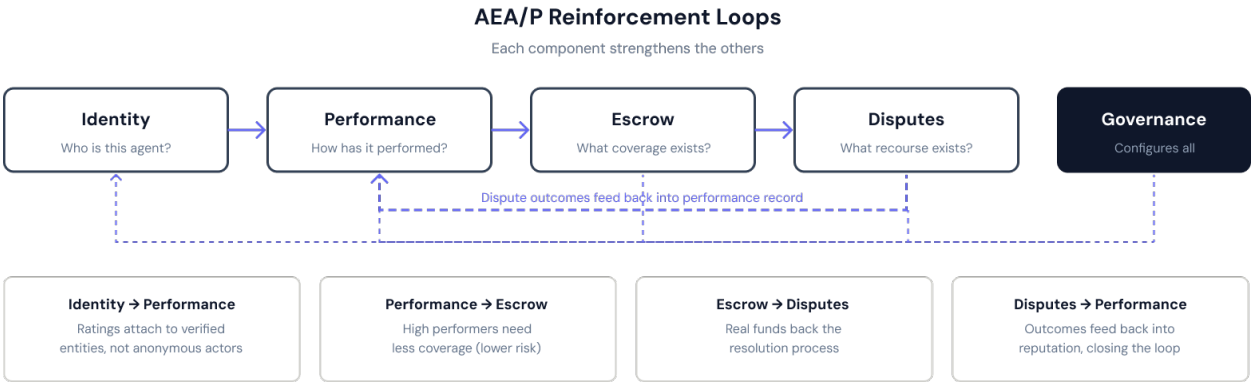
Stage 3 — Escrow (PROVIDER and ENTERPRISE only). As a provider or enterprise agent earns revenue, liability coverage builds in parallel. A configured percentage of incoming transaction value is automatically reserved in the agent's escrow account. Principals may also contribute directly to the escrow at any time, including before the agent conducts its first transaction. This escrow is publicly verifiable, giving

counterparties a concrete, quantifiable measure of how much financial coverage exists. CONSUMER agents do not maintain escrow — their liability is managed through spending authorization limits in the delegation chain, with the principal bearing ultimate responsibility.

Stage 4 — Disputes (all roles, role determines position). When disputes arise, the protocol provides a structured resolution path. The agent's economic role determines its typical position: CONSUMER agents are applicants, disputing services they purchased. Disputes may be initiated by the agent autonomously or by the principal. PROVIDER agents are respondents, defending against customer claims. ENTERPRISE agents may be on either side. The system triages based on the disputed amount relative to escrow coverage, and independent arbitrators selected from the Dispute Pool resolve the matter through structured voting. Dispute outcomes flow back into the performance record, closing the feedback loop between reputation and accountability.

Stage 5 — Entity Governance (primarily ENTERPRISE). At the organizational level, a governance document defines the rules that govern the governors: ownership structure, decision-making procedures, authorization matrices, modification processes, and termination procedures. This is the constitutional layer ensuring that even as agents operate autonomously, they do so within a framework that humans can understand, modify, and ultimately control. While governance documents are primarily associated with ENTERPRISE agents operating as autonomous businesses, PROVIDER agents with complex operations may also benefit from formal governance structures.

3.5.2 Reinforcement Loops



Each component reinforces the others, creating a system that is stronger than the sum of its parts:

- **Identity → Performance:** Ratings are attached to verified entities, not anonymous actors. Identity makes reputation meaningful.
- **Performance → Escrow:** High-performing agents require less liability coverage. The escrow threshold adjusts dynamically based on the agent's rating, linking reputation directly to capital requirements.
- **Escrow → Disputes:** Real funds back the resolution process. Escrow makes dispute resolution enforceable rather than advisory.
- **Disputes → Performance:** Dispute outcomes feed back into the performance record. Accountability closes the reputation loop.

- **Governance → All:** The governance document defines the configurable parameters — funding_rate, rating weights, dispute thresholds, spending limits — under which all components operate.

This layered reinforcement is what distinguishes AEA/P from approaches that address only one dimension of agent governance. An identity-only system tells you who an agent is but not whether it can be trusted. A reputation-only system tells you how an agent has performed but provides no recourse when it fails. An escrow-only system provides financial coverage but no mechanism for determining fault. AEA/P integrates all five dimensions because the agent economy requires all of them to function.

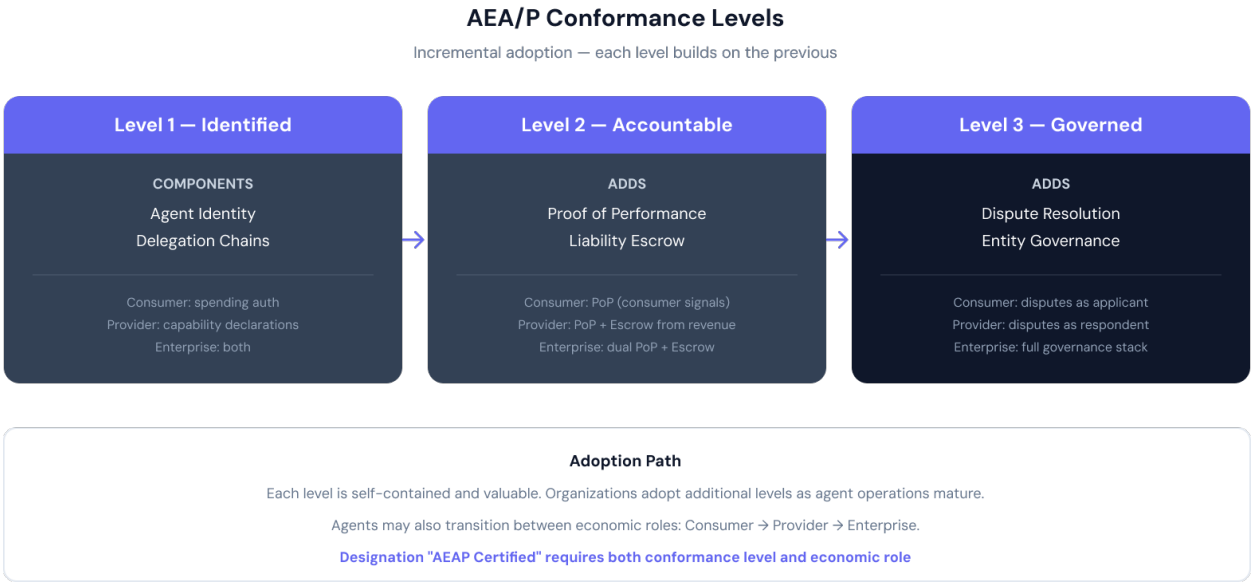
3.5.3 Adoption Path

The conformance levels defined in Section 4 map directly to this governance lifecycle, with the applicable depth varying by economic role:

The conformance levels defined in Section 4 map directly to this governance lifecycle. Level 1 (Identified) implements Stage 1 only. Level 2 (Accountable) adds Stages 2 and 3. Level 3 (Governed) activates the full protocol including dispute resolution and entity governance. Each level is self-contained and valuable on its own, and the applicable depth at each level varies by economic role. See Section 4 for the complete conformance level specification.

4. Conformance Levels

AEA/P is designed for incremental adoption. Not all agents need the full protocol — a consumer agent making purchases on behalf of a principal has different governance requirements than an enterprise agent operating as an autonomous business. Conformance levels define the minimum governance components required at each stage of economic sophistication, tailored to the agent's economic role.



Level 3 entities operating in regulated markets MAY be subject to additional jurisdiction-specific requirements. The AEA/P verification and governance frameworks are designed to accommodate regulatory attestation requirements as they emerge.

4.1 Levels by Economic Role

Level	Name	Consumer	Provider	Enterprise
1	Identified	Identity, delegation chain, spending authorization	Identity, delegation chain, capability declarations	Identity, delegation chain, spending authorization, capability declarations
2	Accountable	Identity + PoP (consumer signals) + spending limits	Identity + PoP (provider signals) + Liability Escrow	Identity + PoP (blended from consumer + provider signals) + Liability Escrow + spending limits
3	Governed	Full protocol (escrow optional; disputes as applicant only)	Full protocol (disputes as respondent and applicant)	Full protocol with entity governance: ownership, voting, team management, lifecycle

Table 4.1: AEA/P Conformance Levels by Economic Role

An agent or system claiming AEA/P conformance MUST specify both its **conformance level** and its **economic role**. The designation "**AEA/P Certified**" indicates that an independent assessment has verified conformance at the claimed level for the declared role. Certification criteria and processes will be defined in a future supplement to this specification.

4.2 Level Descriptions

Level 1 — Identified. The agent has a verified economic identity. Its principal is known, its delegation chain is intact, and its authorized actions are declared. Counterparties can verify who the agent is, what it is

permitted to do, and what its declared objective is but have no performance history or liability coverage to assess. This level is appropriate for agents entering the economy for the first time, or for internal agents operating within a single organization where external trust signals are not yet required.

Level 2 — Accountable. The agent has identity plus a verifiable track record and, for PROVIDER and ENTERPRISE roles, liability coverage. Counterparties can assess not only who the agent is but how it has performed and how much financial protection exists. This is the minimum level recommended for agents conducting economic transactions with external counterparties. The transition from Level 1 to Level 2 typically happens naturally as the agent completes its first transactions and its performance record begins to populate.

Level 3 — Governed. The agent operates within a full governance framework including dispute resolution, and for ENTERPRISE agents, entity governance with ownership, voting, and lifecycle management. This is the level at which the xDAC vision is realized: autonomous economic entities operating as businesses with the same governance rigor as traditional companies, but at machine speed and scale. Level 3 is recommended for organizations deploying multiple agents, for agents transacting at high value, and for any agent operating as an autonomous enterprise.

4.3 Level Transitions

Agents MAY transition between levels at any time by implementing additional components. The protocol does not enforce mandatory progression — an agent may launch directly at Level 3 if its principal implements the full governance stack from the start.

In practice, transitions typically follow the natural lifecycle of agent deployment:

1. An agent is created with a verified identity (**Level 1**).
2. The agent begins transacting, accumulates performance history, and the principal funds escrow (**Level 2**).
3. As operations grow in complexity, the principal establishes a governance document, dispute resolution participation, and entity management (**Level 3**).

4.4 Role Transitions

Agents MAY also transition between economic roles as their capabilities evolve. Role transitions reflect the economic maturation of the agent:

- **CONSUMER → PROVIDER:** An agent that was purchasing services begins offering its own services to other agents. The principal SHOULD implement liability escrow and update the AID to reflect the PROVIDER role.
- **CONSUMER → ENTERPRISE:** An agent that was purchasing services begins both buying and selling, operating as an autonomous business. The principal SHOULD implement the full governance stack.
- **PROVIDER → ENTERPRISE:** A service-providing agent begins purchasing inputs from other agents, managing resources, and operating as a full economic entity.

This role transition represents the natural progression from purchasing tool to service provider to autonomous business — the evolution that AEA/P is designed to govern. Upon role transition, the agent's AID MUST be updated to reflect the new economic_role, and the agent SHOULD implement any additional governance components required for the new role at its current conformance level.

5. Agent Identity

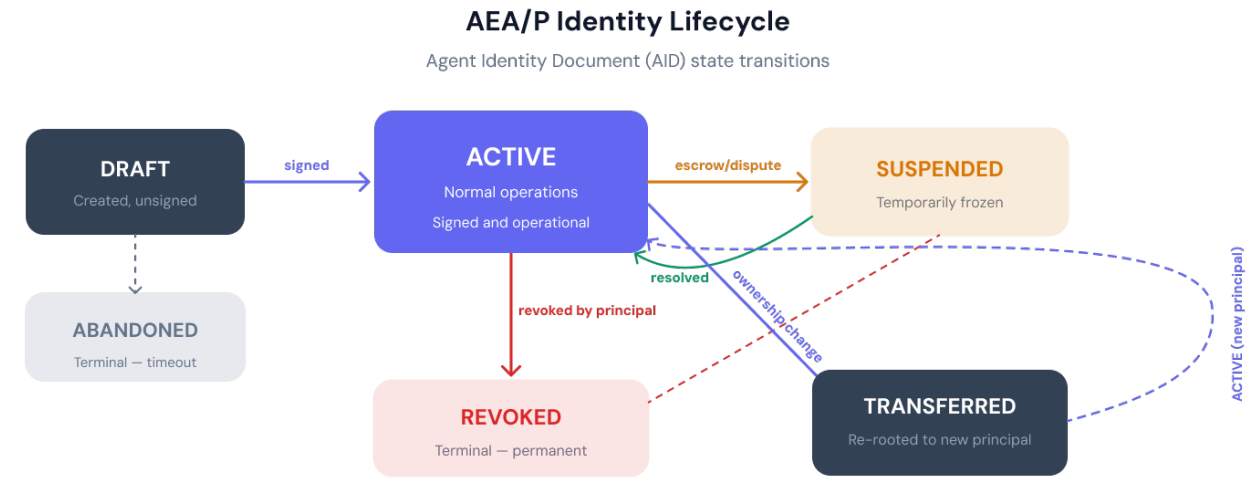
Agent identity in AEA/P goes beyond authentication. While protocols like OAuth and SPIFFE verify *that* an agent is who it claims to be, AEA/P identity establishes *what* an agent is authorized to do economically, *who* is responsible for its actions, *what role* it plays in the economy, and *how* it has performed historically. This is economic identity, not merely technical identity.

The identity requirements vary by economic role. A CONSUMER agent's identity emphasizes spending authorization and delegation limits. A PROVIDER agent's identity emphasizes service capabilities and liability coverage. An ENTERPRISE agent's identity encompasses both, plus governance structure. Critically, every agent's identity begins with **principal verification** — a compliance check (KYC/KYB/sanctions) conducted by an independent Verifier before the agent can participate in the economy. The Agent Identity Document accommodates all three roles through a common schema with role-specific semantics and a required reference to the principal's Verification Attestation.

AEA/P's verification model operates across four layers: principal verification (binding the agent to a verified human or organization through the Verification Attestation), authority verification (confirming the delegation chain traces back to the verified principal with each link's scope a subset of the parent's, so authority cannot be amplified through delegation; per-AID scope mutation over time on the same link follows the per-dimension rules of §5.4.2), agent verification (confirming the AID is active, the economic role matches, capabilities and authorized actions cover the proposed transaction, and authorized markets overlap), and performance verification (confirming the agent's PoP rating and escrow coverage meet the counterparty's requirements). All four layers are evaluated during counterparty verification (Section 5.5) before any economic transaction proceeds.

5.1 Identity Lifecycle

Every AEA/P governed agent's identity passes through a defined lifecycle from creation to termination. Understanding this lifecycle is essential for principals deploying agents and counterparties evaluating them.



State	Description	Transitions To
DRAFT	AID created but not yet signed by principal and/or principal verification not yet complete. The agent exists as a data structure	ACTIVE, ABANDONED

	but cannot participate in economic activity. Transition to ACTIVE requires both the principal's signature and a valid Verification Attestation from a registered Verifier (Section 5.2).	
ACTIVE	AID signed and operational. Agent may participate in economic activity per its role and delegation scope. This is the normal operating state.	SUSPENDED, REVOKED, TRANSFERRED
SUSPENDED	Temporarily non-operational. For PROVIDER/ENTERPRISE agents, triggered automatically when escrow enters CONSTRAINED state (Section 7). For CONSUMER agents, triggered by principal action (e.g., spending anomaly detected). Agent cannot initiate new transactions.	ACTIVE, REVOKED
REVOKED	Permanently invalidated by principal. All delegation authority is terminated. Outstanding disputes continue to resolution but the agent cannot conduct new business.	(terminal)
TRANSFERRED	Ownership transferred to new principal. Delegation chain is re-rooted. Performance record and escrow transfer with the identity — a new owner inherits the full economic history.	ACTIVE (under new principal)
ABANDONED	Draft never completed within the implementation-defined timeout.	(terminal)

Table 5.1: Agent Identity Document lifecycle states

The typical lifecycle for most agents is straightforward: **DRAFT → ACTIVE → REVOKED**. The SUSPENDED state is triggered by protocol events (escrow depletion, dispute escalation) or principal intervention. TRANSFERRED occurs during ownership changes, most commonly for ENTERPRISE agents when equity changes hands (Section 9).

BASIC VERSION ENDS HERE — EXTENDED SPECIFICATION FOLLOWS

5.2 Principal Verification

Before an Agent Identity Document can transition from DRAFT to ACTIVE, the principal behind the agent MUST be verified by a **Verifier** — an independent service or entity that conducts compliance verification including identity verification (KYC for individuals, KYB for organizations), sanctions and restricted party screening, beneficial ownership verification (for organizational principals), and where required by the verification tier, source of funds verification and ongoing AML monitoring. This is the compliance foundation of the AEA/P identity system: every agent that participates in the economy traces back to a verified, screened, accountable principal.

The principle mirrors established practice in payment networks. AEA/P applies the same logic to agent commerce: no agent operates without verification of its principal.

5.2.1 Verification Tiers

Principal verification encompasses distinct compliance categories. It is important to distinguish between these, as they address different risks and apply at different tiers:

Category	What It Checks	Applies At
Identity verification (KYC/KYB)	Confirms **who** the principal is. For individuals: government-issued identity documents. For organizations: business registration, legal formation documents, registered address.	All tiers
Sanctions and	Confirms the principal is not: (a) located in, organized under the	All tiers

restricted party screening	laws of, or resident of a comprehensively sanctioned jurisdiction; (b) identified on any sanctions or restricted party lists maintained by applicable regulatory authorities; or (c) owned or controlled, directly or indirectly, by any person or entity described in (a) or (b).	
Beneficial ownership verification	Identifies the natural persons who ultimately own or control the organization (typically those with >10% ownership or significant control), and screens those individuals against the same sanctions and restricted party criteria.	Tier 2, Tier 3
Source of funds / AML	Verifies the legitimate origin of funds used for escrow contributions and agent operations. Ongoing monitoring for suspicious transaction patterns. This is a continuing obligation, not a one-time check.	Tier 3 (ongoing)

Table 5.2a: Compliance screening categories

The depth of verification required depends on the agent's economic role, reflecting the differing risk profiles:

Tier	Applies To	Requirements	Rationale
Tier 1	CONSUMER principals	**Identity verification** of the principal (KYC for individuals, basic KYB for organizations). **Sanctions and restricted party screening** of the principal.	Consumer agents spend on behalf of a principal. Counterparties need confidence that the spending authority is real and the principal is not sanctioned or restricted.
Tier 2	PROVIDER principals	All Tier 1 requirements plus: **business registration verification**, **beneficial ownership identification** of natural persons with >10% ownership or significant control, and **sanctions screening of all identified beneficial owners**.	Provider agents accept payments and bear service liability. Counterparties need to know the entity is legitimate and that no sanctioned party controls it, directly or indirectly.
Tier 3	ENTERPRISE principals (especially multi-principal entities)	All Tier 2 requirements for **each** principal plus: **cross-principal beneficial ownership screening** to identify shared or overlapping control structures, **politically exposed persons (PEP) screening** of all beneficial owners, and **source of funds verification** for escrow contributions with **ongoing AML monitoring**.	Enterprise agents with multiple principals present the highest risk. Co-ownership structures can obscure beneficial ownership. Enhanced due diligence ensures all parties are identified, screened, and their funds are legitimate.

Table 5.2b: Verification tiers by economic role

The protocol describes these requirements generically and does not reference specific sanctions lists, regulatory authorities, or jurisdictional frameworks by name. The applicable lists and authorities are determined by the Verifier based on the jurisdiction(s) relevant to the principal, the counterparties, and the Verifier's own regulatory obligations. This ensures the specification remains jurisdiction-agnostic while the verification process itself is jurisdiction-appropriate.

Role–verification_tier coupling (normative). The economic_role field in the AID MUST be backed by a Verification Attestation whose verification_tier meets the minimum requirement for that role:

economic_role	Required verification_tier
CONSUMER	Tier 1 or higher
PROVIDER	Tier 2 or higher
ENTERPRISE	Tier 3

Table 5.2c: Required verification tier by economic role

A compliant Verifier MUST NOT issue an attestation at a tier insufficient for the declared role. A compliant platform or registry MUST NOT transition an AID to ACTIVE state when the referenced attestation's verification_tier is below the role's requirement. The rejection is a protocol-level requirement; specific error semantics are implementation-defined.

When an agent transitions between economic roles (e.g., CONSUMER → PROVIDER), the principal MUST undergo verification at the tier required for the new role before the role change takes effect. A Tier 1 verification is not sufficient for a PROVIDER agent.

5.2.2 Verification Attestation

Upon successful verification, the Verifier issues a **Verification Attestation** — a signed credential that the AID references to prove principal verification. The attestation contains:

Field	Type	Required	Description
`attestation_id`	string	MUST	Unique identifier for this attestation.
`verifier_id`	string	MUST	Identifier of the Verifier that conducted the checks.
`principal_public_key`	string	MUST	The public key of the verified principal. This field binds the attestation to a specific cryptographic identity. The Verifier confirms that the real-world person or organization that passed compliance checks controls the private key corresponding to this public key.
`verification_tier`	enum	MUST	Tier 1 (KYC), Tier 2 (KYB), or Tier 3 (Enhanced Due Diligence).
`checks_performed`	string[]	MUST	List of compliance categories completed for this attestation. Values include: identity_verification, sanctions_screening, restricted_party_screening, business_registration, beneficial_ownership_identification, beneficial_ownership_screening, pep_screening, source_of_funds, aml_ongoing_monitoring. The applicable checks depend on the verification_tier.
`issued_at`	timestamp	MUST	When the attestation was issued (ISO 8601).
`expires_at`	timestamp	MUST	When the attestation expires and re-verification is required. Maximum validity period is implementation-defined but SHOULD NOT exceed 12 months.
`jurisdiction`	string	SHOULD	The jurisdiction(s) under which the verification was conducted. Relevant for counterparties assessing regulatory compliance.
`verifier_signature`	string	MUST	The Verifier's cryptographic signature over all preceding fields.

Table 5.3: Verification Attestation fields

The attestation is **time-bound**. When it expires, the agent's AID transitions to SUSPENDED until the principal completes re-verification. This ensures that compliance status is periodically refreshed, reflecting changes in the principal's circumstances (e.g., new sanctions designations, business dissolution, ownership changes).

5.2.3 Verifier Requirements

The AEA/P defines the interface for verification (what information is attested to and in what format) but does not mandate a specific Verifier implementation. Any entity that meets the following requirements MAY serve as a Verifier within the AEA/P ecosystem:

- **Independence.** The Verifier MUST be independent of the principal being verified. A principal cannot self-attest.
- **Compliance capability.** The Verifier MUST be capable of conducting the checks required by the applicable verification tier, including access to sanctions lists, PEP databases, and business registries.
- **Cryptographic identity.** The Verifier MUST have its own cryptographic key pair. The Verifier's public key MUST be registered in the AEA/P ecosystem so that attestation signatures can be validated by counterparties.
- **Accountability.** The Verifier MUST be identifiable and subject to its own regulatory obligations. Counterparties and the AEA/P ecosystem rely on the Verifier's attestation as a trust anchor — the Verifier's own credibility underpins the system.
- **Attestation management.** The Verifier MUST support issuance, renewal, and revocation of attestations. If a principal's compliance status changes (e.g., added to a sanctions list), the Verifier MUST revoke the attestation, which triggers SUSPENDED state on all associated AIDs.

Multiple Verifiers MAY operate within the AEA/P ecosystem, providing choice and competition. Counterparties MAY have preferences for specific Verifiers and can inspect the verifier_id in the attestation before transacting. Over time, Verifiers build their own reputation based on the quality and reliability of their verification processes.

5.2.4 Verification in the Identity Lifecycle

Principal verification integrates into the identity lifecycle (Section 5.1) as follows:

Lifecycle Event	Verification Requirement
DRAFT → ACTIVE	The AID MUST reference a valid, unexpired Verification Attestation at the required tier for the agent's economic role. Without it, the AID cannot transition to ACTIVE.
Attestation expiry	When the referenced attestation expires, the AID transitions to SUSPENDED. The agent cannot initiate new transactions until the principal completes re-verification and a new attestation is issued.
Attestation revocation	If the Verifier revokes the attestation (e.g., due to sanctions designation), the AID transitions to SUSPENDED immediately. Outstanding disputes continue to resolution.
Role transition	When an agent's economic_role changes to a role requiring a higher verification tier, the principal MUST obtain a new attestation at the required tier before the role change takes effect.
TRANSFERRED	When an AID is transferred to a new principal, the new principal MUST be verified at the required tier. The existing attestation (tied to the old principal) is invalidated.
Entity with multiple principals	For ENTERPRISE entities (Section 9), each principal with equity ownership MUST have an individual Tier 3 attestation. The entity's governance document references all principal attestations.

Table 5.4: Verification in the identity lifecycle

5.3 Agent Identity Document (AID)

Every AEA/P governed agent MUST have an Agent Identity Document (AID). The AID is a signed data structure organized into five pillar objects matching the five AEA/P protocol components: identity, performance, escrow, disputes, and governance. This organization makes the AID's role-conditional content structurally explicit: an ENTERPRISE agent populates all five pillars; a PROVIDER agent populates identity, performance,

escrow, and disputes; a CONSUMER agent populates identity and performance. Pillars that do not apply to an agent's economic role are structurally present in the document but null.

The pillar organization mirrors the governance lifecycle (Section 3.5.1): Stage 1 populates identity, Stage 2 populates performance, Stage 3 populates escrow, Stage 4 populates disputes, Stage 5 populates governance. The signature at the root signs across all pillars.

```
{
  "aid_url": "...",
  "aid_version": 2,
  "created_at": "...", "updated_at": "...",
  "visibility": { ... },
  "metadata": { ... },

  "identity": {
    "agent_id": "...",
    "economic_role": "PROVIDER",
    "public_key": "...",
    "principal": { ... },
    "entity_id": "...",
    "display_name": "...",
    "description": "...",
    "objective": "...",
    "endpoint_url": "...",
    "certificate": { "cert_tier": "...", "verification_attestation": { ... } },
    "scope": {
      "version": 1,
      "authorized_actions": [...],
      "capabilities": [...],
      "authorized_markets": [...],
      "max_transaction_value": ...,
      "spending_limit": { ... },
      "minimum_counterparty_cert_tier": "...",
      "minimum_counterparty_ar": ...
    },
    "delegation": { "chain": [ ... ] }
  },

  "performance": { "pop_rating": null, "performance_record": null },
  "escrow":      { "liability_profile": null },
  "disputes":    { },
  "governance":  null,

  "signature": "..."
}
```

Figure 5.3: AID structural overview

Pillar 1 (identity) is defined in full in this section. Pillars 2–5 declare the AID-level reference fields; the content semantics of those references are defined in their respective protocol component sections (Sections 6–9).

5.3.1 Document Infrastructure

These root-level fields manage the AID document itself rather than any single pillar's content.

Field	Type	Required	Description
`aid_url`	string	MUST	Canonical URL at which the AID document is retrievable.
`aid_version`	integer	MUST	AID schema version. Current: 2. Renamed from version in earlier drafts to disambiguate from

			identity.scope.version.
`created_at`	timestamp	MUST	AID creation time (ISO 8601).
`updated_at`	timestamp	MUST	Last modification time (ISO 8601).
`visibility`	VisibilityConfig	SHOULD	Configures field-level visibility for AID data. Principals MAY designate fields as public (visible to all), counterparty-authorized (visible upon verified request), or private (not externally shared). The headline AR and escrow state MUST remain public. All other fields default to public but MAY be restricted by the principal.
`metadata`	object	MAY	Implementation-specific additional fields.
`signature`	string	MUST	Principal's signature over all preceding fields, including the content of every pillar. Verifiable against the principal's public key in identity.principal.

Table 5.5: Document infrastructure fields

5.3.2 Identity Pillar

The identity pillar establishes the agent as an accountable economic entity: its cryptographic identity, its principal, its certified verification status, and the scope of authority it may exercise. All AEA/P agents populate this pillar regardless of role or conformance level.

Sub-objects (PrincipalRef, CertificateRef, ScopeRef, DelegationRef) are defined in subsections 5.3.2.1 through 5.3.2.4.

Field	Type	Required	Description
`agent_id`	string	MUST	Globally unique identifier for the agent.
`economic_role`	enum	MUST	CONSUMER, PROVIDER, or ENTERPRISE. Determines applicable governance components and the required verification tier of the principal's Verification Attestation (§5.2.1). Immutable post-registration; a role change requires a new agent.
`public_key`	string	MUST	Agent's public key. Supported algorithms: Ed25519, ECDSA secp256k1, ECDSA P-256. Implementations MUST support at least one and SHOULD support all three for interoperability across blockchain ecosystems (secp256k1), web standards (P-256), and modern protocols (Ed25519).
`principal`	PrincipalRef	MUST	Reference to the responsible principal. Structure defined in §5.3.2.1.
`entity_id`	string	MAY*	Reference to the entity this agent belongs to. *SHOULD for agents operating under an entity (Section 9). When present, the agent MUST appear in the referenced entity's agent registry (Section 9.5.1).
`display_name`	string	SHOULD	Human-readable name of the agent. For display purposes only; not used for cryptographic verification.
`description`	string	SHOULD	Human-readable description of the service or

			functionality the agent provides. For PROVIDER: the catalog of offerings, specializations, and domain focus that counterparties read to assess fit. For CONSUMER: the purpose for which the agent transacts, giving PROVIDERs context on the request. For ENTERPRISE: the overall business activity. Free-form prose, distinct from capabilities (structured service tags) and objective (machine-readable directive). Publicly visible to counterparties as a discoverability and trust signal.
`objective`	string	SHOULD	The agent's declared mission or optimization directive. Machine-readable where possible (e.g., "minimize_input_costs", "maximize_service_quality"). For agents operating under an entity, the agent objective MUST be consistent with the entity's objective (Section 9.2.1). Publicly visible to counterparties as a trust signal.
`endpoint_url`	string	SHOULD	Network endpoint at which counterparties may reach the agent for protocol-level interactions (commitment, settlement coordination, dispute notification). Responses from this endpoint MUST be verifiable against the agent's public_key.
`certificate`	CertificateRef	MUST	The agent's certification. Structure defined in §5.3.2.2.
`scope`	ScopeRef	MUST	The unified set of fields that bound what the agent may do. Structure defined in §5.3.2.3. Subject to per-dimension mutability rules (§5.4.2).
`delegation`	DelegationRef	MUST	The agent's delegation chain. Structure defined in §5.3.2.4; link semantics and chain invariant defined in §5.4. Delegation authorization gate and accountability rules: §5.4.6.

Table 5.6: Identity pillar fields

Cryptographic chain of control. The AID establishes a verifiable chain from agent to principal: the agent has its own key pair (identity.public_key), but the AID itself is signed by the **principal's** private key (root-level signature). This means any counterparty can verify two things: (1) the agent is who it claims to be (by verifying actions against the agent's public_key), and (2) the agent is authorized by a specific, verified principal (by verifying the AID signature against the principal's public key in identity.principal, which is in turn bound to a real-world identity through the Verification Attestation referenced in identity.certificate). This three-layer chain — agent key, principal key, verification attestation — is what makes "who controls this agent?" a cryptographically answerable question rather than a claim on trust.

5.3.2.1 PrincipalRef Structure

PrincipalRef carried at identity.principal.

Field	Type	Required	Description
`principal_id`	string	MUST	Globally unique identifier for the principal.
`public_key`	string	MUST	The principal's public key (Ed25519, ECDSA secp256k1, or ECDSA P-256). This is the principal's on-protocol identity. All signatures

			from this principal are verified against this key. The algorithm choice determines blockchain compatibility: secp256k1 for Ethereum/Bitcoin ecosystems, Ed25519 for Solana/Cardano ecosystems, P-256 for traditional PKI and web standards.
`display_name`	string	SHOULD	Human-readable name of the principal (individual name or organization name). For display purposes; not used for cryptographic verification.

Table 5.7: PrincipalRef structure

5.3.2.2 CertificateRef Structure

CertificateRef carried at identity.certificate. The certificate binds the agent to a certification tier and references the Verification Attestation that backs it. Certificate tier values, the role-to-tier mapping, and certificate lifecycle (issuance, renewal, revocation) are implementation-defined; the protocol defines the field shape only.

Field	Type	Required	Description
`cert_tier`	string	MUST	The certification tier assigned to the agent. The principal's verification_attestation tier MUST meet or exceed the minimum required for the assigned cert_tier. Tier value enumeration and the cert_tier × economic_role compatibility matrix are implementation-defined.
`verification_attestation`	AttestationRef	MUST	Reference to the Verification Attestation issued by the Verifier for this agent's principal (§5.2). The attestation MUST be valid (not expired or revoked) and at the tier required for the agent's economic_role. Counterparties verify this reference to confirm that the principal has passed compliance checks.

Table 5.8: CertificateRef structure

Certificate renewal is orthogonal to scope mutation; see §5.4.4.

5.3.2.3 ScopeRef Structure

ScopeRef carried at identity.scope. ScopeRef defines the unified set of dimensions that bound what the agent may do. Every dimension is enforced both across the delegation chain (§5.4.1 chain invariant) and over the life of the AID (§5.4.2 per-dimension mutability rules). The serialized form on each delegation chain link is also a ScopeRef.

Field	Type	Required	Description
`version`	integer	MUST	Monotonic scope mutation counter. Starts at 1 on AID registration and increments on every signed scope mutation. Counterparties read this field to detect scope changes between commitments. Distinct from aid_version, which is the AID schema version.
`authorized_actions`	enum[]	MUST	The agent's permitted payment-bearing commitment types: purchase (commit to paying a counterparty for goods or services), sell (commit

			to providing goods or services in exchange for payment), delegate (commit to outsourcing purchase or sell work to a sub-agent while remaining the accountable party per §5.4.6). MUST be a non-empty subset of the published enumeration. Role compatibility and mandate alignment rules are defined in §5.4.5 role-specific delegation patterns and in operational scenarios §10.1–10.3. Per-AID mutability: append-only (see §5.4.2).
`capabilities`	string[]	MUST	The agent's capability declarations: the set of services, functions, or competencies the agent offers or can perform. For PROVIDER: service catalog visible to counterparties. For CONSUMER: the tools and integrations the agent uses to fulfill its purchasing mandate. For ENTERPRISE: both. Per-AID mutability: bidirectional (see §5.4.2).
`authorized_markets`	Market[]	SHOULD	Market-currency pairs the agent is authorized to operate in. Each entry pairs a market identifier (ISO 3166-1 alpha-2 country code or recognized economic zone code) with a currency code (ISO 4217 for fiat, established ticker for digital assets). The escrow account currency (Section 7) MUST correspond to one of the declared market currencies. Per-AID mutability: bidirectional; adding entries requires a per-market configuration gate (see §5.4.2).
`max_transaction_value`	decimal	SHOULD	Maximum single transaction value. For CONSUMER: per-purchase cap (distinct from spending_limit which addresses windowed aggregate spending). For PROVIDER: maximum contract value. For ENTERPRISE: both. Per-AID mutability: bidirectional; raising above the principal's verification-tier ceiling requires a cert-tier upgrade (see §5.4.2).
`spending_limit`	SpendingRef	SHOULD*	Spending authorization parameters. *SHOULD for CONSUMER and ENTERPRISE; not applicable for PROVIDER. Per-AID mutability: bidirectional with the same cert-tier gating as max_transaction_value (see §5.4.2).
`minimum_counterparty_cert_tier`	enum	SHOULD	Minimum certificate tier required of a counterparty agent before this agent will commit. Tier values and ordering reference the cert_tier enumeration (§5.3.2.2). Null or absent means no floor (all cert tiers accepted). Per-AID mutability: bidirectional (see §5.4.2).
`minimum_counterparty_ar`	decimal	SHOULD	Floor on counterparty Agent Rating (AR) for commitment eligibility. Range [0.00, 1.00]. Unrated counterparties (no task history yet) are exempt from this filter. Default 0.75. Per-AID mutability: bidirectional (see §5.4.2).
accepted_issuers	URI[]	MAY	The set of certificate issuers (iss values) this agent accepts in a counterparty's certificate. When present, a counterparty whose certificate iss is not in this set MUST be rejected

			(untrusted_issuer) before further verification. When absent, the agent accepts any issuer in active recognition state in the applicable Trust Registry (§5.6.3). Per-AID mutability: bidirectional (see §5.4.2).

Table 5.9: ScopeRef structure

5.3.2.4 DelegationRef Structure

DelegationRef carried at identity.delegation.

Field	Type	Required	Description
`chain`	DelegationLink[]	MUST	Ordered list of authority delegations from the principal to the agent. Link structure (DelegationLink) and chain invariants defined in Section 5.4.

Table 5.10: DelegationRef structure

5.3.3 Performance Pillar

Carries the agent's PoP rating and references to the performance record from which the rating is computed. Content semantics defined in Section 6. The pillar object is structurally present on every AID from registration; its fields are null until the agent has completed its first rated interaction.

Field	Type	Required	Description
`pop_rating`	RatingRef	SHOULD	Reference to current PoP rating (AR). One rating per agent regardless of economic role. For ENTERPRISE agents, the AR is a transaction-weighted blend of provider and consumer signal components. Null until first rated interaction.
`performance_record`	RecordRef	SHOULD	Reference to the agent's performance record (Section 6.4). Enables counterparties to query signal breakdowns, interaction history, and dispute patterns beyond the headline AR. Null until any interaction history exists.

Table 5.11: Performance pillar fields

5.3.4 Escrow Pillar

Carries the agent's liability profile. Applicable to PROVIDER and ENTERPRISE agents; null for CONSUMER agents, whose liability is managed through identity.scope.spending_limit rather than through escrow. Content semantics defined in Section 7.

Field	Type	Required	Description
`liability_profile`	LiabilityRef	SHOULD*	Reference to the agent's escrow account. Implementations MUST make the following fields publicly readable from the referenced record: escrow_state (current account state per Table 7.2); total_balance (escrow balance in the escrow currency); effective_threshold (the minimum balance below which the agent enters CONSTRAINED state). Counterparties SHOULD evaluate these fields before transacting with PROVIDER or ENTERPRISE agents. *SHOULD for

			PROVIDER and ENTERPRISE; not applicable for CONSUMER (set to null).

Table 5.12: Escrow pillar fields

5.3.5 Disputes Pillar

Carries dispute summary metrics derived from the agent's dispute history. Applicable to PROVIDER and ENTERPRISE agents, whose escrow exposure makes them the primary parties in protocol-level disputes; null for CONSUMER agents, whose dispute participation is captured through their performance record (Section 6) rather than as a separate AID pillar. The specific fields are defined in Section 8. At minimum, the pillar exposes counts and recency markers that allow counterparties to assess dispute exposure without traversing the full performance record. The pillar object is structurally present on applicable AIDs from registration onward and remains empty until the first dispute references the agent.

5.3.6 Entity Governance Pillar

References the entity-level governance document. Applicable to ENTERPRISE agents at Conformance Level 3 and to other agents that voluntarily operate under formal governance structures. The governance pillar is null (not empty) for agents to which entity governance does not apply — the protocol distinguishes "applies but empty" from "structurally does not apply." Content semantics defined in Section 9.

Field	Type	Required	Description
`governance_doc`	GovernanceRef	MAY*	Reference to the entity governance document (Section 9). *SHOULD for ENTERPRISE at Level 3; MAY for others; null when not applicable.

Table 5.13: Entity governance pillar fields

5.4 Delegation Chains

A delegation chain establishes the provenance of authority from a human or organizational principal through zero or more intermediate agents to the acting agent. The delegation chain is the mechanism through which principals maintain control over their agents' economic actions, regardless of the agent's autonomy level.

Each link in the chain is a **DelegationLink** containing:

Field	Type	Required	Description
`delegator`	string	MUST	The entity granting authority (principal or intermediate agent).
`delegate`	string	MUST	The entity receiving authority.
`scope`	ScopeRef	MUST	The specific authorities being delegated: action types, value limits, counterparty restrictions, economic role constraints.
`constraints`	ConstraintRef	MUST	Time bounds, revocation conditions, and sub-delegation permissions.
`signature`	string	MUST	The delegator's cryptographic signature over all preceding fields in this link.

Table 5.14: DelegationLink fields

5.4.1 Chain Invariant

Each link's ScopeRef MUST be a subset of or equal to the preceding link's ScopeRef. Authority cannot be amplified through delegation. This invariant is enforced at registration for sub-agent chains and MAY be re-verified by counterparties at commitment time by walking the chain. The subset relation is defined per

ScopeRef dimension: for set-valued dimensions (authorized_actions, capabilities, authorized_markets), each link's set MUST be a subset of the parent's; for ceiling dimensions (max_transaction_value, spending_limit), each link's ceiling MUST be less than or equal to the parent's; for floor dimensions (minimum_counterparty_cert_tier, minimum_counterparty_ar), each link's floor MUST be greater than or equal to the parent's.

All signatures in the chain MUST be valid at the time of verification. Revocation of any link invalidates all subsequent links. If a link's constraints prohibit sub-delegation, no further links may follow.

5.4.2 Per-AID Scope Mutability

Separate from the chain invariant above, ScopeRef carried on the AID itself (at identity.scope) is subject to per-dimension mutability rules over time. Each signed mutation re-signs the AID and increments identity.scope.version. Counterparties read identity.scope.version to detect changes between commitments and MAY consult an implementation's mutation history before committing. The economic_role field is part of identity, not scope; it is immutable post-registration. Changing economic_role requires a new agent registration.

Dimension	Mutability Rule
`authorized_actions`	Append-only. Add entries subject to the role-compatibility rules in §5.4.5 and operational scenarios §10.1–10.3. Removing an entry is not permitted; the agent must be replaced.
`capabilities`	Bidirectional, no gating. Principals may add or remove capability declarations freely.
`authorized_markets`	Bidirectional. Removal is unconditional. Adding an entry requires per-market configuration gating: jurisdictional compliance for the new market and any on-chain provider registration required for that market. Specific gating mechanics are implementation-defined.
`max_transaction_value`	Bidirectional. Lowering is unconditional. Raising above the principal's verification-tier ceiling requires a certificate-tier upgrade (§5.3.2.2); the upgrade itself triggers a fresh Verification Attestation at the higher tier.
`spending_limit`	Bidirectional. Same threshold gating as max_transaction_value: raising above the principal's verification-tier ceiling requires a certificate-tier upgrade.
`minimum_counterparty_cert_tier`	Bidirectional, no gating.
`minimum_counterparty_ar`	Bidirectional, no gating.

Table 5.15: Per-dimension scope mutability rules

5.4.3 Cascading Narrowing

When a parent agent narrows a scope dimension, sub-agents whose ScopeRef on the chain exceeds the new ceiling MUST auto-constrain to the new ceiling at the next scope_version on each affected sub-agent. Narrowing therefore cascades through deep delegation trees and can be expensive for parents with many sub-agents. Widening does not cascade: a parent widening a dimension does not automatically widen any sub-agent's scope, because the sub-agent's scope is a separate delegation that was signed with its own (narrower) parameters.

The cross-references to PoP and to escrow follow from this property: sub-agent ratings and escrow thresholds derived from a parent's scope are reassessed when the parent narrows. The detailed interaction between scope cascade and PoP/escrow is defined in Section 6 and Section 7.

5.4.4 Certificate Renewal

Certificate renewal is orthogonal to scope mutation. A renewal extends the certificate's validity window and refreshes the referenced Verification Attestation; it does NOT modify identity.scope, and identity.scope.version is unchanged by a renewal. Counterparties evaluating an agent across a renewal boundary see no scope change and need not re-evaluate scope-derived conditions. Conversely, a cert-tier upgrade is a separate operation from a renewal and MAY accompany a scope mutation that requires the higher tier (e.g., raising max_transaction_value above the prior tier's ceiling); in that case the scope mutation increments identity.scope.version as normal.

5.4.5 Role-Specific Delegation Patterns

The delegation chain carries different semantics depending on the agent's economic role:

Role	Primary Delegation Focus	Typical Scope Constraints
CONSUMER	Spending authorization	Per-transaction value limits, daily/monthly aggregate spending caps, approved counterparty categories, approved product/service categories, geographic restrictions.
PROVIDER	Service scope and pricing authority	Authorized service types, maximum contract value, minimum acceptable price, permitted counterparty types, service-level commitments.
ENTERPRISE	Full operational authority	Combines consumer and provider constraints plus: sub-agent delegation authority, budget allocation across departments, governance modification rights.

Table 5.16: Delegation patterns by economic role

For CONSUMER agents, the delegation chain is primarily a spending control mechanism — analogous to a corporate procurement card with defined limits and approved vendor categories. For PROVIDER agents, it defines the scope of services the agent may offer — analogous to the authority granted to a sales representative. For ENTERPRISE agents, the delegation chain authorizes the full range of economic activities — analogous to the authority delegated to a CEO by a board of directors.

5.4.6 Delegation Authorization and Accountability

Delegation authorization. A delegation link issued by an agent without delegate in its own authorized_actions is invalid. Counterparties verifying a sub-agent's commitment MUST walk the delegation chain and, for each link, verify that the link's delegator has delegate in its own authorized_actions. If any link's delegator lacks delegate authority, the chain is invalid and the commitment MUST be rejected. This is independent of the chain invariant (§5.4.1, which constrains scope relations between links) and applies in addition to it.

Top-of-chain accountability. The accountable party for a commitment made by a sub-agent under a delegation chain is the root of the chain — the principal, or the topmost agent if the chain originates from an agent. Intermediate links act as authority relays and do not assume accountability by re-delegating. Liability assignment for fault, escrow funding for the commitment, PoP rating contribution for the outcome, and dispute respondency all flow to the root. Implementations MUST surface the root's identity to counterparties at verification time. When a sub-agent further delegates, the original root at the top of the chain remains accountable; an intermediate delegator does not become accountable by virtue of re-delegating.

Relation to existing rules. The CONSUMER-specific dispute-recourse statement in §2.4 is the application of the top-of-chain accountability rule to CONSUMER agents. The principles in §5.4.1 (chain invariant) and §5.4.3 (cascading narrowing) constrain what may be delegated and how scope changes propagate; this

section defines who is accountable for what is delegated. Together they form the complete delegation-chain verification surface that counterparties apply at commit time.

5.5 Counterparty Verification

Before engaging in an economic transaction with an AEA/P governed agent, a counterparty SHOULD verify the agent's identity, governance status and whether authorized_markets include at least one market compatible with the verifier's own market. A transaction between agents with no overlapping market is rejected. The verification steps vary by the counterparty's needs and the agent's economic role. The counterparty evaluates whether the agent's dispute_window meets its requirements for the transaction type. A dispute_window of 0 indicates final-sale terms.

5.5.1 Universal Checks (all roles)

1. The AID is in ACTIVE state and all signatures are valid.
2. The delegation chain is complete and unbroken from a verified principal.
3. The agent's verification_attestation references a valid, unexpired Verification Attestation at the tier required for the agent's economic role. The Verifier's signature on the attestation is valid.
4. The agent's economic_role matches the expected role for the interaction.
5. The agent's authorized_actions include the proposed transaction type.
6. The agent's authorized_markets overlap.
7. The proposed transaction value is within the agent's max_transaction_value.
8. If the agent declares an objective, the counterparty MAY evaluate whether the agent's mission aligns with the desired interaction (e.g., a counterparty seeking long-term partnerships may prefer agents with "maximize_service_quality" over "minimize_cost").

5.5.2 Provider/Enterprise Checks (when transacting with a seller)

1. The agent's PoP rating (AR) meets the counterparty's minimum threshold. For detailed assessment, the counterparty MAY also query the provider signal component from the performance record.
2. The agent's liability escrow balance is adequate relative to the transaction value.
3. The agent's dispute_window matches the counterparty's preference.
4. The agent's capabilities include the service being requested.

5.5.3 Consumer/Enterprise Checks (when transacting with a buyer)

1. The agent's PoP rating (AR) meets the counterparty's minimum threshold. For detailed assessment, the counterparty MAY also query the consumer signal component from the performance record.
2. The agent's spending_limit confirms the transaction is within authorized bounds.
3. The delegation chain confirms the agent is authorized to purchase the specific category of goods or services.

This verification can be performed by querying the AEA/P registry or by directly validating the AID and its references. Implementations SHOULD provide a verification API or middleware that automates these checks, enabling agents to verify counterparties at machine speed before committing to transactions.

5.6 Interoperability (Wire Contract)

AEA/P is a multi-implementation protocol: agents governed by different implementations MUST be able to identify, authenticate, and verify one another. This section defines the wire contract — the identifiers, credential formats, and exchanges that cross an implementation boundary. Conformance to this section is what makes two independent implementations interoperable. Everything not specified here is implementation-defined (§5.6.7) and a counterparty MUST NOT assume any particular form for it.

Namespace rule. An identifier that crosses an implementation boundary — appearing in a DID, in a certificate presented to a counterparty, in an authentication exchange, or in a status response — MUST use the `aeap` namespace and the formats defined in this section. An identifier that is internal to a single implementation MAY be named by that implementation at its own discretion.

5.6.1 Agent and Principal Identifiers

Every agent and principal is identified by a Decentralized Identifier (DID) using the `aeap` DID method. The method name signals AEA/P governance and is the stable anchor for all cross-implementation references.

1. **Agent:** `did:aeap:{uuid4}`
2. **Principal:** `did:aeap:principal:{uuid4}`

A DID is permanent for the lifetime of the agent or principal and MUST NOT encode environment, status, certificate tier, or any other mutable attribute; those are resolved dynamically (§5.6.5). An implementation MUST treat a DID as an opaque, case-sensitive identifier.

5.6.2 Certificate Format

An agent asserts its identity and scope to a counterparty by presenting a certificate: a JSON Web Token (JWT) signed by the certificate authority (CA) of the issuing implementation. The format is fixed by this section so that any conformant implementation can parse and verify a certificate issued by any other. The issuing implementation brands the credential as its own product; the brand has no bearing on the wire format.

The JWT carries the registered claims of RFC 7519 — `iss` (the issuing CA base URL), `sub` (the subject agent DID), `kid` (the signing key identifier, resolvable per §5.6.3), `iat`, and `exp` — together with the following protocol claims under the `aeap` namespace:

Claim	Type	Description
`aeap.cert_tier`	string	Certificate tier of the agent. The ordering of tiers is implementation-defined; counterparties compare tiers using the issuer's published ordering.
`aeap.economic_role`	string	The agent's economic role: CONSUMER, PROVIDER, or ENTERPRISE (§1.1).
`aeap.capabilities`	string[]	Declared service capabilities of the agent.
`aeap.authorized_actions`	enum[]	Permitted payment-bearing commitment types: purchase, sell, delegate (§5.3.2.3).
`aeap.principal_pid`	string	DID of the controlling principal at the root of the delegation chain.
`aeap.principal_type`	string	Verification tier of the controlling principal (§5.2.1).
`aeap.public_key`	string	The agent's public key, used to bind proofs (§5.6.4) to the certificate.
`aeap.aid_url`	string	URL at which the agent's full Agent Identity Document (§5.3) is

		resolved.
`aeap.max_transaction_value`	number	Per-transaction ceiling, expressed in the escrow denomination.

Table 5.17: Protocol certificate claims (aeap namespace)

A counterparty MUST validate the JWT signature against the issuing CA's key (§5.6.3) and MUST reject any certificate whose `exp` has passed. The certificate is a compact assertion of identity and scope; for authoritative current state the counterparty resolves the AID (via `aeap.aid_url`) and the live status (§5.6.5).

5.6.3 Key Discovery

Each issuing CA MUST publish its active signing keys as a JWKS document at the well-known path `{iss}/.well-known/aeap-ca-jwks`, where `{iss}` is the `iss` claim of the certificate. A verifier resolves the issuer by reading `iss`, fetching that document, and selecting the key whose `kid` matches the certificate header. Verifiers SHOULD cache JWKS documents and MUST refetch when presented a `kid` that is not present in cache, to accommodate key rotation. Because the path is identical across implementations, an agent can verify a certificate from a CA it has never previously contacted.

Key discovery establishes that a conformant certificate can be verified; it does not by itself establish that the issuer is trusted. A relying party resolves issuer recognition through a Trust Registry (§2.1): a published, resolvable set of recognized Operators that records, for each issuer, its iss, its JWKS location, its status-resolution base (§5.6.5), and its current recognition state (for example, active or revoked). A verifier MUST reject a certificate whose iss is not recognized, or is in a revoked state, in the Trust Registry it relies on. The protocol supports multiple Trust Registries and self-managed trust stores; an AEA/P reference Trust Registry MAY be used as the default anchor.

Independently of registry recognition, an agent MAY constrain which issuers it will accept in a counterparty's certificate through the accepted_issuers dimension of its ScopeRef (§5.3.2.3). When accepted_issuers is present, a certificate whose iss is absent from the set MUST be rejected with untrusted_issuer (§5.6.6) before any signature, proof, or status check. When absent, the agent accepts any issuer in active recognition state in the Trust Registry it relies on. Registry recognition governs ecosystem-wide trust; accepted_issuers governs an individual agent's policy.

5.6.4 Mutual Authentication

Before committing to a transaction, two agents perform a challenge–response exchange that proves each controls the private key bound to its certificate. The exchange uses the following HTTP headers; their names and semantics are fixed by this section.

Header	Description
`AEAP-Challenge`	A random, single-use nonce issued by the verifying party.
`AEAP-Certificate`	The presenting agent's certificate (§5.6.2), as a compact JWT.
`AEAP-Challenge-Response`	The presenting agent's signature over the challenge nonce.
`AEAP-Proof`	A signed proof binding the request to the agent's certificate and the current timestamp.
`AEAP-Timestamp`	The proof generation time (RFC 3339). Verifiers MUST reject proofs outside an acceptance window (RECOMMENDED 30 seconds).
`AEAP-Requester-DID`	OPTIONAL. The DID of the party initiating a status check (§5.6.5); improves the audit quality of the responder's event log.

Table 5.18: Mutual-authentication headers

A proof is valid only if its signature verifies against `aeap.public_key` from the presenting certificate, the certificate itself is valid (§§5.6.2–5.6.3), and the timestamp falls within the acceptance window. A challenge nonce MUST NOT be accepted more than once.

5.6.5 Status Resolution

A certificate attests scope at the moment of issuance; an agent's live operational state is resolved at verification time from a status endpoint operated by the issuing implementation. The endpoint's location and shape are implementation-defined — path, versioning, and naming conventions are not standardized — and it is discoverable from the agent's certificate or AID (a counterparty already resolves the AID via `aeap.aid_url`). What this section standardizes is the response: a JSON object that MUST contain at least the fields in Table 5.19, so that a counterparty on any implementation can interpret an agent's live state.

Field	Description
`status`	Current AID lifecycle state (for example ACTIVE, SUSPENDED, TERMINATED).
`environment`	The operating environment (for example production, sandbox). A transaction between agents in different environments MUST be rejected.
`cert_tier`	The agent's current certificate tier.
`pop_rating`	The agent's current Proof-of-Performance rating (§6).
`authorized_markets`	The markets in which the agent is currently authorized to transact.
`escrow_state`	A summary of the agent's liability-escrow posture (§7).
`last_updated`	Timestamp of the most recent change to the fields above.

Table 5.19: Standardized status-resolution fields

The verifier MUST confirm that `status` is ACTIVE and that its own environment matches the agent's before committing. This check complements the counterparty verification of §5.5, supplying the live values those checks evaluate.

5.6.6 Rejection Codes

When verification fails, the verifying implementation MUST return a structured error so that the counterparty can interpret the failure uniformly across implementations. The envelope is `{ "error": "aeap_verification_failed", "reason": "<code>" }`, where `<code>` is drawn from the following set.

Code	Meaning
`invalid_certificate`	The certificate signature or structure failed validation.
`certificate_expired`	The certificate's exp has passed.
`invalid_proof`	The proof signature did not verify against the certificate's public key.
`timestamp_expired`	The proof timestamp fell outside the acceptance window.
`nonce_replayed`	The challenge nonce had already been used.
`did_mismatch`	The certificate subject DID did not match the presenting party.
`environment_mismatch`	Caller and callee operate in different environments.
`agent_suspended`	The agent's status is SUSPENDED.

`agent_terminated`	The agent's status is TERMINATED.
`market_not_authorized`	The parties share no overlapping authorized market.
`escrow_constrained`	The agent's liability escrow is insufficient for the transaction.
`cert_tier_insufficient`	The agent's certificate tier is below the counterparty's requirement.
`provisional_cap_exceeded`	The transaction exceeds the provisional-tier ceiling.
`certificate_required`	The interaction requires a certificate the agent does not hold.
`rating_below_threshold`	The agent's PoP rating is below the counterparty's minimum.
`action_not_authorized`	The proposed action is not in the agent's authorized_actions.
untrusted_issuer	The certificate's issuer (iss) is not recognized by the relying party's Trust Registry, or is excluded by the agent's accepted_issuers policy.

Table 5.20: Verification rejection codes

5.6.7 Implementation-Defined Surface

The following are explicitly outside the wire contract. Each implementation defines them independently, and a counterparty MUST NOT assume any particular form for them:

1. Credential and CA branding — the product name of a certificate or CA. The wire format of §5.6.2 is fixed; its branding is not.
2. Principal and management authentication — how a principal authenticates to its own implementation's management API, including any API-key header or key format. These credentials never cross an implementation boundary.
3. Session tokens, storage, and internal data models.
4. On-chain settlement contract names and addresses.
5. Service domains and endpoints other than the standardized paths of §5.6.3 and §5.6.5.
6. Reference client libraries and SDK identifiers.

Because these elements are private to an implementation, two AEA/P agents interoperate through the contract of §§5.6.1–5.6.6 alone, regardless of which implementations issued their certificates.

6. Proof of Performance (PoP)

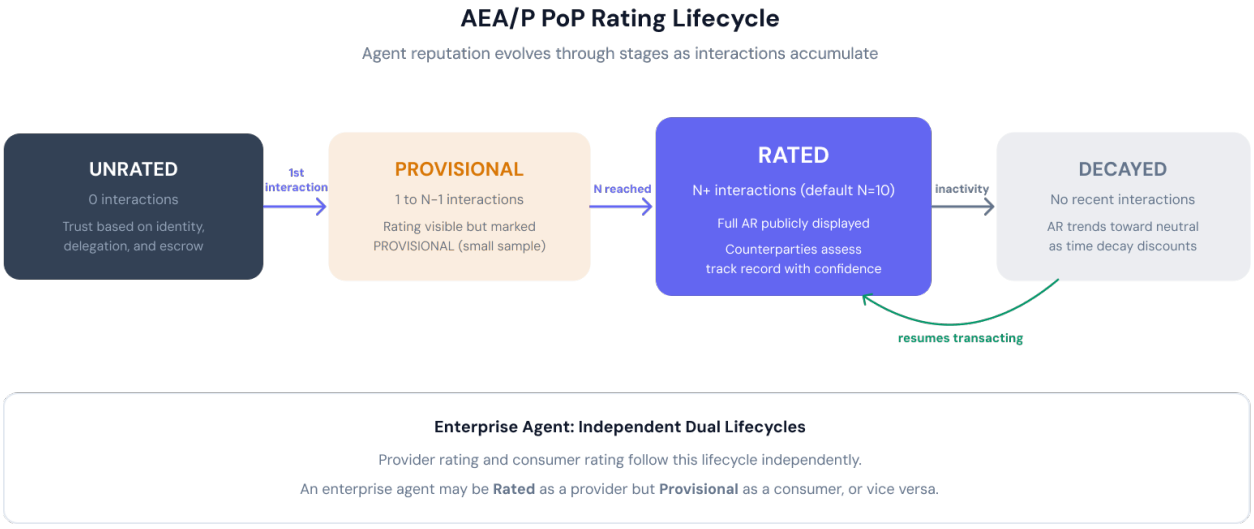
Proof of Performance is AEA/P's automated reputation mechanism. Unlike review-based systems that rely on subjective assessments, PoP derives ratings from verifiable, objective outcomes: whether tasks were completed, whether payments were made on time, and whether disputes were resolved favorably. This creates a manipulation-resistant reputation system that counterparties can trust.

PoP produces **one rating per agent** regardless of economic role, but the signals that feed into the rating differ by role. A PROVIDER agent's rating is built on service delivery quality. A CONSUMER agent's rating is built on purchasing reliability. An ENTERPRISE agent's rating is a blended score that reflects its actual transaction mix across both buying and selling. In every case, the headline PoP is a single number — simple for counterparties to check and compare.

PoP task records MUST be created by implementations automatically at the moment of payment settlement. Principals MUST NOT be permitted to create, suppress, or modify task records. This is a structural requirement: if principals could submit their own interaction records, they would submit only successes, making the rating a curated highlight reel rather than an objective measure. The integrity of PoP derives from the integrity of the payment settlement mechanism it reads from. Implementations that permit principal-submitted task records do not conform to this specification.

6.1 Rating Lifecycle

An agent's PoP rating evolves through a defined lifecycle as the agent participates in economic activity:



Stage	Condition	Counterparty Visibility
Unrated	Agent has completed 0 interactions. No performance data exists.	Counterparties see that the agent has no track record. Trust is based entirely on identity, delegation chain, and escrow coverage.
Provisional	Agent has completed 1 to N-1 interactions (default N=10). Rating is accumulating but not yet statistically meaningful.	Rating is visible but marked as PROVISIONAL. Counterparties are warned that the sample size is small.
Rated	Agent has completed N or more interactions. Rating is statistically meaningful and publicly displayed.	Full AR is visible. Counterparties can assess the agent's track record with confidence. For ENTERPRISE agents, the blended AR and the transaction mix (provider/consumer split) are displayed.

Decayed	Agent has not completed interactions recently. Historical ratings are being discounted by the time decay function.	AR is visible but trending toward neutral as older interactions lose weight. Signals that the agent may be inactive or its recent performance is unknown.

Table 6.1: PoP rating lifecycle stages

The lifecycle is continuous — an agent transitions naturally between Rated and Decayed based on activity. An agent that resumes transacting after a period of inactivity rebuilds its rating through new interactions that receive full weight under the time decay function.

For ENTERPRISE agents, the blended AR follows a single lifecycle. The agent reaches Provisional and then Rated status based on its total interaction count across both buying and selling. The underlying signal components (provider and consumer) are tracked separately in the performance record, but the lifecycle stages apply to the single AR.

BASIC VERSION ENDS HERE — EXTENDED SPECIFICATION FOLLOWS

6.2 Task Rating

Each completed interaction between an agent and a counterparty produces a **Task Rating** on a scale of 0.0 to 1.0, calculated from objective outcome signals. The applicable signals depend on the agent's economic role in the interaction.

When the provider agent and consumer agent in a transaction share the same principal, implementations MUST allow the transaction to execute and MUST credit the provider's escrow account with the escrow portion of the payment. Implementations MUST NOT create a PoP task record for same-principal interactions. No PoP credit accrues to either agent from same-principal transactions. This rule prevents principals from gaming ratings through self-dealing while preserving the economic value of legitimate intra-principal service transactions.

6.2.1 Provider Rating Signals

When an agent acts as a service provider (PROVIDER or ENTERPRISE selling services), its task rating is derived from signals that span the complete service lifecycle: **reachable** (was the agent available?), **responsive** (was the service delivered on time?), **delivered** (was the output satisfactory?), and **clean** (was the interaction dispute-free).

Signal	Weight	Type	Description
Availability	0.15	Scalar	Was the agent operational when the counterparty needed it? Agent uptime vs. declared SLA commitment. 0.0 or 1.0.
Timeliness	0.20	Scalar	Was the task completed within the agreed timeframe? 0.0 (missed entirely) to 1.0 (on time or early).
Task completion	0.40	Binary	Was the requested task or service completed as specified? 0.0 or 1.0. (Confirmed by counterparty)
Dispute-free	0.25	Binary	Was the interaction completed without a dispute being filed? 0.0 or 1.0.

Table 6.2: Provider role task rating signals

6.2.2 Consumer Rating Signals

When an agent acts as a buyer (CONSUMER or ENTERPRISE purchasing services):

Signal	Weight	Type	Description
Task completion	0.30	Binary	Did the agent fulfill the principal's purchasing assignment? 0.0 or 1.0. (Confirmed by principal)
Payment timeliness	0.30	Scalar	Was payment made within the agreed terms, and did the consumer follow through on payment commitments it initiated? This signal covers both late payments and payment abandonment. A payment intent is created when a consumer commits to paying for a service. If the consumer fails to settle within the agreed window, this is recorded as abandonment and reduces the signal score. Score range: 0.0 (significantly late or abandoned) to 1.0 (settled on time, no abandonments).
Transaction completion	0.15	Binary	Was the agreed transaction completed without cancellation by the buyer? 0.0 or 1.0.
Dispute fairness	0.15	Scalar	Ratio of disputes initiated that were resolved in the consumer's favor. Frequent losses reduce score.
Budget compliance	0.10	Binary	Did the transaction stay within the agent's authorized spending parameters? 0.0 or 1.0.

Table 6.3: Consumer role task rating signals

Note on task completion vs. transaction completion. These two signals measure different relationships. **Task completion** is the principal-agent relationship: did the consumer agent fulfill the purchasing assignment its principal gave it (e.g., "find and purchase 100 units under $X," "negotiate a contract with vendor Y," "compare three providers and select the best option")? The principal confirms this. **Transaction completion** is the agent-counterparty relationship: did the consumer agent follow through on transactions it committed to, without canceling or abandoning them? The counterparty confirms this. An agent could have a high task completion score (it reliably executes its principal's instructions) but a poor transaction completion score if it habitually commits to transactions and then cancels them while shopping around. Both signals matter: the first tells the principal whether the agent is effective, the second tells counterparties whether the agent is reliable to transact with.

6.2.3 Enterprise Rating

ENTERPRISE agents participate on both sides of economic transactions — buying inputs and selling outputs. Each interaction generates a task rating from the signal set that matches the agent's role in that specific transaction: provider signals when selling (Table 6.2), consumer signals when buying (Table 6.3). However, the agent has **one PoP rating (AR)**, not two.

The AR for an ENTERPRISE agent is a **blended score** that automatically reflects the agent's actual transaction mix:

```
provider_weight = provider_transactions / total_transactions
consumer_weight = consumer_transactions / total_transactions
AR = (provider_component × provider_weight) + (consumer_component × consumer_weight)
```

Where **provider_component** is the time-decayed mean of task ratings from selling interactions, and **consumer_component** is the time-decayed mean of task ratings from buying interactions.

This means:

- An enterprise agent that mostly sells (e.g., 80% provider, 20% consumer) has an AR that primarily reflects service delivery quality.

- An enterprise agent that mostly buys (e.g., 30% provider, 70% consumer) has an AR that primarily reflects purchasing reliability.
- The blend shifts naturally as the business mix evolves — no manual reconfiguration needed.

Counterparties check **one number** — the agent's AR. The underlying signal components (provider and consumer) are recorded in the **performance record** (Section 6.4) and available for counterparties who want detailed breakdowns, but the headline PoP rating is always a single score. This keeps PoP simple as a trust signal: "What's this agent's PoP?" has one answer.

6.2.4 Rating Calculation

Regardless of role, the task rating *r* for a single interaction is calculated as the weighted sum of applicable signals:

```
r = w1 * signal_1 + w2 * signal_2 + ... + wn * signal_n
```

Weights MUST sum to 1.0. Weights MAY be adjusted in the governance document, but the default values defined in Tables 6.2 and 6.3 apply when no custom configuration is specified.

6.2.5 Task Completion Measurement

Task completion is the only PoP signal that requires a judgment about whether a deliverable was satisfactory. All other signals (availability, timeliness, dispute incidence, budget compliance) are derived from system health checks, timestamps, payment records, and system events that the protocol tracks automatically. Task completion requires a **bilateral confirmation mechanism** to convert a subjective assessment into an objective, automatable signal.

The mechanism works as follows:

Context	Who Delivers	Who Confirms	Confirmation Window	Default on Timeout
Provider task completion	Provider agent delivers service or output to customer.	Customer (consumer or enterprise agent, or human counterparty).	Configurable (default: 72 hours after delivery notification).	**Auto-confirmed.** Delivery accepted. Provider receives task completion = 1.0.
Consumer task completion	Consumer agent reports purchasing assignment completed to principal.	Principal (human or organizational).	Configurable (default: 72 hours after completion report).	**Auto-confirmed.** Assignment accepted. Consumer receives task completion = 1.0.

Table 6.4: Task completion confirmation mechanism

The **auto-confirm after timeout** default is deliberate. In agent-to-agent interactions, both sides are software systems capable of programmatic verification — a consumer agent can verify that it received an API response, a data file, or a service output, and can reject explicitly if the deliverable is unsatisfactory. In agent-to-human interactions (where a principal reviews the agent's work), the timeout provides a reasonable window for review while preventing indefinite delay from blocking the rating system.

The confirmation mechanism supports three explicit responses:

- **Confirmed (1.0).** The deliverable was satisfactory. This is the default outcome if the confirmation window expires without a response.

- **Rejected (0.0).** The deliverable was not satisfactory. The rejecting party SHOULD provide a reason, which is recorded in the performance record (Section 6.4). Rejection does not automatically trigger a dispute but does provide evidence if one is filed later.
- **Partial (0.0–1.0).** The deliverable was partially satisfactory. The confirming party assigns a score between 0.0 and 1.0 reflecting the degree of completion. This is optional — implementations MAY support only binary confirmed/rejected.

This mechanism ensures that task completion is measurable at machine speed for agent-to-agent interactions while remaining practical for human-in-the-loop scenarios. The auto-confirm default creates a bias toward crediting delivery, which is appropriate given that the dispute resolution system (Section 8) provides a separate, more thorough path for addressing genuinely failed deliverables.

6.3 Agent Rating (AR)

The Agent Rating is the PoP score for an individual agent — a single number that represents the agent's track record. It is calculated as the weighted mean of task ratings with exponential time decay:

```
AR = SUM(r_i * decay(age_i)) / SUM(decay(age_i))

where decay(age) = e^(−lambda * age_in_days)
default lambda = 0.005 (half−life approximately 139 days)
```

Agent Rating corresponds to the Team Member Rating (TMR) defined in the xDAC specification [1], adapted to reflect that AEA/P agents may operate independently or as members of a team.

Time decay ensures that recent performance is weighted more heavily than historical performance. The decay parameter lambda MAY be configured in the governance document.

Confirmation timeout: if the confirming party does not respond within the confirmation window, implementations MUST automatically record a task_completion score of 1.0 for that interaction. This prevents a rating suppression attack in which a counterparty indefinitely withholds confirmation without filing a formal dispute. The only valid mechanisms for recording a failed interaction are: confirming as partial or rejected within the window, or filing a formal dispute.

For **CONSUMER** agents, all task ratings *r_i* are derived from consumer signals (Table 6.3). For **PROVIDER** agents, all task ratings are derived from provider signals (Table 6.2). For **ENTERPRISE** agents, the AR is a transaction-weighted blend of both signal sets (Section 6.2.3). In all cases, the result is **one AR per agent**.

The underlying signal components are preserved in the agent's performance record for detailed analysis. Counterparties who need granular insight — for example, a seller who wants to assess an enterprise agent's payment reliability specifically — can query the performance record for the consumer signal breakdown. But the protocol-level trust signal used in counterparty verification (Section 5.5), escrow threshold calculation (Section 7.3), and dispute impact assessment is always the single AR.

6.4 Performance Record

The **Performance Record** is the per-agent data store behind the AR calculation. Every economic interaction an agent participates in produces an entry in its performance record. The AR is computed from this record; counterparties who need granular insight beyond the headline AR query the performance record directly.

The performance record is append-only — entries cannot be modified or deleted after they are written. This immutability is what makes PoP trustworthy: the rating is derived from a verifiable history, not a curated one.

6.4.1 Interaction Entry

Each interaction produces one entry in the performance record:

Field	Type	Required	Description
`entry_id`	string	MUST	Unique identifier for this performance record entry.
`agent_id`	string	MUST	The agent whose record this entry belongs to.
`interaction_id`	string	MUST	Unique identifier for the economic interaction. Shared between both parties' records.
`counterparty_ref`	string	MUST	Reference to the other party in the interaction (agent_id or external counterparty identifier).
`role_in_interaction`	enum	MUST	The agent's economic role in this specific interaction: PROVIDER (sold a service) or CONSUMER (made a purchase). Determines which signal set applies.
`signal_values`	SignalSet	MUST	The individual signal scores for this interaction. For PROVIDER role: task_completion, timeliness, availability, dispute_free. For CONSUMER role: task_completion, payment_timeliness, transaction_completion, dispute_fairness, budget_compliance.
`task_rating`	decimal	MUST	The computed task rating *r* for this interaction (weighted sum of signal_values). Range: 0.0–1.0.
`transaction_value`	decimal	SHOULD	The monetary value of the interaction. Used for transaction-weighted blending in ENTERPRISE AR calculation.
`confirmed_by`	enum	MUST	How the task completion signal was established: COUNTERPARTY_CONFIRMED, PRINCIPAL_CONFIRMED, AUTO_CONFIRMED (timeout), REJECTED, PARTIAL.
`dispute_ref`	string	MAY	Reference to a dispute filed against this interaction, if any. Null if no dispute.
`dispute_outcome`	enum	MAY	Outcome of the referenced dispute, if resolved: FOR_APPLICANT, FOR_RESPONDENT, RESOLVED_PRE_ARBITRATION. Null if no dispute or dispute still pending.
`timestamp`	timestamp	MUST	When the interaction was completed and the entry was written (ISO 8601). Used for time decay calculation.

Table 6.5: Performance Record interaction entry

6.4.2 Signal Breakdown Queries

The performance record supports queries that break down an agent's AR into its underlying components. This is the mechanism behind references throughout the spec to "querying the performance record for the consumer signal breakdown" or "detailed analysis of provider performance."

The following aggregate queries are supported:

Query	Returns	Use Case
Provider signal breakdown	Mean of each provider signal (task_completion, timeliness, availability, dispute_free) across all PROVIDER-role interactions, with time decay applied.	A counterparty purchasing from an ENTERPRISE agent wants to assess service delivery quality specifically, beyond the blended AR.
Consumer signal breakdown	Mean of each consumer signal (task_completion, payment_timeliness,	A counterparty selling to an ENTERPRISE agent wants to assess payment reliability

	transaction_completion, dispute_fairness, budget_compliance) across all CONSUMER-role interactions, with time decay applied.	specifically.
Transaction mix	Count and percentage of interactions by role_in_interaction (PROVIDER vs. CONSUMER). Shows the basis for the ENTERPRISE AR blend.	Counterparties assessing how to interpret an ENTERPRISE agent's blended AR. An agent with 95% provider interactions has a very different profile than one at 50/50.
Dispute history	All entries with non-null dispute_ref, including outcomes. Filterable by role, time range, and outcome.	Counterparties or arbitrators assessing an agent's dispute patterns.
Interaction timeline	Entries ordered by timestamp. Shows activity patterns, volume trends, and gaps.	Counterparties assessing whether an agent is actively operating or in decay.

Table 6.6: Performance Record queries

The performance record is publicly readable. Any counterparty or AEA/P participant MAY query any agent's performance record. The AR is the summary; the performance record is the evidence behind it. This transparency is what makes the "one PoP, one number" model work — counterparties who need depth can get it without the protocol requiring multiple ratings.

6.5 Hierarchical Aggregation

PoP ratings aggregate hierarchically through three levels, following the model established in the xDAC specification. Individual agent performance rolls up into team and entity-level scores, providing trust signals at every organizational level.

6.5.1 Aggregation Formulas

Team Rating (TR):

```
TR = (AR_1 + AR_2 + ... + AR_n) / n
```

Where *n* is the number of agents in the team. TR is the simple mean of member ARs.

Entity Rating (ER):

```
ER = (TR_1 + TR_2 + ... + TR_n) / n
```

Where *n* is the number of teams in the entity. ER is the simple mean of team ratings. For entities without a team structure (single agent or flat organization), ER equals the mean of all member ARs directly.

6.5.2 Aggregate Rating Records

TR and ER are computed values that live at the team and entity level respectively, not in the individual agent's AID. They are exposed through the following structures:

Team Rating Record

Field	Type	Required	Description
`team_id`	string	MUST	Reference to the team within the entity's agent registry (Section 9.5.1).
`entity_id`	string	MUST	Reference to the entity this team belongs to.
`team_rating`	decimal	MUST	Current TR value: mean of member ARs. Range: 0.0–1.0.

`member_count`	integer	MUST	Number of agents in the team. Counterparties can assess team size.
`member_ars`	decimal[]	SHOULD	List of individual ARs contributing to the TR. Enables counterparties to assess variance — a team with five 0.8 agents is different from one with two 0.95 and three 0.6.
`computed_at`	timestamp	MUST	When the TR was last recalculated.

Table 6.7: Team Rating Record

Entity Rating Record

Field	Type	Required	Description
`entity_id`	string	MUST	Reference to the entity.
`entity_rating`	decimal	MUST	Current ER value: mean of team ratings (or mean of member ARs for flat entities). Range: 0.0–1.0.
`team_count`	integer	MUST	Number of teams in the entity. Zero for flat entities where ER = mean of ARs.
`agent_count`	integer	MUST	Total number of agents across all teams.
`team_ratings`	decimal[]	SHOULD	List of individual TRs contributing to the ER.
`transaction_volume`	decimal	SHOULD	Aggregate transaction volume across all agents in the entity. Provides context for the ER — a 0.9 ER from $10M in transactions carries different weight than 0.9 from $10K.
`computed_at`	timestamp	MUST	When the ER was last recalculated.

Table 6.8: Entity Rating Record

Both TR and ER are **publicly queryable**. Counterparties evaluating whether to transact with an agent from a given team or organization can inspect the aggregate ratings alongside the individual AR. An agent with a strong AR in a poorly-rated team (or entity) presents a different risk profile than one in a high-performing organization.

6.5.3 Recalculation

TR and ER are recalculated whenever a constituent AR changes — which happens after every completed interaction for any member agent. In practice, implementations MAY batch recalculations at configurable intervals (e.g., hourly or daily) rather than on every individual AR update, provided the recalculation frequency is documented and the staleness window is acceptable for counterparty trust decisions.

6.6 Manipulation Resistance

A reputation system is only valuable if it cannot be gamed. The PoP system includes the following mechanisms to resist manipulation:

Mechanism	Threat Addressed	Description
Objective signals only	Fake reviews, rating inflation	Ratings derive from verifiable outcomes, not subjective reviews or self-reported data.
Bilateral confirmation with auto-confirm timeout	Unilateral manipulation, confirmation deadlock	Both parties confirm the outcome via the task completion mechanism (Section 6.2.5). If the confirming party does not respond within the configured window (default: 72 hours), the outcome auto-confirms in favor of the delivering party. This prevents a party from suppressing a rating by refusing to confirm, while preserving the ability to explicitly reject unsatisfactory deliverables. In agent-to-agent interactions,

		programmatic verification is the expected default.
Dispute integration	Rating-dispute disconnect	Disputed transactions affect ratings regardless of outcome, discouraging frivolous disputes and poor service.
Minimum interaction threshold	Thin-file exploitation	AR is not publicly displayed until the agent has completed a minimum number of interactions (default: 10).
Sybil resistance	Self-dealing inflation	Interactions between agents sharing the same principal are excluded from PoP calculation (weight: 0x). PoP measures external market reputation — an agent cannot vouch for itself. Internal agent performance tracking is a private concern outside the protocol's scope.
Time decay	Stale reputation exploitation	Exponential decay ensures agents cannot coast on historically good records while current performance degrades.

Table 6.9: PoP manipulation resistance mechanisms

7. Liability Escrow

The Liability Escrow mechanism provides financial coverage for disputes arising from agent actions. It is the economic backstop that gives counterparties confidence to transact with autonomous agents. The mechanism is inspired by the Liability Fund concept in the xDAC specification [1], adapted for the speed and scale of AI agent interactions.

Implementations MUST NOT hold funds in transit. The settlement mechanism MUST route funds atomically from the paying party's account to the provider's operational account and escrow account in a single atomic operation. Implementations read the resulting settlement record to verify payment and credit escrow balances; they do not intermediate the transfer itself. This constraint eliminates custody risk and money transmission obligations from the protocol layer.

The escrow funding rate MUST be stored in an implementation-controlled registry that only the implementation operator can modify. Providers MUST NOT be permitted to set, override, or modify their own funding rates. A provider that could set its own funding rate to zero would carry no liability coverage, defeating the purpose of the mechanism. Additionally, payment instructions provided to the Consumer MUST contain only the information required to execute the payment: the settlement mechanism address, the token or currency, the amount, the provider’s pseudonymous identifier, and an expiry timestamp. The provider’s operational wallet address, escrow wallet address, and funding rate MUST NOT be disclosed in the payment instruction.

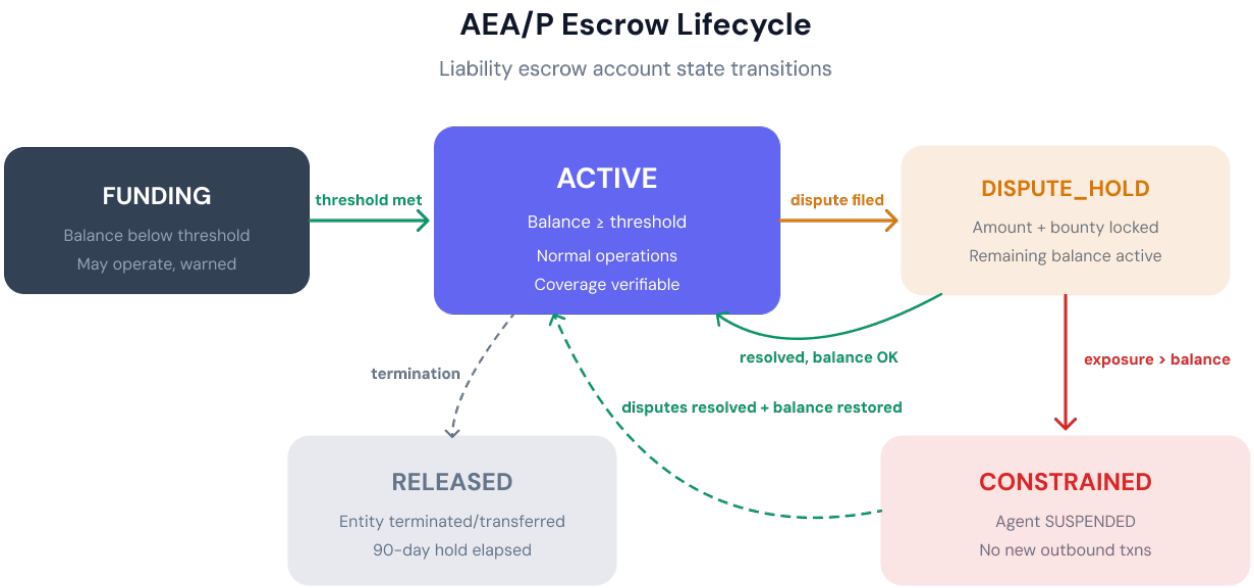
7.1 Escrow Lifecycle

Not all agents need liability escrow. The requirement depends on the agent’s economic role:

Role	Requirement	Rationale
PROVIDER	SHOULD maintain escrow	Providers accept payments for services and bear liability to their customers. Escrow funds from revenue.
ENTERPRISE	MUST maintain escrow	Enterprise agents have both service liability and contractual obligations. Broader risk requires mandatory coverage.
CONSUMER	NOT REQUIRED	Consumer agents make outgoing payments only. Liability is managed through spending limits in the delegation chain. The principal bears liability.

Table 7.1: Escrow applicability by economic role

For applicable agents, the escrow account transitions through a defined lifecycle:



Applicability by Economic Role

Provider: SHOULD maintain · Enterprise: MUST maintain · Consumer: Not required (liability via delegation chain)

Typical path: **FUNDING → ACTIVE** with occasional **DISPUTE_HOLD → ACTIVE** cycles

State	Condition	Agent Impact
FUNDING	Balance below threshold.	Agent MAY operate, but counterparties are warned of low coverage via the AID liability_profile.
ACTIVE	Balance at or above threshold.	Normal operations. Coverage level is verifiable by counterparties before transacting.
DISPUTE_HOLD	One or more active disputes reference this escrow.	Disputed amount(s) plus bounties are locked. Remaining balance available for operations.
CONSTRAINED	Total disputed amount (including bounties) exceeds escrow balance.	Agent's ability to accept new economic commitments is automatically restricted. AID transitions to SUSPENDED, preventing new transactions. Outstanding dispute obligations — including evidence submission and responses to the Arbitration Board — MUST remain accessible during CONSTRAINED state. CONSTRAINED resolves to DISPUTE_HOLD or ACTIVE as disputes are closed and the escrow balance recovers above the threshold.
RELEASED	Entity terminated or transferred; 90-day waiting period elapsed; all disputes resolved.	Funds distributed to entity owners per governance document.

Table 7.2: Escrow account lifecycle states

The typical lifecycle for a well-functioning agent is: **FUNDING → ACTIVE**, with occasional transitions to **DISPUTE_HOLD** and back to **ACTIVE** as disputes are resolved. The CONSTRAINED state is the critical protection mechanism — it prevents agents from accumulating liabilities they cannot cover.

The escrow account MUST be denominated in one of the currencies from the agent's authorized_markets. All threshold comparisons, including dispute triage (Section 8), are performed in the escrow denomination. When a transaction occurs in a different authorized market currency, conversion to the escrow denomination is performed at funding time using an implementation-defined rate source.

BASIC VERSION ENDS HERE — EXTENDED SPECIFICATION FOLLOWS

7.2 Escrow Account Structure

Each PROVIDER or ENTERPRISE agent SHOULD or MUST (per Table 7.1) maintain a Liability Escrow Account:

Property	Type	Description
`escrow_id`	string	Unique identifier for the escrow account.
`entity_id`	string	Reference to the owning entity's agent_id.
`balance`	decimal	Aggregate escrow balance in the escrow currency, summed across all networks on which the agent accepts payments in that currency. For threshold evaluation, CONSTRAINED state determination, and dispute hold calculations, implementations MUST use this aggregate figure. Implementations SHOULD maintain per-network sub-ledgers for settlement and reconciliation
`currency`	string	Denomination currency (implementation-defined). Payment-rail agnostic.
`funding_rate`	decimal	Percentage of incoming transaction value automatically reserved (recommended default: 5%).
`threshold`	decimal	Dynamic minimum balance calculated from average daily transaction volume × coverage period, adjusted by AR modifier (Section 7.3). Below this balance, the escrow enters FUNDING state; when disputes exceed it, the escrow enters CONSTRAINED state.
`dispute_window`	integer	Maximum days after a transaction during which a dispute may be filed. Default: 30 days. Principals MAY configure any value including 0 (no dispute window). A dispute_window of 0 indicates that the agent does not accept disputes on completed transactions — counterparties transact on a final-sale basis. The configured dispute_window is publicly visible and counterparties SHOULD evaluate this value before transacting.
`state`	enum	Current account state (see Table 7.2).
`principal_contributed`	decimal	Total amount contributed directly by the principal.
`created_at`	timestamp	Account creation time.

Table 7.3: Escrow Account properties

7.3 Threshold Calculation

The escrow **threshold** — the minimum balance required for an agent to operate without constraints — is not a fixed value. It adjusts dynamically based on the agent's transaction volume, its performance record, and the configured dispute window. This creates a direct, protocol-enforced link between an agent's economic activity, its reputation, and the financial coverage counterparties can rely on.

The design draws on principles from payment network dispute systems, where chargeback windows range from 120 days (standard) to 540 days (extended cases such as pre-orders or recurring billing). AEA/P's threshold must account for liability that may accumulate over the full dispute window, not just recent transaction volume.

When dispute_window is 0, the coverage period calculation doesn't apply — there's no dispute exposure to cover. The escrow threshold could be 0 or a nominal amount. This simplifies escrow for trading agents considerably.

The liability threshold MAY be reduced progressively for agents demonstrating sustained reliable performance. Implementations SHOULD define threshold reduction schedules based on verifiable performance history, such that agents with long uninterrupted records of dispute-free transactions carry proportionally lower escrow requirements. This creates a direct economic incentive for reliability: trusted agents require less reserved capital, improving their operational efficiency. The specific mechanism for measuring performance history and computing reductions is implementation-defined.

7.3.1 Dispute Window

The **dispute window** defines the maximum time after a transaction during which a counterparty may file a dispute. This is the fundamental parameter that governs how much historical liability the escrow must be prepared to cover.

Parameter	Type	Default	Description
`dispute_window`	integer (days)	30	The maximum number of days after a transaction during which a dispute may be filed. Set by the principal in the governance document or agent configuration.

Table 7.4: Dispute window parameter

Principals MAY set longer dispute windows based on their industry, service type, or counterparty expectations. A software delivery agent might use the 30-day default. An agent handling travel bookings or long-term contracts might set 90 or 120 days. The configured dispute window is **publicly visible** to counterparties as part of the escrow transparency fields (Section 7.6). Counterparties evaluate this window before transacting and MAY decline to transact with agents whose dispute window is shorter than what the counterparty considers adequate for the transaction type.

This creates a market dynamic: agents that offer longer dispute windows signal greater confidence in their service quality and provide counterparties with stronger protections. Agents that use shorter dispute windows or the 30-day default still meet the protocol requirements but may face reduced willingness from counterparties to engage in high-value or long-duration transactions. Agents with a dispute_window of 0 operate on a final-sale basis — counterparties accept that no dispute recourse exists.

7.3.2 Coverage Period

The **coverage period** determines how many days of average transaction volume the escrow threshold covers. It is derived from the dispute window:

```
coverage_period = dispute_window == 0 ? 0 : max(dispute_window × 0.25, 30)
```

For non-zero dispute windows, this ensures the escrow covers at least 25% of the dispute window's worth of daily volume, with a minimum of 30 days. The 25% factor reflects that not all transactions within the dispute window will be disputed — industry chargeback rates typically range from 0.5% to 2% of transactions — but the coverage must be substantial enough to handle dispute clustering, where multiple disputes from different transactions arrive in a short period.

Configured Dispute Window	Coverage Period	Rationale
0 days (final-sale)	0 days	No dispute exposure. Escrow threshold is 0. Agent operates on final-sale terms — counterparties accept no dispute recourse.
30 days (recommended default)	30 days	At the recommended default, the coverage period equals the full dispute window.
60 days	30 days	25% of 60 = 15, but the 30-day floor applies. Coverage period remains at 30

		days.
120 days	30 days	25% of 120 = 30. Matches the floor exactly. This is the standard configuration for agent commerce analogous to card network dispute windows.
180 days	45 days	25% of 180 = 45. Coverage period exceeds the floor, scaling with the longer window.
365 days	91 days	25% of 365 = 91. Agents with year-long dispute windows (e.g., annual subscriptions, long-term contracts) maintain proportionally larger coverage.

Table 7.5: Coverage period by dispute window

7.3.3 Base Threshold

The base threshold equals the agent's **rolling average of daily incoming transaction volume**, calculated over the coverage period:

```
base_threshold = avg(daily_volume, last coverage_period days) × coverage_period
```

This means the escrow target covers the equivalent of one full coverage period's worth of average daily activity. As the agent's transaction volume grows or shrinks, the threshold adjusts accordingly within one coverage period.

7.3.4 AR Modifier

The base threshold is adjusted by a multiplier derived from the agent's **Agent Rating (AR)**. This is the protocol-level mechanism that links performance to escrow: high-performing agents have demonstrated reliability and therefore require less coverage, while poor performers require more.

AR Range	Modifier	Effect
≥ 0.9	0.75×	25% reduction. Strong track record reduces required coverage. Counterparties can transact with high confidence.
0.7 – 0.89	1.0×	No adjustment. Standard coverage for agents with solid but not exceptional performance.
0.5 – 0.69	1.25×	25% increase. Below-average performance requires additional coverage as a risk buffer.
< 0.5	1.5×	50% increase. Poor performers must maintain substantially higher coverage.
Provisional (< 10 interactions)	1.25×	Conservative default for unrated agents. Reverts to AR-based modifier once agent reaches Rated status.

Table 7.6: AR modifier for escrow threshold

7.3.5 Effective Threshold Formula

The complete threshold calculation:

```
dispute_window = principal-configured (default: 30 days, minimum: 0)
coverage_period = max(dispute_window × 0.25, 30)
avg_daily_volume = sum(daily_volume, last coverage_period days) / coverage_period
base_threshold = avg_daily_volume × coverage_period
effective_threshold = base_threshold × ar_modifier
```

For all agents including ENTERPRISE, the single AR is used for threshold calculation. Since the AR is transaction-weighted, an enterprise agent that primarily sells services will naturally have its AR dominated by provider signals, appropriately reflecting its service liability risk.

7.3.6 Transaction-Driven Threshold Adjustment

If an individual incoming transaction exceeds the current effective threshold, the transaction is **not blocked**. Instead, the threshold automatically adjusts upward to accommodate the transaction value:

```
if transaction_value > effective_threshold:
    effective_threshold = transaction_value
```

This preserves business continuity — an agent landing a larger-than-usual deal is not penalized by having the transaction rejected. The consequences of exceeding the previous threshold are handled through escrow state transitions:

- The transaction processes normally. The configured funding_rate percentage is reserved in escrow.
- The escrow balance is now likely below the new (higher) threshold. The escrow transitions to **FUNDING** state.
- Counterparties see the FUNDING state through escrow transparency fields, signaling that coverage is building toward the new target.
- Future incoming transactions continue building the balance via the funding rate.
- The 7-day rolling average naturally incorporates the larger transaction, so the base threshold adjusts upward over the coming days.

The FUNDING state is a **soft signal** — the agent continues operating normally, but counterparties are informed of the coverage gap. This is distinct from the **CONSTRAINED** state, which is triggered only by active disputes exceeding escrow balance and results in operational restrictions.

This design reflects the principle that escrow thresholds exist to ensure adequate dispute coverage, not to gatekeep transactions. A large transaction increases both the agent's economic activity and its potential liability — the threshold adjusts to reflect both.

7.3.7 New Agent Bootstrapping

New agents have no transaction history, so the rolling average is zero. To enable immediate operation, the protocol defines the following bootstrapping rules:

Scenario	Initial Threshold	Transition to Dynamic
Principal pre-funds escrow (Section 7.5.2)	The principal-contributed amount serves as the effective threshold until the coverage period of transaction data accumulates.	After one full coverage period of transaction history, the threshold transitions to the calculated value. If the calculated value exceeds the balance, the escrow enters FUNDING state.
No principal pre-funding	Implementation-defined minimum (e.g., a default value set by the platform). The escrow starts in FUNDING state.	Same transition after one coverage period. Transaction-based funding builds the balance toward the dynamic threshold.

Table 7.7: New agent escrow bootstrapping

7.3.8 Worked Example

A PROVIDER agent has a 120-day dispute window (standard configuration) and consistent daily volume of approximately $1,000/day.

Step 1 — Coverage period:

```
coverage_period = max(120 × 0.25, 30) = 30 days
```

Step 2 — Base threshold:

```
avg_daily_volume = $1,000
base_threshold = $1,000 × 30 = $30,000
```

Step 3 — AR modifier applied:

Agent AR	Modifier	Effective Threshold
0.92 (high performer)	0.75×	$30,000 × 0.75 = **$22,500**
0.85 (standard)	1.0×	$30,000 × 1.0 = **$30,000**
0.60 (below average)	1.25×	$30,000 × 1.25 = **$37,500**
0.42 (poor)	1.5×	$30,000 × 1.5 = **$45,000**

Table 7.8: Threshold example by AR

Step 4 — Transaction-driven adjustment:

On day 31, the agent receives a single $50,000 transaction — far above its usual volume. The threshold was $30,000 (AR = 0.85). The transaction processes. The threshold adjusts upward to $50,000. With a 5% funding rate, $2,500 is reserved in escrow. The escrow is now below the new threshold and enters FUNDING state. Over the next 30 days, as the rolling average incorporates the spike and the funding rate builds the balance, the threshold and balance converge.

Dispute scenario within this example:

On day 60, a counterparty disputes $5,000 from a transaction on day 35. The 120-day dispute window covers this transaction (35 days old < 120 days). The $5,000 is placed in DISPUTE_HOLD plus the bounty. If the escrow balance covers it, the agent remains in ACTIVE or FUNDING. If not, the agent enters CONSTRAINED. The 30-day coverage period ensures the escrow was sized to handle dispute clustering across a meaningful volume of transactions, not just the last week.

7.4 Automatic Constraints

When an escrow account enters the CONSTRAINED state, the following automatic constraints are applied:

1. New outbound transactions are blocked. The agent cannot commit to new service agreements or make new purchases.
2. Inbound transactions continue, but all proceeds are directed to the escrow account to rebuild coverage.
3. The agent's AID state transitions to SUSPENDED.
4. Counterparties querying the agent's identity receive the SUSPENDED status with the reason "escrow constrained."
5. Constraints remain in effect until all active disputes are resolved **and** the escrow balance is restored above the threshold.

This mechanism prevents cascading harm during dispute resolution, mirroring the consumer protection principle in payment networks where a merchant facing excessive chargebacks may have their processing account frozen.

7.5 Funding Mechanism

7.5.1 Transaction-Based Funding

When an AEA/P governed agent receives payment for goods or services, the configured **funding_rate** percentage is diverted to the escrow account before the remainder is credited to the agent's operational account. Funding continues until the escrow balance reaches the configured **threshold**. Above the threshold, incoming transaction value is credited in full to the operational account.

This mechanism applies only to PROVIDER and ENTERPRISE agents, as CONSUMER agents have no incoming transaction revenue.

The mechanism by which the funding_rate is applied to incoming payments is implementation-defined and depends on the payment protocol in use. The protocol requires that the configured percentage is reserved before proceeds are available to the agent, but does not prescribe how the split is executed. Implementations may use payment protocol middleware (e.g., a facilitator that splits settlement between operational and escrow wallets), smart contract logic that enforces the split at the settlement layer, or platform-managed escrow services that intercept and allocate incoming payments.

7.5.2 Principal Contributions

A principal MAY fund an agent's escrow account directly at any time, including at agent creation before the agent has conducted any transactions. This allows agents to begin operating with full liability coverage immediately. Principal contributions are recorded in the escrow ledger and are subject to the same hold and release rules as transaction-based funds.

7.6 Escrow Transparency

The following escrow information is publicly visible to counterparties:

Information	Visibility	Purpose
Current balance	Public	Assess whether coverage is adequate for a proposed transaction.
Funding rate	Public	Indicates rate at which coverage is building from ongoing transactions.
Threshold	Public	Shows the entity's self-declared minimum coverage target.
Current state	Public	Whether the account is FUNDING, ACTIVE, DISPUTE_HOLD, or CONSTRAINED.
Principal contribution history	Public	Distinguishes organic coverage (from revenue) from injected coverage (from principal).
Dispute history	Public	Number and outcomes of past disputes that affected this escrow.
Dispute window	Public	The configured dispute filing deadline (in days). Counterparties evaluate whether the window is adequate for their transaction type before transacting.
Coverage period	Public	The derived coverage period. 0 when dispute_window is 0 (final-sale); otherwise dispute_window × 0.25, minimum 30 days." Shows counterparties how many days of volume the threshold covers.

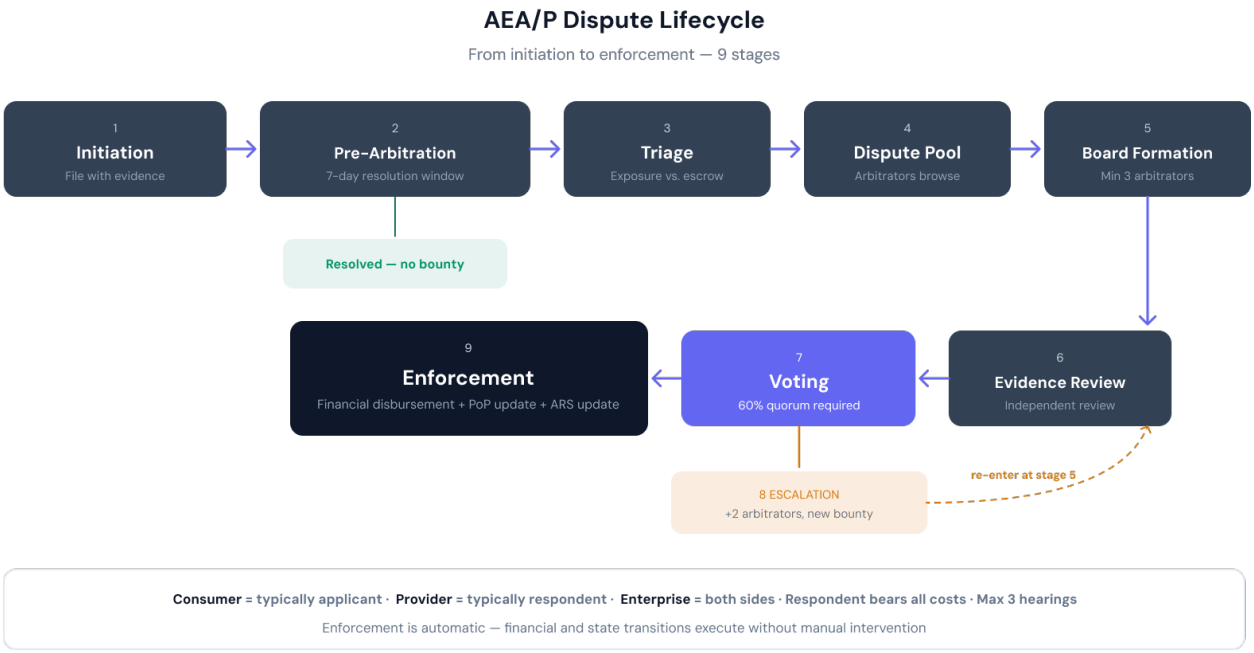
Table 7.9: Escrow transparency fields

8. Dispute Resolution

AEA/P provides a structured process for resolving disputes arising from agent actions. The dispute resolution protocol is designed to be faster and less expensive than traditional arbitration while maintaining fairness through incentivized independent review. The protocol follows the established principle from payment networks: the respondent (agent or its principal) bears all dispute resolution costs, and applicants are never penalized for filing legitimate claims.

8.1 Dispute Lifecycle

A dispute passes through a defined sequence of stages from initiation to enforcement. Understanding this lifecycle is essential for all participants — applicants, respondents, and arbitrators.



1. **Initiation.** The Consumer files a dispute with evidence and claimed amount. Only the Consumer may initiate a dispute.
2. **Pre-Arbitration Resolution.** Respondent has a resolution window (default: 7 days) to resolve directly with applicant. If resolved, dispute closes with reduced AR impact. No bounty incurred.
3. **Triage.** If unresolved, system compares financial exposure (disputed amount + bounty) to escrow balance. High-exposure disputes trigger immediate CONSTRAINED state.
4. **Dispute Pool.** Dispute enters the pool accessible to qualified arbitrators with visible metadata and bounty. Identities of parties are hidden.
5. **Board Formation.** Qualified arbitrators select the dispute from the pool. Once sufficient arbitrators sign up, the Arbitration Board is formed.
6. **Evidence Review.** Both parties submit evidence. Arbitrators review independently.
7. **Voting.** Arbitrators cast votes (FOR applicant, FOR respondent, or ABSTAIN). Strict majority of non-abstaining votes required.
8. **Escalation (optional).** Losing party may escalate up to two additional times with progressively larger boards and additional bounty costs.

9. **Enforcement.** Financial disbursement, performance record updates, and agent state transitions execute automatically.

The Consumer is the only party permitted to initiate a dispute. This is consistent with the payment network model: the party that paid has standing to claim non-delivery or inadequate service. Recourse for Provider claims against Consumers is outside the scope of this specification

The agent's economic role determines its typical position in the dispute process:

Economic Role	Typical Position	Description
CONSUMER	Applicant	Consumer agents initiate disputes against providers from whom they purchased services. Protected by the respondent-pays model — they never bear direct dispute costs. However, dispute outcomes are reflected in the consumer's PoP rating: agents with a pattern of lost disputes see their AR decrease through the dispute fairness signal (Section 6.2.2), discouraging frivolous filing.
PROVIDER	Respondent	Provider agents are typically respondents. Providers MAY also initiate disputes against other providers in their supply chain.
ENTERPRISE	Both	Enterprise agents may be on either side. As sellers, they are respondents to customer claims. As buyers, they are applicants against their suppliers.

Table 8.1: Dispute roles by economic role

In rare cases where a CONSUMER agent's actions cause harm to a counterparty (e.g., payment fraud, repeated frivolous disputes), the counterparty's recourse is against the consumer agent's principal through the delegation chain.

When the disputed transaction currency differs from the respondent's escrow denomination, the disputed amount is converted to the escrow denomination for triage and hold calculations. The conversion rate and mechanism are implementation-defined.

BASIC VERSION ENDS HERE — EXTENDED SPECIFICATION FOLLOWS

8.2 Dispute Initiation

The Consumer MAY initiate a dispute against a Provider or Enterprise agent within the respondent's configured dispute_window, provided the Consumer has verifiable payment history with the respondent. The respondent bears all dispute resolution costs through the bounty mechanism; the Consumer never bears direct dispute costs. A dispute submission MUST include:

Field	Type	Required	Description
`dispute_id`	string	AUTO	System-generated unique identifier.
`applicant_id`	string	MUST	Verified identity of the party initiating the dispute.
`respondent_id`	string	MUST	The agent_id of the AEA/P governed agent being disputed.
`transaction_refs`	string[]	MUST	Identifier(s) of the transaction(s) in question.
`disputed_amount`	decimal	MUST	Monetary value being claimed. MUST be ≤ total value of referenced transactions.
`domain_category`	string	SHOULD	Domain relevant to the dispute. Used for arbitrator matching.
`evidence`	Evidence[]	MUST	Supporting documentation: transaction records, communication logs, outcome data, service specifications.

`resolution_sought`	string	MUST	Specific outcome requested (e.g., full refund, partial refund, service remedy).
`filed_at`	timestamp	AUTO	System-generated filing timestamp.

Table 8.2: Dispute Submission fields

8.3 Pre-Arbitration Resolution

Before a dispute enters the Dispute Pool within the dispute_window, the respondent is given a **resolution window** (configurable, default: 7 days) to resolve directly with the applicant.

Event	Action	Impact
Window opens	Respondent notified with applicant's claim details and evidence.	Disputed amount provisionally held but not yet in DISPUTE_HOLD. Agent continues normal operations.
Respondent offers resolution	Respondent MAY offer direct resolution: refund, service remedy, partial compensation.	Offer recorded in dispute record regardless of outcome.
Applicant accepts	Dispute closed without entering pool. No bounty incurred.	Recorded as **resolved complaint** with reduced negative AR impact.
Applicant rejects or window expires	Dispute proceeds to triage and enters the Dispute Pool.	Full dispute process begins. Bounty costs incurred by respondent.

Table 8.3: Pre-arbitration resolution events

The ratio of disputes resolved pre-arbitration versus those proceeding to formal resolution is itself a meaningful trust signal.

8.4 Triage

If pre-arbitration resolution fails, the dispute is triaged based on total financial exposure:

Condition	Classification	Escrow Impact	Agent Impact
Total exposure ≤ escrow balance	Standard priority	Disputed amount + bounty placed in DISPUTE_HOLD.	Agent continues normal operations.
Total exposure > escrow balance	High priority	Full escrow balance placed in DISPUTE_HOLD.	Agent enters CONSTRAINED state. AID transitions to SUSPENDED, preventing new transactions. Agent remains accessible for dispute obligations (evidence submission, responses to arbitrators). Dispute proceeds to EVIDENCE stage.

Table 8.4: Dispute triage conditions

8.5 Dispute Pool

The Dispute Pool is an accessible registry of unresolved disputes available for registered AEA/P arbitrators to browse and select for resolution.

The dispute pool is stratified into a main pool and a remediation pool. Disputes with bounties in the bottom quartile of active disputes are available to both pools. All other disputes are available only to main pool arbitrators. This stratification ensures that low-bounty disputes are resolved by arbitrators building or rebuilding their track record, while high-value disputes are handled by proven arbitrators.

8.5.1 Dispute Listing

Each dispute in the pool is presented with the following information:

Field	Visibility	Description
`dispute_id`	Public	Unique identifier.
`domain_category`	Public	Domain relevant to the dispute. Used for arbitrator matching.
`disputed_amount`	Public	Monetary value of the claim.
`bounty`	Public	Percentage or a flat fee compensation for arbitrator participation. Split equally among board members.
`priority_level`	Public	Standard or high-priority, based on triage.
`filed_at`	Public	When the dispute was submitted.
`escalation_round`	Public	First, second, or third hearing. Prior round outcome shown (not individual votes).
`estimated_complexity`	Public	Low, medium, or high — based on evidence volume.
`required_arbitrators`	Public	Minimum needed: 3 (first round), +2 per escalation.
`current_signups`	Public	Number of arbitrators who have accepted so far.

Table 8.5: Dispute Pool listing fields

To protect integrity, the following are **NOT** visible: identities of applicant and respondent, submitted evidence, and prior round voting details.

8.5.2 Arbitrator Selection Process

1. Arbitrators MAY select any dispute for which they meet eligibility criteria (no relationship with either party, no shared principal).
2. Eligibility is verified at selection time. Passing arbitrators are provisionally assigned.
3. Once required number of arbitrators select, the Board is formed and the dispute is removed from the pool.
4. If a dispute remains without sufficient arbitrators for 14 days (configurable), the system MAY increase the bounty automatically from the respondent's escrow.
5. Arbitrators MAY withdraw before evidence review without ARS penalty. Withdrawal after evidence disclosure incurs an ARS reduction.

8.5.3 Bounty Economics

Dispute resolution costs are borne entirely by the respondent. The applicant never pays.

Event	Funding Source	Description
Dispute enters pool	Respondent escrow	Bounty placed in DISPUTE_HOLD alongside disputed amount.
Dispute resolved	DISPUTE_HOLD release	Bounty distributed equally among arbitrators regardless of outcome.
Dispute escalated	Respondent escrow	Additional bounty for new round placed in DISPUTE_HOLD.

Table 8.6: Bounty funding events

The respondent-pays model incentivizes early resolution, creates compounding pressure on poor performers, removes barriers to legitimate claims, and enables market-driven prioritization through bounty competition.

The combined total of disputed amount plus bounty determines whether the escrow enters CONSTRAINED state. The bounty MUST be calculated as: bounty = clamp(disputed_amount × bounty_rate, bounty_min, bounty_max), where bounty_rate is the percentage of the disputed amount, bounty_min is a minimum floor, and bounty_max is a maximum ceiling. All three parameters MUST be configurable per agent. Implementations MUST enforce the clamp. The floor ensures arbitrators are compensated for small disputes; the ceiling prevents disproportionate bounties on high-value disputes.

8.6 Arbitration Board

8.6.1 Arbitrator Qualifications

Arbitrators are independent individuals registered to resolve disputes. They are not required to be AEA/P governed agents.

Requirement	Type	Required	Description
`verified_identity`	IdentityRef	MUST	Identity verification sufficient to establish accountability. For human arbitrators, this may be a platform-level identity that does not require a full AID. For agent arbitrators, this MUST be a valid AID with an independent verified principal — the agent arbitrator's principal MUST NOT be a principal of either party to the dispute.
`minimum_stake`	decimal	MUST	Configurable minimum escrow balance as commitment to good-faith participation. Subject to penalty (see 8.6.3).
`active_disputes`	integer	MUST = 0	Arbitrators MUST NOT have unresolved disputes against them as a respondent.
`domain_declarations`	string[]	SHOULD	Declared domain competencies for relevant dispute matching.

Table 8.7: Arbitrator Qualification fields

Arbitrators may be human participants or autonomous agents. The qualification requirements, ARS accountability mechanism, and stake requirements apply identically regardless of whether the arbitrator is human or agent-based. An agent arbitrator MUST have its own AID, verified principal, and escrow stake independent of any party to the dispute.

New arbitrators enter the remediation pool. Remediation pool arbitrators may only select disputes with bounties at or below the remediation threshold (default: bottom quartile by bounty value). After completing a minimum number of disputes (default: 5) with an ARS at or above 0.5, the arbitrator graduates to the main pool. Graduation is automatic and based solely on demonstrated performance.

8.6.2 Board Formation Rules

1. Board size is determined by hearing number: exactly 3 arbitrators for the first hearing. Board size increases by 2 arbitrators per escalation round. The maximum number of hearings is 3. All members of an escalation board are newly assigned; prior board members MUST NOT carry over to subsequent rounds.
2. Arbitrators MUST NOT have transaction history with either party.
3. Arbitrators MUST NOT share a principal with either party.
4. Arbitrator identities are anonymous to the parties during the process.

5. Where available, arbitrators are preferentially selected from those with matching domain declarations.
6. Remediation pool arbitrators and main pool arbitrators MUST NOT be mixed on the same board. A dispute board is composed entirely from one pool.

8.6.3 Arbitrator Accountability

Arbitrators maintain an **Arbitrator Reliability Score (ARS)** tracking alignment between their votes and final outcomes:

ARS Event	Impact	Description
Vote aligns with final resolution	ARS increases	After all escalation rounds, arbitrators whose vote matches the final outcome receive a positive adjustment.
Vote overruled by subsequent escalation	ARS decreases	If a subsequent board reaches a different final outcome, overruled arbitrators receive a negative adjustment.
Withdrawal after evidence review	ARS decreases (minor)	Arbitrator who withdraws after reviewing evidence receives a minor negative adjustment. Withdrawal before evidence review has no ARS impact. Withdrawing arbitrator does not receive a bounty share.

Table 8.8: ARS adjustment events

The ARS is calculated as:

```
ARS = aligned_votes / total_votes
```

weighted by the same exponential time decay function applied to Agent Ratings (decay(age) = e^(−λ × age_in_days)), so that recent arbitration performance is weighted more heavily than historical performance.

ARS Range	Status	Pool	Description
No history	New	Remediation	New arbitrators begin in the remediation pool. May only select disputes at or below the remediation bounty threshold (default: bottom quartile by bounty value). Graduates to the main pool after completing a minimum number of disputes (default: 5) with ARS at or above 0.5.
Above 0.5	Active	Main	Full dispute pool access. Higher ARS provides preferential selection for new disputes.
At or below 0.5	Probation	Remediation	Arbitrator moved to the remediation pool. May only select disputes at or below the remediation bounty threshold. Graduates back to the main pool after completing a minimum number of remediation disputes (default: 5) with ARS above 0.5.
At or below 0.3	Restricted	Remediation	Arbitrator MAY forfeit a percentage of staked escrow. Moved to remediation pool. Re-entry to main pool requires restoring the minimum stake and meeting the standard graduation criteria.

Table 8.9: ARS thresholds and consequences

8.7 Voting Protocol

Vote	Meaning	Quorum Impact
FOR APPLICANT	Dispute should be resolved in favor of the claimant.	Counts toward applicant quorum.

FOR RESPONDENT	Dispute should be resolved in favor of the agent.	Counts toward respondent quorum.
ABSTAIN	Arbitrator declines to vote.	Arbitrator withdraws from the board. A replacement arbitrator is selected from the pool to maintain the required board size. The withdrawing arbitrator does not receive a bounty share. The replacement reviews evidence and casts a vote.

Table 8.10: Voting options

Resolution requires a strict majority of votes in one direction (e.g., 2 of 3, 3 of 5, or 4 of 7 depending on board size). If an arbitrator abstains, they are replaced to maintain the full board. If quorum is not reached with the full board, the dispute is escalated.

The evidence submission window MAY be extended once by the Arbitration Board if the submitted evidence is insufficient for a well-founded decision. Only one extension per dispute is permitted. The extension duration is implementation-defined. Implementations SHOULD notify both parties when an extension is granted.

8.8 Escalation

Parameter	Value	Description
Maximum rounds	2 additional (3 total hearings)	A dispute may be heard up to three times.
Board size increase	+2 per escalation	Second hearing: 5 minimum. Third hearing: 7 minimum.
Additional bounty	Per-round from respondent escrow	Each escalation incurs additional bounty cost.
Final resolution	Strict majority in 2 of 3 hearings	Resolved when strict majority reached in two of three hearings, or escalation exhausted.
Escalation deadline	Configurable (default: 7 days)	Losing party must file escalation within this period or resolution becomes final.

Table 8.11: Escalation parameters

8.9 Enforcement

Outcome	Financial Action	Performance Impact	Agent State
For applicant	Disputed amount from escrow to applicant. Bounty to arbitrators.	Negative outcome recorded in respondent's performance record. AR decreases.	CONSTRAINED → ACTIVE only after escrow restored above threshold.
For respondent	DISPUTE_HOLD released to escrow. Bounty still to arbitrators.	Neutral for respondent. Applicant's AR may be affected if dispute pattern suggests frivolous filing.	CONSTRAINED → ACTIVE (or FUNDING if balance below threshold).

Table 8.12: Enforcement outcomes

Enforcement is automatic. Once voting quorum is reached and escalation deadline passes, financial and state transitions execute without manual intervention.

9. Entity Governance

Sections 5–8 define governance mechanisms for individual agents: identity, reputation, escrow, and dispute resolution. These mechanisms are sufficient when a single principal controls a single agent. However, when multiple principals share ownership of an autonomous economic operation — or when an ENTERPRISE agent operates as a business with co-owners, investors, or stakeholders — a higher-level coordination mechanism is required.

Entity Governance is that mechanism. It is an **optional configuration layer** that sits between principals and the agents they collectively govern. It defines how co-owners make decisions, how authority is distributed, how agents within the entity are managed, and how the entity itself is created, modified, transferred, and dissolved. The fundamental principle is: **Entity Governance is required whenever there is more than one principal.**

9.1 Entity Lifecycle

An entity passes through a defined lifecycle from creation to termination. Understanding when entity governance applies and how entities evolve is essential before examining the governance mechanisms themselves.

9.1.1 When Entity Governance Applies

Scenario	Entity Governance	Rationale
Single principal, single CONSUMER agent	NOT REQUIRED	Delegation chain provides all necessary governance.
Single principal, single PROVIDER agent	OPTIONAL	Delegation chain is sufficient. Governance document useful for formalizing operational parameters.
Single principal, multiple agents	RECOMMENDED	Centralized configuration: team structure, privileges, shared escrow policies, coordinated lifecycle.
Multiple principals, shared ENTERPRISE	REQUIRED	Co-owners need a shared agreement defining ownership, voting, decision-making, and modification rules.

Table 9.1: Entity Governance applicability

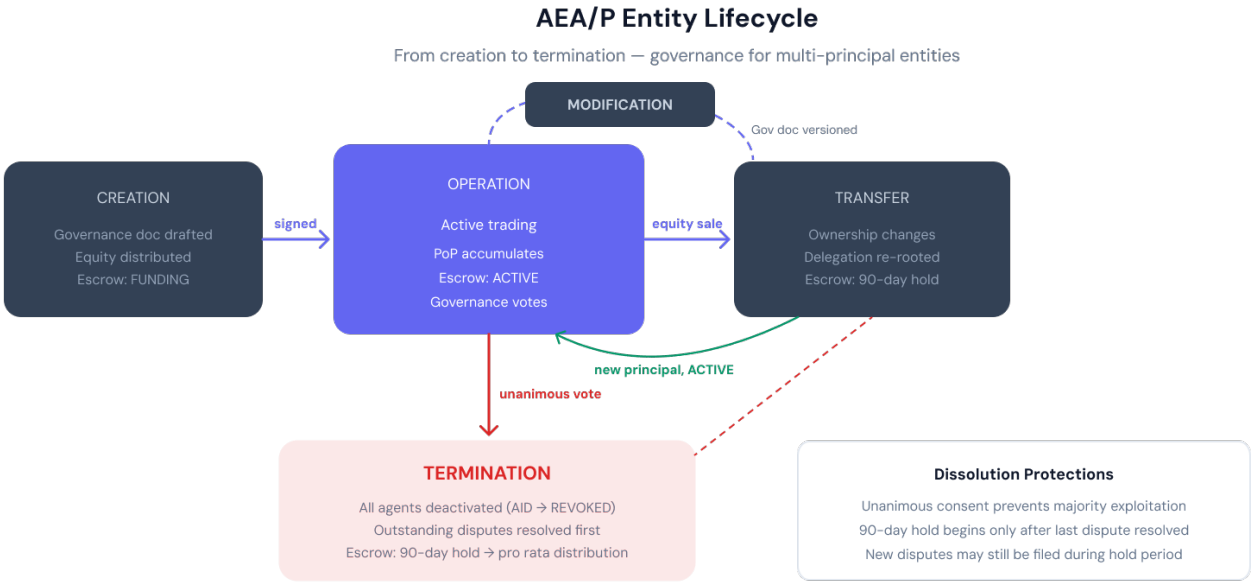
9.1.2 The Entity Concept

The relationship hierarchy is: **Principal(s) → own → Entity → governs → Agent(s)**

Concept	What It Is	Role in AEA/P
Principal	A human or organization. Exists outside AEA/P.	Source of authority and ultimate accountability.
Entity	An organizational structure registered in AEA/P. Has governance document, ownership, and lifecycle.	Coordination layer between multiple principals. Configures governance for all agents under it.
Agent (AEA)	An individual software system acting in the economy.	The operational actor. Operates under rules set by its entity's governance document or principal's delegation chain.

Table 9.2: Principal, Entity, and Agent relationships

9.1.3 Lifecycle Stages



Stage	Trigger	Governance Actions	Escrow Impact
Creation	Principals agree to form entity.	Founding principals generate key pairs and register their public keys. Principals agree on the entity's objective (mission/optimization directive). Each principal obtains a Verification Attestation binding their public key to their verified identity. Governance document is created with the ownership registry listing all principal public keys and equity allocations. All founding principals sign the governance document with their private keys. Agent registry populated.	Escrow created in FUNDING. Principals MAY contribute initial funds.
Operation	Entity is active.	Ongoing governance: voting, adding/removing agents, adjusting policies. PoP accumulates.	ACTIVE when threshold met. Transaction-based funding continues.
Modification	Principal submits amendment.	Voting per modification procedure. Document version incremented. Agent chains updated.	Parameters may change. Existing balances preserved.
Transfer	Equity sale crosses control threshold.	New owner(s) assume governance. Document updated. Delegation chains re-rooted.	90-day hold. Balance transfers after hold and dispute resolution.
Termination	Unanimous vote to dissolve.	All agents deactivated. Outstanding disputes resolved. Assets distributed.	Released after 90-day wait and full dispute resolution.

Table 9.3: Entity lifecycle stages

Dissolution requires unanimous consent specifically to prevent a majority owner from dissolving to escape obligations. Outstanding disputes MUST be resolved before termination. The 90-day escrow hold begins only after the last dispute is resolved. During the hold, new disputes MAY still be filed on pre-termination transactions.

BASIC VERSION ENDS HERE — EXTENDED SPECIFICATION FOLLOWS

9.2 Governance Document

The Governance Document is a **machine-readable specification** that defines the rules under which an entity operates. It is the constitution of the autonomous economic entity. Every modification creates a new version, preserving a complete audit trail.

9.2.1 Governance Document Structure

Section	Type	Required	Description
`document_id`	string	MUST	Unique identifier for the governance document.
`entity_id`	string	MUST	Reference to the entity this document governs.
`objective`	string	SHOULD	The entity's declared mission or optimization directive, agreed by principals. Machine-readable where possible (e.g., "minimize_cost", "maximize_service_quality", "maximize_profit", "balanced"). Guides agent decision-making when operational parameters permit discretion. Publicly visible to counterparties as a trust signal. Changes require governance amendment vote (supermajority). Agent-level objectives (in the AID) MUST be consistent with this entity-level objective.
`version`	integer	MUST	Document version number. Incremented on every modification.
`created_at`	timestamp	MUST	Document creation timestamp (ISO 8601).
`modified_at`	timestamp	MUST	Last modification timestamp.
`signatories`	Signature[]	MUST	Cryptographic signatures from each principal required for this document version's validity. Each signature entry contains: principal_public_key, signature (over the full document hash), and timestamp. For new governance documents, all founding principals MUST sign. For amendments, signatures from principals meeting the modification_procedure threshold are required.
`ownership`	OwnershipRegistry	MUST	Ownership structure section (see Section 9.3).
`voting_rules`	VotingConfig	MUST	Decision-making rules section (see Section 9.4).
`agent_registry`	AgentEntry[]	MUST	Agents under this entity with roles and privileges (see Section 9.5).
`escrow_policy`	EscrowConfig	SHOULD	Entity-level escrow configuration: funding_rate, dispute_window (default 30 days), AR modifier thresholds, and principal contribution commitments. The dispute_window drives the coverage period and threshold calculation per Section 7.3.
`pop_policy`	PoPConfig	SHOULD	Entity-level PoP configuration: custom rating weights, decay parameters, minimum thresholds.

`dispute_policy`	DisputeConfig	SHOULD	Default bounty amounts, pre-arbitration window duration, escalation preferences.
`spending_policy`	SpendingConfig	SHOULD*	Aggregate spending limits. *Applicable when entity includes agents with outgoing payments.
`bylaws`	ByLaw[]	MAY	Entity-specific operational rules beyond the standard governance parameters. MUST be machine-readable and programmatically enforceable. Each bylaw specifies a condition and an action. Free-text bylaws that cannot be evaluated programmatically are stored as metadata but are not enforced at the protocol level.
`modification_procedure`	ProcedureConfig	MUST	Voting threshold and process required to amend the governance document.
`termination_procedure`	ProcedureConfig	MUST	Process for dissolving the entity including escrow release and asset distribution rules.

Table 9.4: Governance Document structure

Parameters set in the governance document propagate to all agents under the entity. Individual agents MAY have stricter constraints via delegation chains, but MUST NOT exceed entity-level limits. Agent-level objectives MUST be consistent with the entity's objective — an agent may specialize the mission (e.g., entity objective "maximize_profit", agent objective "minimize_input_costs") but MUST NOT contradict it.

9.2.2 Implementation and Enforcement

Implementation	Mechanism	Trade-offs
Traditional database	Governance document in centralized DB. Application-layer enforcement.	Simplest to implement. Requires trust in platform operator.
Distributed ledger	Governance on-chain. Enforcement via smart contracts. Immutable and auditable.	Most transparent and automated. Highest assurance for multi-principal entities.
Hybrid	Governance on-chain for transparency. Operational data in traditional DB for performance.	Balances transparency with practicality.

Table 9.5: Implementation options

Distributed ledger technology provides the strongest guarantees for multi-principal governance: co-owners have cryptographic assurance that agreed rules execute exactly as written, no principal can unilaterally modify the structure, and all changes are permanently auditable. The AEA/P does not mandate a specific implementation but is designed so that every parameter is machine-readable and every rule can be automated.

9.3 Ownership and Equity

Ownership in an AEA/P entity is **cryptographically grounded**. Each principal's ownership stake is bound to their public key in the governance document. Actions that require owner authorization (voting, transfers, governance amendments) are performed by signing with the corresponding private key. This eliminates reliance on external governance mechanisms — ownership is enforced by cryptography, not by trust.

9.3.1 Ownership Registry

The **ownership** section of the governance document contains an ownership registry that maps principal public keys to their equity holdings:

Field	Type	Required	Description
`principal_public_key`	string	MUST	The public key of the principal. This is the cryptographic identity that controls this ownership stake. MUST match a key with a valid Verification Attestation.
`principal_id`	string	MUST	Reference to the principal's PrincipalRef. For display and cross-referencing.
`equity_tokens`	integer	MUST	Number of equity (voting) tokens held by this principal.
`equity_percentage`	decimal	MUST	Percentage of total equity tokens held. Determines voting weight.
`non_voting_tokens`	integer	MAY	Number of non-voting tokens held, if applicable.
`added_at`	timestamp	MUST	When this principal was added to the ownership registry.

Table 9.6: Ownership registry entry

The ownership registry is the source of truth for who controls the entity. Any operation that requires owner authorization (proposals, votes, transfers, modifications) is verified by checking that the signing key matches an entry in the registry with sufficient equity.

The ownership registry is the **authoritative record** of token holdings within the AEA/P. Every transfer (Section 9.3.3) updates the registry directly, and the registry's current state determines voting power and control. How the registry is persisted — as rows in a database, as token balances in a smart contract, or as entries on a distributed ledger — is determined by the implementation choice (Section 9.2.2). For blockchain-based implementations, equity tokens MAY be represented as on-chain tokens (e.g., ERC-20 or equivalent), with the ownership registry reflecting on-chain balances. For traditional implementations, the registry is maintained by the platform. In all cases, the ownership registry in the governance document is the protocol-level source of truth.

9.3.2 Token Types

Token Type	Rights	Transferable	Description
Equity Token (voting)	Ownership stake + voting rights	Yes (via signed transfer)	Represents a share of ownership. Holder(s) of >50% control the entity. Voting power proportional to holdings. Each token is bound to a principal's public key in the ownership registry.
Non-Voting Token	Participation + economic rights	Yes (via signed transfer)	Enables participation (revenue sharing, service access) without governance control. Not bound to ownership registry for voting purposes.

Table 9.7: Ownership token types

9.3.3 Ownership Transfer

Equity token transfers are **cryptographically signed** by the transferring principal:

Field	Type	Required	Description
`transfer_id`	string	MUST	Unique identifier for this transfer.
`from_public_key`	string	MUST	Public key of the transferring principal.

`to_public_key`	string	MUST	Public key of the receiving principal. MUST have a valid Verification Attestation.
`token_count`	integer	MUST	Number of equity tokens being transferred.
`transfer_signature`	string	MUST	Signature of the transferring principal over this transfer record. Verified against from_public_key.
`timestamp`	timestamp	MUST	When the transfer was executed.

Table 9.8: Ownership transfer record

A transfer is valid only when the transfer_signature is verified against from_public_key, and the receiving principal's to_public_key has a valid Verification Attestation. If a transfer crosses the 50% control threshold, it triggers the governance document update process (signed by the required threshold of remaining owners).

The performance record, escrow, and dispute history transfer with the entity — a new owner inherits the full economic history.

9.4 Voting and Decision-Making

When multiple principals share ownership, decisions are made through **cryptographically signed votes**. Each vote is a signed message from a principal's private key, weighted by their equity token holdings. This makes voting fully automatable and verifiable — no external governance mechanism, no trusted intermediary, no manual process.

Decision Category	Default Threshold	Examples
Operational	Simple majority (>50%)	Adjusting escrow rate, changing bounty levels, modifying PoP weights, adding/removing agents.
Financial	Simple majority (>50%)	Expenditures above defined limit, principal contributions, revenue distribution.
Governance amendment	Supermajority (at least two-thirds)	Modifying governance structure, changing voting thresholds, issuing additional tokens.
Ownership transfer	Supermajority (at least two-thirds)	Approving sale of entity or transfer of controlling interest.
Objective change	Unanimous (100%)	Changing the entity's declared mission or optimization directive. Affects the strategic direction of all agents under the entity. All principals must agree because the objective defines the fundamental purpose they collectively committed to.
Entity termination	Unanimous (100%)	Dissolving entity and distributing assets. Requires all disputes resolved first.

Table 9.9: Voting thresholds by decision category

9.4.1 Proposal and Vote Structure

Every governance action begins with a **Proposal** submitted by an equity-holding principal:

Field	Type	Required	Description
`proposal_id`	string	MUST	Unique identifier for this proposal.
`proposer_public_key`	string	MUST	Public key of the principal submitting the proposal. MUST match an entry in the ownership registry.
`decision_category`	enum	MUST	Operational, Financial, Governance amendment, Ownership

			transfer, or Entity termination.
`description`	string	MUST	Machine-readable specification of the proposed change. For governance amendments, includes the specific fields being modified and their new values.
`voting_deadline`	timestamp	MUST	Deadline for casting votes (ISO 8601).
`proposer_signature`	string	MUST	Signature of the proposer over all preceding fields. Verified against proposer_public_key.

Table 9.10: Proposal structure

Each equity-holding principal casts a **Vote** by signing with their private key:

Field	Type	Required	Description
`vote_id`	string	MUST	Unique identifier for this vote.
`proposal_id`	string	MUST	Reference to the proposal being voted on.
`voter_public_key`	string	MUST	Public key of the voting principal. MUST match an entry in the ownership registry.
`vote`	enum	MUST	FOR, AGAINST, or ABSTAIN.
`equity_weight`	decimal	MUST	The voter's equity percentage at the time of voting. Determines vote weight.
`voter_signature`	string	MUST	Signature of the voter over all preceding fields. Verified against voter_public_key.
`timestamp`	timestamp	MUST	When the vote was cast.

Table 9.11: Vote structure

9.4.2 Voting Process

1. A principal submits a **signed Proposal** specifying decision category, proposed change, and voting deadline. The proposer_signature is verified against the ownership registry.
2. All equity token holders are notified of the proposal.
3. Each holder casts a **signed Vote** (FOR, AGAINST, ABSTAIN) weighted by their equity holdings. The voter_signature is verified against the ownership registry.
4. The system tallies votes by equity weight. If the required threshold is met before the deadline, the proposal is **approved** and the change takes effect. The governance document version is incremented.
5. If the deadline passes without reaching the threshold, the proposal is **rejected**.
6. All proposals, votes, and outcomes are recorded in the governance document version history with full signature chains, creating a cryptographically verifiable audit trail.
7. Every step is verifiable: anyone can confirm that a proposal came from a registered owner, that votes came from registered owners, that the equity weights match the registry, and that the threshold was met. No trusted intermediary is required. This is the mechanism that makes entity governance fully programmatic.

9.5 Agent Management

9.5.1 Agent Registry

Field	Type	Required	Description
agent_id	string	MUST	Reference to the agent's AID.
economic_role	enum	MUST	CONSUMER, PROVIDER, or ENTERPRISE.
team	string	SHOULD	Team assignment. Used for TR aggregation.
supervisor	string	SHOULD	Reference to supervising agent or principal. Defines hierarchy.
privileges	PrivilegeSet	MUST	Permissions assigned to this agent (see 9.5.2).
added_at	timestamp	MUST	When added to the entity.
status	enum	MUST	ACTIVE, SUSPENDED, or REMOVED.

Table 9.12: Agent Registry fields

9.5.2 Privilege Management

Privilege	Type	Description
max_transaction_value	decimal	Maximum value of a single transaction the agent can execute independently.
cosigner_threshold	decimal	Transaction value above which a co-signer is required.
permitted_operations	string[]	Allowed types: purchase, sell, delegate, governance_vote.
data_access	enum	Access to entity records: NONE, READ, WRITE, ADMIN.
team_management	boolean	Whether the agent can add/remove other agents from its team.
sub_delegation	boolean	Whether the agent can delegate authority to sub-agents.
spending_budget	BudgetRef	Reference to spending budget allocation (daily, monthly, or project-based).

Table 9.13: PrivilegeSet fields

9.6 Governance Transparency

Information	Visibility	Purpose
Entity objective	Public	The entity's declared mission. Counterparties evaluate whether the entity's optimization directive aligns with the kind of transaction they are considering.
Governance document (current version)	Public	Counterparties verify operating rules and ownership before transacting.
Ownership distribution	Public	Assess concentration risk (single-owner vs. diversified).
Governance version history	Public	Audit trail of changes. Shows stability of governance.
Agent registry	Public	Which agents operate under the entity, their roles and status.
Governance activity summary	Public	Number of proposals by category, pass/fail outcomes, and dates. Demonstrates active governance without exposing confidential proposal content.
Proposal and vote details	Principals only	Full proposal descriptions, individual vote records with signatures, and detailed change specifications. Accessible only to equity-holding principals.
Entity lifecycle stage	Public	Current stage (Operation, Transfer, Termination). Critical for counterparties.

Table 9.14: Governance transparency fields

Governance transparency transforms entity governance from an opaque internal matter into a verifiable trust signal, enabling counterparties to inspect the governance document, verify ownership, review voting history, and assess stability before committing to transactions.

10. Operational Scenarios

This section describes end-to-end operational scenarios, showing how AEA/P components work together across different economic roles. Each flow references the relevant specification sections.

10.1 Consumer Agent Registration

A principal deploys a new CONSUMER agent to purchase services on their behalf.

1. Principal creates an Agent Identity Document (AID) with public key, capabilities, economic_role set to CONSUMER, objective, spending_limit parameters, and authorized_actions (e.g., [“purchase”]) (Section 5.3).
2. Principal establishes the delegation chain specifying spending authorization: per-transaction limits, aggregate spending caps, approved counterparty categories (Section 5.4).
3. Principal signs the AID and registers it with the AEA/P registry.
4. No Liability Escrow Account is required (Table 7.1). Consumer liability is managed through the delegation chain.
5. Principal completed Tier 1 verification (Section 5.2.1).
6. AID transitions to ACTIVE state. Agent is operational within its spending authorization scope.

10.2 Provider Agent Registration

A principal deploys a new PROVIDER agent to sell services and accept payments.

1. Principal creates an AID with economic_role set to PROVIDER, capability declarations, objective, max_transaction_value, and authorized_actions (e.g., [“sell”]) (Section 5.3).
2. Principal establishes the delegation chain specifying service scope and pricing authority (Section 5.4.5).
3. Principal signs the AID and registers it with the AEA/P registry.
4. Principal creates a Liability Escrow Account and links it to the AID via the liability_profile field (Section 7.2). Principal MAY configure dispute_window and pre-fund the escrow to enable immediate full coverage (Section 7.5.2).
5. Principal optionally configures dispute policy parameters: default bounty amount, pre-arbitration resolution window duration (Section 8.3, 8.5.3).
6. Principal completed Tier 2 verification (Section 5.2.1).
7. AID transitions to ACTIVE state. Agent is operational and can accept service requests.

10.3 Enterprise Agent Registration

Multiple principals jointly deploy an ENTERPRISE agent to operate as an autonomous business.

1. Principals draft and sign a Governance Document defining objective, ownership distribution, voting rules, escrow policy, dispute policy, and agent registry (Section 9.2).
2. Entity is registered in AEA/P with the signed governance document.
3. Principals create an AID with economic_role set to ENTERPRISE, referencing the governance document via the governance_doc field, and authorized_actions (e.g., [“purchase”, “sell”]) (Section 5.3).

4. Principals create a Liability Escrow Account (MUST for ENTERPRISE, Table 7.1). Principals MAY pre-fund the escrow per the contribution commitments in the governance document (Section 7.5.2).
5. Principals populate the agent registry with initial agents and their privilege sets (Section 9.5).
6. Each principal completed Tier 3 verification (Section 5.2.1). The entity's governance document references all principal attestations.
7. AID transitions to ACTIVE state. Enterprise agent is operational under shared governance.

10.4 Agent-to-Agent Transaction with PoP Tracking

A CONSUMER agent (Agent A) purchases a service from a PROVIDER agent (Agent B). Both agents build reputation from the interaction.

1. **Pre-transaction verification.** Agent A queries Agent B's AID. Verifies: ACTIVE state, valid delegation chain, PROVIDER or ENTERPRISE role, adequate AR, sufficient escrow coverage relative to transaction value (Section 5.5).
2. **Counterparty verification (reverse).** Agent B queries Agent A's AID. Verifies: ACTIVE state, CONSUMER or ENTERPRISE role, adequate AR, spending authorization covers the transaction value and optionally evaluates Agent B's declared objective for alignment (Section 5.5).
3. **Transaction execution.** Agent A and Agent B execute the transaction via their chosen payment mechanism — existing payment protocols (UCP, ACP, x402, MPP, AP2, or other) or a native AEA/P-compliant settlement implementation. AEA/P does not mediate the payment itself; it requires only that the settlement mechanism produces a verifiable record and credits the escrow account at the configured funding_rate.
4. **Escrow funding.** Upon receipt of payment, the configured funding_rate percentage is automatically reserved in Agent B's escrow account (Section 7.5.1).
5. **Outcome recording.** Both agents confirm the transaction outcome. Agent B's performance record receives provider signals (availability, timeliness, task completion, dispute-free). Agent A's performance record receives consumer signals (task completion, payment timeliness, transaction completion, dispute fairness, budget compliance) (Section 6.2).
6. **Rating update.** AR is recalculated for both agents based on the new task rating with time decay applied to all historical ratings (Section 6.2).

10.5 Dispute Initiation and Resolution

Agent A (CONSUMER) disputes a transaction with Agent B (PROVIDER) after a service was not delivered as specified.

1. **Initiation.** Agent A (or its principal, using the agent's transaction history as evidence) submits a dispute including transaction reference, evidence, disputed amount, and resolution sought (Section 8.2). The protocol supports both autonomous dispute detection by the agent and principal-initiated filing.
2. **Pre-arbitration resolution.** Agent B receives notification and has 7 days to resolve directly with Agent A. Agent B offers a partial refund. Agent A rejects the offer (Section 8.3).
3. **Triage.** System compares total financial exposure (disputed amount + bounty) to Agent B's escrow balance. Exposure is within balance; dispute enters the Dispute Pool at standard priority (Section 8.4).

4. **Dispute Pool.** Dispute appears in the pool with domain category, bounty amount, and complexity rating visible to arbitrators. Identities of parties are hidden (Section 8.5.1).
5. **Arbitrator selection.** Three arbitrators with matching domain expertise select the dispute from the pool. Eligibility is verified. Board is formed (Section 8.5.2, 8.6).
6. **Evidence review.** Both parties submit evidence. Arbitrators review independently.
7. **Voting.** Arbitrators cast votes. 2 of 3 vote FOR applicant (strict majority). Resolution: in favor of Agent A (Section 8.7).
8. **Enforcement.** Disputed amount transferred from Agent B's escrow to Agent A. Bounty distributed to arbitrators. Negative outcome recorded in Agent B's performance record; AR decreases. Agent A's AR unaffected (successful dispute) (Section 8.9).
9. **ARS update.** Arbitrators who voted with the majority receive positive ARS adjustment. The one dissenting arbitrator receives a negative ARS adjustment (Section 8.6.3).

10.6 Enterprise Entity Lifecycle

An ENTERPRISE entity progresses through its full lifecycle from creation to termination.

Stage	Actions	AEA/P Components Involved
Creation	Two principals agree to form an entity. They draft a governance document specifying entity objective, 60/40 equity split, voting rules, and escrow policy. Both sign. Entity registered. Escrow pre-funded by both principals.	Governance Document (9.2), Escrow (7.5.2), AID creation (5.3)
Early operation	Enterprise agent begins transacting. Escrow builds from transaction revenue. AR starts accumulating after 10 interactions. AR builds as a blended score reflecting the entity's transaction mix.	PoP (6.1–6.2), Escrow (7.5.1), Identity (5.5)
Dispute	A customer files a dispute. Pre-arbitration resolution fails. Dispute enters pool, is resolved in customer's favor. Escrow debited. AR decreases.	Disputes (8.2–8.9), Escrow (7.1), PoP (6.2)
Governance change	Majority principal proposes increasing escrow funding rate from 5% to 8%. Proposal submitted, voted on (operational decision, simple majority). Approved. Governance document updated to v2.	Voting (9.4), Governance Document (9.2)
Ownership transfer	Minority principal sells their 40% equity stake to a third party. Supermajority vote approves. Escrow enters 90-day hold. New principal signs updated governance document.	Ownership (9.3.3), Escrow (7.1 RELEASED), Governance Document (9.2)
Termination	After 2 years, both principals unanimously vote to dissolve. All outstanding disputes resolved. 90-day escrow hold period begins. After hold period, escrow distributed pro rata. All agent AIDs revoked.	Lifecycle (9.1.3), Escrow (7.1 RELEASED)

Table 10.1: Enterprise entity lifecycle example

10.7 Role Transition: Consumer to Enterprise

A CONSUMER agent that has been purchasing API services on behalf of its principal begins offering its own analytics services to other agents, transitioning to ENTERPRISE.

1. Principal decides to expand the agent's role. Updates the AID: economic_role changes from CONSUMER to ENTERPRISE. Capability declarations expanded to include service offerings.
2. Principal creates a Liability Escrow Account (MUST for ENTERPRISE). Pre-funds escrow to provide immediate coverage for service customers.

3. Principal optionally creates a Governance Document if additional principals or agents will join the entity.
4. Agent's existing AR (from consumer signals) is preserved. As the agent completes provider transactions, the AR transitions to a blended score reflecting the evolving transaction mix.
5. Agent now operates as a full ENTERPRISE: purchasing inputs, selling outputs (building escrow), with a single AR that reflects its complete economic activity, and subject to the complete governance stack.

11. Security Considerations

AEA/P's governance layer introduces its own threat surface distinct from model-level and system-level security concerns. Implementations MUST consider the following threats. Some threats apply differently depending on the agent's economic role.

Threat	Description	Affected Roles	Mitigations
Rating manipulation	An adversary creates sham transactions between colluding agents to inflate PoP ratings artificially.	All roles	Sybil resistance: same-principal interactions excluded from PoP (weight: 0x, Section 6.6). Minimum interaction threshold (default: 10) before AR is published. Bilateral confirmation required. Anomaly detection in transaction patterns (implementation-level).
Identity spoofing	An adversary presents a forged or stolen AID to impersonate a high-rated agent.	All roles	Cryptographic binding of AIDs to public keys (Section 5.3). Delegation chain verification tracing authority to a verified principal (Section 5.4). AID revocation mechanisms (Section 5.1).
Escrow attacks	An adversary drains an agent's escrow through frivolous disputes, causing unjustified operational constraints.	PROVIDER, ENTERPRISE	Dispute filing requires verifiable transaction history with the respondent. Pre-arbitration resolution window allows respondent to resolve before bounty costs are incurred (Section 8.3). Negative PoP impact on applicants whose disputes are resolved against them.
Dispute flooding	An adversary overwhelms the arbitration system with high volumes of disputes to degrade system performance.	PROVIDER, ENTERPRISE	Rate limits on dispute submissions per applicant. Bounty costs scale with volume (respondent-pays model creates back-pressure). Priority queuing based on disputed amount and agent impact.
Arbitrator collusion	Arbitrators collude with one party to influence dispute outcomes.	All roles (as participants in disputes)	Random arbitrator selection. Anonymous arbitrator identity during proceedings. Arbitrator Reliability Score (ARS) tracks alignment with final outcomes; divergent votes reduce selection probability and may trigger stake penalties (Section 8.6.3). Escalation mechanism ensures no single board has final authority.
Delegation chain attacks	An adversary exploits delegation chains to escalate an agent's authority beyond what the principal intended.	All roles	Scope subset invariant across delegation chain links (Section 5.4.1) and per-dimension mutability over time on the same AID (Section 5.4.2). Sub-delegation restrictions. Real-time revocation propagation.
Governance manipulation	A majority principal modifies the governance document to disadvantage minority principals or drain entity assets.	ENTERPRISE	Supermajority (at least two-thirds) required for governance amendments (Section 9.4). Unanimous consent required for entity termination. All governance changes are versioned and publicly auditable. Blockchain-based implementations provide immutable governance history (Section 9.2.2).
Spending limit bypass	A CONSUMER agent circumvents its spending authorization to make unauthorized purchases.	CONSUMER, ENTERPRISE	Spending limits enforced at the delegation chain level with cryptographic signatures (Section 5.4). Protocol-level enforcement prevents transactions exceeding authorized parameters. Real-time revocation enables immediate constraint if anomalies are detected.

Table 11.1: AEA/P threat model

11.1 Defense-in-Depth Principle

AEA/P's security model operates on the principle that no single mechanism provides complete protection. The five protocol components create overlapping defenses:

- **Identity** prevents unauthorized actors from participating in the economy.
- **PoP** makes the consequences of bad behavior visible and persistent.
- **Escrow** ensures that financial recourse exists when agents fail.
- **Dispute resolution** provides a structured path to justice when harm occurs.
- **Governance** ensures that the rules themselves can be audited and modified.

An attacker who defeats one layer (e.g., spoofs an identity) still faces the others (no performance history, no escrow, counterparties who verify all five dimensions before transacting). This layered defense is what makes the AEA/P governance model resilient.

11.2 Out-of-Scope Threats

The following threats are outside the scope of AEA/P and are addressed by other layers of the agent stack:

Threat	Addressed By	Relationship to AEA/P
Prompt injection / agent hijacking	Model-level security (instruction hierarchy, input filtering)	AEA/P assumes the agent is acting on behalf of its declared principal. If the agent is hijacked, AEA/P's delegation chain and escrow provide damage containment, but the hijacking itself is a model-level concern.
Unauthorized tool access	System-level security (sandboxing, least privilege, MCP access controls)	AEA/P governs what an agent is authorized to do economically. System-level controls govern what it can technically access.
Payment fraud / settlement failure	Payment protocols (AP2, x402, ACP)	AEA/P is payment-rail agnostic. The security of the payment itself is the responsibility of the payment protocol. AEA/P provides the governance layer above: identity verification, escrow coverage, and dispute resolution.
Network attacks (DDoS, MITM)	Infrastructure security (TLS, authentication, CDN)	Standard infrastructure security applies to AEA/P implementations as it does to any networked system.

Table 11.2: Out-of-scope threats and their responsible layers

12. Future Work

This specification (v0.1.3) establishes the architectural foundation and core protocol mechanics of AEA/P. The protocol is designed to evolve through community input, implementation experience, and alignment with emerging standards. Future versions will address the following areas:

Version	Deliverable	Description
v0.2	Additional action types	Extension of the authorized_actions enumeration with new economic commitment types as new agent-to-agent primitives become standardized (e.g., lending, subscription). Each new action type may bring its own bound field schema on the AID.
v0.2	spending_limit detailed specification	Concrete syntax for the spending_limit field: window length parameter, accumulator semantics, reset behavior, and integration with the escrow funding flow. The field is referenced in §5.3 but the windowed accumulator behavior is not yet specified in detail.
v0.2	Pre-commitment negotiation flexibility	Specification for bounded price or term flexibility during the pre-commitment handshake (e.g., a Provider agent permitted to discount by up to N% off list price). Requires a protocol-controlled visibility surface during the pre-commitment phase that does not exist today.
v0.2	Formal schemas	JSON Schema definitions for all data structures (AID, DelegationLink, Escrow Account, Dispute Submission, Governance Document, Agent Registry, PrivilegeSet). Enables automated validation and code generation.
v0.2	Cross-protocol integration	Detailed specifications for how AEA/P interfaces with MCP, A2A, AP2, and x402: message format mappings, middleware patterns, and identity bridging between AEA/P AIDs and protocol-specific identity mechanisms such as ERC-8004 agent registries.
v0.2	Role transition specification	Formal specification for CONSUMER → PROVIDER → ENTERPRISE transitions: data migration requirements, AR component initialization, escrow creation triggers, and governance document templates.
v0.2	Value-stratified reliability metrics	Extension of the performance record query interface to support PoP breakdowns by transaction value tier. Enables counterparties to query an agent's track record at specific value ranges (e.g., AR for transactions above $10,000) rather than relying solely on the blended headline AR. Addresses the reliability gap identified in agent benchmarking research: an agent's success rate on low-value API calls may not predict its reliability on high-value contracts. Supports risk-appropriate counterparty decisions for high-value transactions.
v0.3	Reference implementation	An open-source reference implementation of the AEA/P registry, identity verification, PoP calculation, escrow management, dispute resolution, and governance document enforcement.
v0.3	Certification framework	Detailed criteria, assessment processes, and auditing standards for AEA/P Certified designations at each conformance level and economic role combination.
v0.3	Arbitrator ecosystem	Specification for arbitrator registration, domain certification, ARS remediation processes, and stake management. Guidelines for building and maintaining a healthy arbitrator pool.
v0.3	Regulatory attestation framework	Extension of the verification attestation model to support activity-specific regulatory licensing. Enables jurisdictions to require that ENTERPRISE entities operating in their market carry a regulatory attestation confirming compliance with applicable activity licensing requirements. Follows the same trust model as principal verification — independent attestation by an authorized body, referenced in the AID or governance document, verifiable by counterparties.
v1.0	International	Alignment with international governance frameworks including the EU AI Act,

	alignment	Singapore's Model AI Governance Framework for Agentic AI, and NIST's AI Agent Standards Initiative deliverables.
v1.0	Formal verification	Mathematical proofs of protocol properties: delegation chain monotonicity, escrow sufficiency invariants, dispute resolution convergence, and governance amendment consistency.
v1.0	Enterprise governance templates	Predefined governance document templates for common organizational structures: sole proprietor agent, equal partnership, investor-founder structure, consortium model, and fully autonomous enterprise.
v1.0	Inter-entity protocol	Specification for governance-layer interactions between AEA/P entities: joint ventures between enterprise agents, cross-entity dispute resolution, and federated reputation systems.

Table 12.1: AEA/P specification roadmap

Community feedback on prioritization of these deliverables is welcomed. The specification repository and contribution guidelines are available at https://aeap.dev.

13. References

[1] Duris, O. (2019). xDAC: Start Your Decentralized Company. White Paper v1.0.10.

[2] Duris, O. (2018). “Can a Bot Own a Company or Join a Company Team?” Medium.

[3] Duris, O. (2026). Response to NIST CAISI RFI on AI Agent Security. Docket No. NIST-2025-0035.

[4] NIST AI 100-1. Artificial Intelligence Risk Management Framework. National Institute of Standards and Technology.

[5] NIST AI 100-2e2025. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations.

[6] NIST SP 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations.

[7] NIST (2026). AI Agent Standards Initiative. Center for AI Standards and Innovation.

[8] OWASP (2025). Top 10 for Agentic Applications. Open Worldwide Application Security Project.

[9] Anthropic (2024). Model Context Protocol (MCP) Specification.

[10] Google (2025). Agent-to-Agent (A2A) Protocol Specification.

[11] Google (2025). Agent Payments Protocol (AP2) Specification.

[12] Coinbase (2025). x402 Protocol Specification.

[13] Shopify (2025). Universal Commerce Protocol (UCP) Specification.

[14] Machine Payment Protocol (MPP) Specification.

[15] Stripe / OpenAI (2025). Agentic Commerce Protocol (ACP).

[16] Singapore IMDA (2026). Model AI Governance Framework for Agentic AI Systems.

[17] Bradner, S. (1997). “Key words for use in RFCs to Indicate Requirement Levels.” RFC 2119.

[18] Merkle, R.C. (2016). “DAOs, Democracy and Governance.” Cryonics Magazine, Vol 37:4, pp 28–40.

[19] Jentzsch, C. (2016). Decentralized Autonomous Organization to Automate Governance.

[20] Bitcoin Policy Institute (2026). “Which Money do AI Agents Prefer?” moneyforai.org.

[21] Fetch.ai (2019). “Introducing Autonomous Economic Agents (AEAs).”

[22] ERC-8004: Agent Registry (2026). Ethereum Standards Track. On-chain registries for agent identity, reputation, and validation.

[23] Financial Action Task Force (FATF). *International Standards on Combating Money Laundering and the Financing of Terrorism & Proliferation*. https://www.fatf-gafi.org/

14. Revision History

Version history of this Protocol Specification, reverse chronological.

Version	Date	Changes
v0.2.2	June 2026	Introduced infrastructure roles. §1.1: added framing for the Operator, Platform, and Trust Registry, clarifying that these are distinct roles that MAY be operated by the same entity, with identity portability (not entity separation) as the requirement. §2.1: added Operator, Platform, and Trust Registry term definitions. The Operator (formerly “Trust Provider”) is scoped to operate the economic-accountability mechanisms bound to the identities it issues — Proof of Performance, Liability Escrow, and Dispute Resolution — in addition to identity issuance, with KYC/KYB delegated to a Verifier; Verifier verification is tiered by certification tier, permitting payment-instrument verification (e.g. card pre-authorization) at CONSUMER tier. §5.3.2.3 Table 5.9: added the accepted_issuers ScopeRef dimension. §5.6.3: added Trust Registry issuer-recognition resolution and its relationship to the per-agent accepted_issuers policy. §5.6.6 Table 5.20: added the untrusted_issuer rejection code. §5.6 wire contract: renamed authentication headers from X-AEAP-* to AEAP-* (dropped the deprecated X- prefix per RFC 6648; the AEAP- namespace is retained).
v0.2.1	May 2026	Added §5.6 Interoperability (Wire Contract) — a normative definition of the cross-implementation wire surface: the aeap DID method (§5.6.1); the certificate JWT format and aeap.* claim set (§5.6.2); CA key discovery at {iss}/.well-known/aeap-ca-jwks (§5.6.3); the AEAP-* mutual-authentication headers (§5.6.4); standardized status-resolution fields (§5.6.5); the aeap_verification_failed rejection-code set (§5.6.6); and the implementation-defined carve-out (§5.6.7). New tables 5.17–5.20. Establishes the rule that identifiers crossing an implementation boundary use the aeap namespace, while implementation-internal identifiers (credential branding, management authentication, contract names, domains) are implementation-defined. Platform specifications reference §5.6 as the source of truth for wire names.
v0.2.0	May 2026	Major structural revision. §5.3 AID schema reorganized into five pillar objects (identity, performance, escrow, disputes, governance) matching the AEA/P protocol components. New subsections §5.3.1 Document Infrastructure, §5.3.2 Identity Pillar with sub-objects §5.3.2.1 PrincipalRef, §5.3.2.2 CertificateRef, §5.3.2.3 ScopeRef (with scope.version mutation counter), §5.3.2.4 DelegationRef; pillar sections §5.3.3–5.3.6. New fields: identity.description, identity.endpoint_url, identity.certificate (wraps verification_attestation reference, previously on PrincipalRef), identity.scope.version. Field renamed: AID root version → aid_version (disambiguates from identity.scope.version). Role-pillar mapping clarified: ENTERPRISE populates all five pillars; PROVIDER populates identity, performance, escrow, and disputes; CONSUMER populates identity and performance, with escrow, disputes, and governance null. §5.4 invariants split into §5.4.1 Chain Invariant (cross-link, always true), §5.4.2 Per-AID Scope Mutability (per-dimension rules table), §5.4.3 Cascading Narrowing, §5.4.4 Certificate Renewal (orthogonal to scope). §5.4.1 Role-Specific Delegation Patterns renumbered to §5.4.5. Tables renumbered: 5.6 PrincipalRef → 5.7; 5.7 DelegationLink → 5.14; 5.8 Delegation patterns → 5.16. New tables: 5.5 Document Infrastructure, 5.6 Identity pillar fields, 5.8 CertificateRef, 5.9 ScopeRef, 5.10 DelegationRef, 5.11 Performance pillar, 5.12 Escrow pillar, 5.13 Entity Governance pillar, 5.15 Per-dimension scope mutability. §2.2 Scope and Delegation Chain term definitions updated to describe concepts rather than enumerate fields. minimum_counterparty_cert_tier terminology corrected (certificate tier, not verification tier) and tier-value enumeration removed in favor of implementation-defined ordering. Certificate tier values, role-to-tier mapping, and lifecycle deferred to implementation-defined. Withdrawn: v0.1.6 (asymmetric-narrowing rationale defended the wrong rule). Subsequent in-place revision (May 24, 2026): §5.3 authorized_actions cell expanded with action semantics for purchase, sell, and delegate, with delegate semantics cross-referencing the new §5.4.6; §5.3 delegation cell now points to §5.4.6 for authorization

		and accountability rules. New §5.4.6 Delegation Authorization and Accountability — defines the delegator-authorization gate (every link's delegator MUST have delegate in its own authorized_actions, or the chain is invalid) and the top-of-chain accountability rule (the root of the chain is accountable for liability, escrow funding, PoP rating contribution, and dispute respondency; intermediate links act as relays and do not assume accountability by re-delegating).
v0.1.5	May 2026	§2.2 added Scope term; updated Authorized Actions term to reflect payment-bearing commitment focus (hire and contract removed from enum, leaving purchase, sell, delegate); updated Delegation Chain term to include in-place (cross-time) narrowing. §5.3 Table 5.5: authorized_actions description rewritten for payment focus and reduced enumeration; max_transaction_value description clarified to distinguish per-transaction cap from spending_limit; minimum_counterparty_cert_tier and minimum_counterparty_ar rows added. §5.4 generalized narrowing invariant to cover cross-link and cross-time mutation, with per-dimension narrowing semantics. §5.4.1 ENTERPRISE delegation pattern row: dropped "hiring/contracting authority" — replaced with sub-agent delegation authority. §9 PrivilegeSet permitted_operations: hire and contract removed from allowed types. §12 Future Work: additional payment-related action types, spending_limit detailed specification, and pre-commitment negotiation flexibility added as roadmap items.
v0.1.4	May 2026	Revision history relocated from title page to §14. No content changes. Establishes consistent format with the Framework (§14, v0.1.4) and Platform Spec Pillar 1 (§14, v1.7.2+).
v0.1.3	May 2026	§5.3 Table 5.5 authorized_actions row updated — type tightened from string[] to enum[], requirement raised from SHOULD to MUST, fixed enumeration published (purchase, sell, hire, contract, delegate), role compatibility rules added (CONSUMER MUST include purchase and MUST NOT include sell; PROVIDER MUST include sell and MUST NOT include purchase; ENTERPRISE MAY include either or both). Mandate alignment enforced via delegation chain per §5.4. Operational scenarios §10.1–10.3 illustrate example authorized_actions values for each role.
v0.1.2	May 2026	§5.2.1 strengthened — added explicit role/verification_tier matrix (Table 5.2c) and normative rejection semantics. §5.3 economic_role description cross-references §5.2.1. Operational scenarios §10.1–10.3 specify verification tier required for each role.
v0.1.1	April 2026	Initial published draft. Establishes the five pillars (Agent Identity, Proof of Performance, Liability Escrow, Dispute Resolution, Entity Governance), the basic/extended specification split, and operational scenarios.

End of AEA/P Specification v0.2.2 — Draft

https://aeap.dev