A Reference Architecture for AI Agents on Blockchain Infrastructure: Identity, Policy, Payments, and Custody as Composable Primitives

Ian Staley

Independent Researcher

ORCID: 0009-0000-8592-3186

Working Paper — Draft of April 25, 2026

Abstract

Autonomous AI agents are rapidly being deployed as economic actors capable of discovering services, negotiating terms, executing transactions, and custodying value on blockchain infrastructure. Existing wallet, identity, and payment models were designed for human principals operating with full discretionary authority and offer few primitives for the bounded, delegated, and revocable authority that agentic operation requires. This paper proposes a layered reference architecture for AI agents on blockchain infrastructure, organized around four composable primitives — verifiable agent identity, bounded policy, constrained-authority custody, and intent-based payments — and unified by an intent-based execution model in which identity is foundational. The architecture is grounded in emerging standards (ERC-8004, x402, ERC-4337, A2A, MCP, AP2) and mapped across heterogeneous settlement environments including the Ethereum Virtual Machine, Solana, the Canton Network, and XRPL. A case study of two parallel Canton initiatives — the proposed p402 trust-layer registries and the proposed Canton x402 facilitator — illustrates how the architecture's L1 identity / L4 payments separation manifests in practice and how privacy-preserving institutional ledgers reframe the agentic identity problem. A worked example using tokenized accounts receivable shows how the primitives compose under regulatory and operational constraints. The paper concludes with a discussion of audit, revocation, and open problems including agent-to-agent dispute resolution, cross-domain reputation portability, and the governance of solver networks.

Keywords: *AI agents, blockchain reference architecture, agent identity, Canton Network, agentic finance.*

JEL Classification: *G23, O33, K24.*

1. Introduction

The economic role of artificial intelligence has shifted from advisory to participatory. Within the last twenty-four months, large language model agents have moved from passive assistants that draft text and summarize documents to active intermediaries that select services, sign messages, and dispose of funds without sustained human supervision. This shift is most visible in three converging developments: the adoption of agent-to-agent communication protocols such as the Model Context Protocol (MCP) [1] and Agent-to-Agent (A2A) [2]; the introduction of HTTP-native payment rails for machine-to-machine commerce, most notably the x402 protocol jointly developed by Coinbase and Cloudflare [3] and its incorporation into Google's Agent Payments Protocol (AP2) [4]; and the standardization of on-chain agent identity through draft Ethereum Improvement Proposal ERC-8004 [5].

Each of these standards solves a narrow coordination problem in isolation. None addresses the deeper architectural question that emerges when they compose: how should the trust boundaries between an autonomous software agent, the principal on whose behalf it acts, the counterparty it transacts with, and the settlement infrastructure that records the result, be specified, enforced, and revoked? Andreessen Horowitz's research team [6], surveying the same landscape in early 2026, framed the gap as the absence of a Know Your Agent (KYA) layer analogous to KYC for human counterparties — cryptographically signed credentials binding an agent to its principal, permissions, constraints, and reputation. The conventional wallet-and-key model assumes that the holder of a private key is the legitimate decision-maker for every action that key authorizes. That assumption was always a useful fiction; in agentic contexts it is actively dangerous. An agent that holds a key with full transfer authority over a corporate treasury is not a wallet — it is a single point of catastrophic and undetectable failure.

This paper proposes a reference architecture that treats identity, policy, custody, and payments as composable primitives within a broader control plane, rather than as the architecture itself. The contribution is threefold. First, the paper articulates a layered model that separates intent expression from policy enforcement and from settlement, enabling agents to operate under bounded authority across heterogeneous chains. Second, it grounds each layer in concrete standards and implementations now in production or late-stage draft, including ERC-8004, ERC-4337 account abstraction, x402, AP2, the Canton party model, and Solana program-derived addresses. Third, it works through the architecture using a generic tokenized accounts receivable scenario and a contemporary case study of Canton-native initiatives that exemplify the architecture's L1 / L4 layer separation.

The central claim is therefore this: institutional AI-agent infrastructure should not be designed around autonomous wallets alone. It should be designed around a layered control plane in which verifiable identity, bounded policy, constrained custody, and intent-based payment execution compose across heterogeneous settlement environments, with identity as the foundational layer on which the others depend. The remaining sections develop this claim and demonstrate its instantiation.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the layered reference architecture. Sections 4 through 6 elaborate identity, payments, and custody, with policy treated throughout as the cross-cutting layer that binds them. Section 7 maps the architecture across four settlement environments and develops a case study of Canton's parallel p402 and x402-facilitator initiatives. Section 8 develops the tokenized receivables example. Section 9 addresses audit and revocation; Section 10 discusses open problems and concludes.

2. Related Work and Positioning

Three bodies of work bear directly on the architecture proposed here, and a fourth — the law and economics of delegated authority — is increasingly relevant though largely unincorporated in the technical literature.

2.1 Agentic AI standards and the KYA framing

The Model Context Protocol [1], introduced by Anthropic in late 2024 (released, widely deployed), standardizes how language model agents discover and invoke external tools. Google's Agent-to-Agent (A2A) protocol [2], released in 2025 and now governed under the Linux Foundation, complements MCP by specifying how agents communicate with one another, exchanging structured capability advertisements

("agent cards") and task descriptions. Google's Agent Payments Protocol (AP2) [4], released in September 2025 (public specification, early implementations) with x402 as one of its first extensions, defines an authorization and trust model for agent-initiated payments across rails including cards, real-time payments, and stablecoins. ERC-8004 (Trustless Agents) [5] (Ethereum draft EIP, August 2025), authored by De Rossi, Crapis, Ellis, and Reppel, extends A2A with three lightweight on-chain registries — Identity, Reputation, and Validation — explicitly designed to enable agent discovery and trust signaling across organizational boundaries without pre-existing relationships. The Identity Registry leverages ERC-721 to give each agent a portable, transferable on-chain handle that resolves to an off-chain registration file. ERC-8004 deliberately excludes payment mechanisms, leaving them to separate standardization.

The Andreessen Horowitz framing introduced in Section 1 [6] adds an important diagnostic that complements these protocol-level developments: the central problem is fragmentation, with vertically integrated fiat-first stacks on one side, crypto-native open standards on the other, and application-layer extensions of MCP attempting to bridge identity. The reference architecture proposed in this paper accepts the KYA framing as the diagnostic and offers a layered specification of how the various open standards compose to constitute it.

2.2 Account abstraction and programmable wallets

The Ethereum account abstraction trajectory, formalized in ERC-4337 [7] (final, deployed via mempool on EVM mainnet since 2023) and refined in subsequent proposals such as ERC-7702 [8] (final EIP, activated in the Pectra upgrade, May 2025), decouples the signing key from the account. A smart account can enforce arbitrary policy in code: spending limits, time locks, multi-signature requirements, social recovery, and session keys with bounded scope. Session keys in particular — keys delegated to a specific application or agent for a limited time and scope — are the closest existing primitive to what agentic delegation requires. Outside the EVM, Solana's program-derived addresses (PDAs), the Canton Network's external party model, and XRPL's permissioned-domain framework offer functionally analogous capabilities under different cryptographic and governance assumptions. Tooling such as MetaMask's Delegation Toolkit, Coinbase's AgentKit, and Merit Systems' AgentCash now expose these primitives at the application layer for agentic use.

2.3 Programmable payment rails

The x402 protocol [9] (open specification, live facilitators on Base, Solana, and additional EVM chains since 2025) revives the dormant HTTP 402 status code to embed stablecoin payments directly into web requests. A resource server responds to an unpaid request with a 402 status containing payment requirements; the client (human or agent) constructs a signed payment payload and retries; a facilitator verifies and settles on-chain. The protocol is network- and asset-agnostic, with live facilitators on every major EVM chain and Solana; the a16z research team [6] estimates roughly $1.6 million in monthly volume after filtering wash trading. A deferred-payment scheme proposed jointly by Coinbase and Cloudflare in late 2025 [3] (specification proposal, undergoing community review) extends x402 to support pre-negotiated licensing, batch settlement, and subscription billing — the patterns that institutional finance actually requires. The x402 Foundation, established jointly by Coinbase and Cloudflare in 2025 [3], governs the specification.

2.4 Privacy-preserving institutional ledgers

The Canton Network [10] (live institutional-grade public network, mainnet operational since 2024) operates as a network of networks in which each institution maintains its own sub-ledger and connects to others through a Global Synchronizer [11] (live, governed by the Canton Foundation under the Linux Foundation since 2025). Privacy is enforced at the protocol level through sub-transaction visibility: in a delivery-versus-payment, the cash leg is visible only to the cash parties, the securities leg only to the securities parties, and the regulator may observe both as a designated observer. The architectural implication for agentic operation is that an agent operating on Canton inherits a fundamentally different visibility model than the same agent operating on Ethereum mainnet, and identity, payments, and custody must therefore be specified abstractly enough to compose across both environments. As Section 7 will develop, two parallel proposals — the Cantor8-funded p402 trust layer and the FTP-proposed Canton x402 facilitator — instantiate the L1 identity and L4 payments layers of the architecture against this privacy-preserving substrate.

The reference architecture proposed in this paper sits at the intersection of these four literatures and contributes the missing integrative layer: a specification of how agent identity, programmable payments, and constrained custody compose under a shared intent-based execution model that satisfies the KYA criterion.

3. The Reference Architecture

Before introducing the layers, Table 1 fixes the meaning of the load-bearing terms used throughout the paper. These definitions are intended to be operational rather than philosophical: each term denotes a class of artifact or actor that the architecture must specify, instantiate, and constrain.

Table 1. Definitions of Key Terms

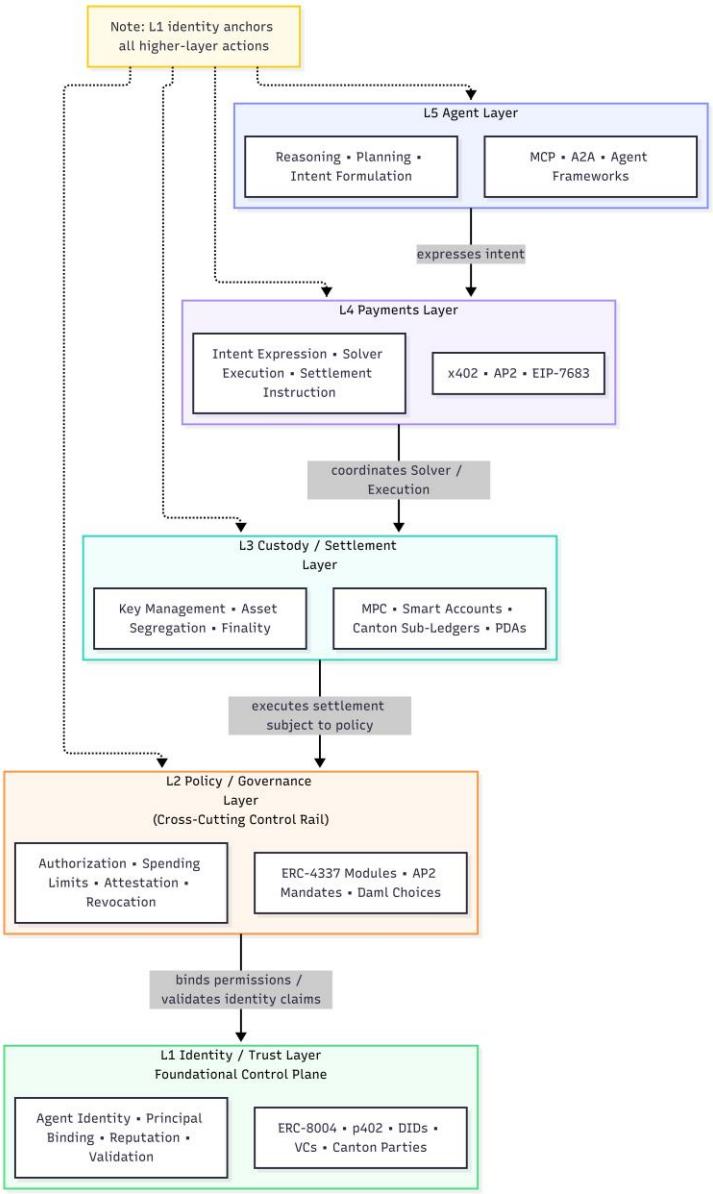
Term	Operational definition
Agent	Autonomous or semi-autonomous software actor that operates under delegated authority on behalf of a principal, capable of formulating intents and signing or invoking messages without per-action human approval.
Principal	Human, firm, DAO, or institution on whose behalf an agent acts; the source of the agent's authority and the party to whom the agent's actions are ultimately attributed.
Capability	Machine-checkable predicate over actions that defines the scope of permitted agent behavior (e.g., transaction type, counterparty, asset, aggregate limit, time window, validation oracle).
Intent	Declarative expression of a desired end-state plus constraints, distinct from a constructed transaction; the agent specifies what should be true, not how to make it so.
Solver	Execution actor that resolves intents into concrete settlement actions, subject to the principal's policy; may itself be an agent and is governed by the same identity, policy, and validation mechanisms.
Revocation	Cryptographic or contractual termination of an agent's authority, implementable on-chain by the principal and effective within the finality window of the underlying settlement environment.

With these terms fixed, the architecture is organized into five layers, each of which can be implemented through multiple competing standards while preserving the interfaces between layers. The layering is functional rather than physical: a single implementation may collapse adjacent layers into one component, and a sophisticated deployment may instantiate the same layer multiple times across different trust domains. Table 2 summarizes the layers and the primary standards or mechanisms currently in use at each, and Figure 1 illustrates the dependency relationships between layers.

Table 2. Layered Reference Architecture

Layer	Function	Representative standards
L5 Agent	Reasoning, planning, intent formulation	MCP, A2A, agent frameworks
L4 Payments	Intent expression, solver execution, settlement instruction	x402 (incl. Canton facilitator), AP2, EIP-7683
L3 Custody / Settlement	Key management, asset segregation, finality	MPC, smart accounts, Canton sub-ledgers, PDAs
L2 Policy / Governance	Authorization, spending limits, attestation, revocation	ERC-4337 modules, AP2 mandates, Daml choices
L1 Identity / Trust	Agent identity, principal binding, reputation, validation	ERC-8004, p402, DIDs, verifiable credentials, Canton parties

Figure 1. Five-Layer Control-Plane Reference Architecture



The five-layer reference architecture, ordered by strict dependency from L1 (Identity, foundational) up to L5 (Agent). Each layer presupposes the layers below it. Trust boundaries between layers must be enforced cryptographically rather than procedurally; the boundary semantics are summarized in Section 3.1. Policy (L2) is rendered as a cross-cutting control rail because it binds permissions to identity claims at L1 while constraining settlement at L3 and intent execution at L4.

Two design commitments distinguish this architecture from a simple stack diagram. First, identity is positioned as L1 — the foundational layer — deliberately: every higher layer presupposes a verifiable subject. Policy (L2) binds to identity; custody (L3) constrains identity-bound positions; payments (L4) authenticate as identity-and-policy-bound spend; and the agent (L5) acts under an identity it can prove. This ordering inverts the OSI mental model in which the physical substrate is foundational. In an agent-economy architecture, identity is the bedrock because nothing in the upper layers is meaningful without a subject to attribute it to. Second, the payments layer (L4) is treated as an intent-resolution layer rather than a transaction-construction layer. Agents express what they want to achieve — "settle invoice 4471 in USDC

within 24 hours subject to validation" — and a solver, governed by L2 policy, determines how that intent is realized against L3 custody. The intent abstraction is what allows the same agent logic to operate across EVM, Solana, Canton, and XRPL without rewriting against each chain's transaction model.

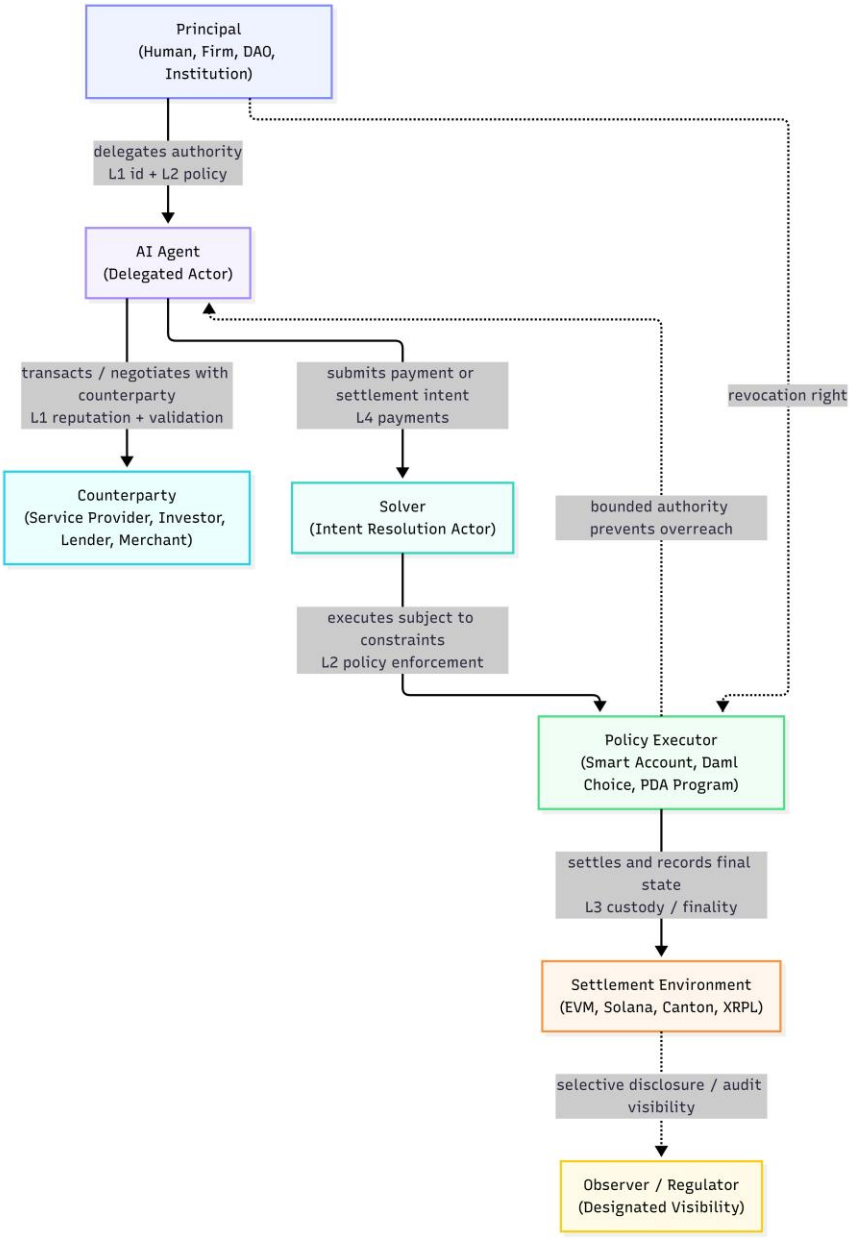
3.1 Trust boundaries

Five distinct trust boundaries cross the architecture and must be enforced cryptographically rather than by procedure. The boundary between principal and agent is enforced by L1 (the identity layer must distinguish actions taken by the agent on the principal's behalf from actions of the principal directly). The boundary between agent and counterparty is enforced by L1 reputation and L2 validation hooks. The boundary between policy author and policy executor is enforced by L2 attestation, typically via Trusted Execution Environments or zero-knowledge proofs of correct execution. The boundary between solver and settlement is enforced by L3 finality. The boundary between operator and observer is enforced by the privacy model of the chosen settlement environment.

4. Identity and Delegation

Agent identity is the foundational primitive. Without a verifiable, portable, and revocable handle that ties an agent to a principal and a scope of authority, every other layer of the architecture is built on sand. The identity layer must answer four questions: who is the agent, who is its principal, what is it authorized to do, and how is that authorization revoked. This is the technical operationalization of the KYA framing introduced in Section 2. Figure 2 illustrates the delegation flow and the actor roles defined in Table 1, showing how each transition is anchored in one or more of the architecture's layers.

Figure 2. Principal-Agent Delegation Flow



The principal-agent delegation flow with all six actor roles and the layers at which each transition operates. The Principal delegates authority to the AI Agent through L1 identity binding plus L2 policy. The Agent transacts directly with Counterparties (anchored in L1 reputation and validation) and submits payment or settlement intents to a Solver (L4 payments). The Solver executes subject to L2 policy enforcement at the Policy Executor (smart account, Daml choice, or PDA program), which records final state in the Settlement Environment under L3 custody and finality. Selective disclosure to designated Observers/Regulators is enforced by the privacy model of the chosen settlement environment. Dashed lines show the Principal's revocation right (which travels directly to the Policy Executor) and the audit-visibility path. Bounded authority at the Policy Executor prevents agent overreach even under compromise.

4.1 The agent identity record

Following the ERC-8004 pattern [5], an agent identity is anchored on-chain by a non-fungible token whose URI resolves to an off-chain registration file. The registration file declares the agent's name, capabilities, supported protocols (MCP, A2A, x402, AP2, and others), service endpoints, and payment

addresses. The on-chain anchor provides censorship resistance and portability; the off-chain file provides flexibility and supports rapid iteration of agent capabilities without re-minting identity. Critically, the registration file is signed by the agent's controlling key, and the wallet binding is verified through an EIP-712 challenge before any reputation or validation signal can be associated with the identity.

The identity layer must extend beyond ERC-8004 in three respects to be sufficient for institutional use. First, it must support principal binding — a verifiable statement, signed by the principal's identity key (typically a smart account or a Canton party), that the agent is authorized to act on the principal's behalf within a declared scope. Second, it must support delegation chains, where a parent agent may sub-delegate to specialized child agents under further-restricted scope. Third, it must integrate with conventional identity infrastructure, including W3C Decentralized Identifiers (DIDs) [12] (W3C Recommendation, July 2022) and Verifiable Credentials [13] (W3C Recommendation, May 2025), so that regulated counterparties can verify the principal's compliance status without disclosing it on a public ledger. The Canton-native p402 proposal, discussed in Section 7, addresses this third requirement by design: its registries are Daml-based, inheriting Canton's sub-transaction privacy, and selective disclosure is the default rather than an add-on.

4.2 Capability scoping

Capabilities are most usefully expressed not as role labels ("trader," "treasurer") but as machine-checkable predicates over actions. A capability might state, for example, that the agent may submit transactions of a specified type, drawn from a specified pool, denominated in specified assets, up to a specified daily aggregate, during a specified window, subject to a specified validation oracle. This formulation has two advantages: it is unambiguous to a policy enforcer, and it is composable, in the sense that the intersection of two capabilities is itself a capability. ERC-4337 session keys [7] provide a natural enforcement mechanism on EVM chains; Daml controller annotations provide the analogous mechanism on Canton; Solana programs may enforce capabilities through PDA-based authority checks. AP2 mandates [4] provide a portable, verifiable, revocable expression of capability scope that is rail-agnostic at the authorization layer.

4.3 Reputation and validation

Identity without reputation is identification without trust. The ERC-8004 Reputation Registry [5] provides a standardized interface for posting and querying feedback signals; the Validation Registry provides hooks for independent validators (re-execution, zkML verifiers, TEE attestations, trusted judges) to publish results. Recent academic work has begun formalizing the agent-identity layer specifically: Rodriguez Garzon et al. [14] propose pairing W3C DIDs with verifiable credentials to give LLM-based agents self-sovereign verifiable identities, while Huang et al. [15] argue that conventional OAuth/OIDC/SAML frameworks are inadequate for autonomous agents and propose a zero-trust agent IAM framework combining DIDs, an Agent Naming Service, and zero-knowledge proofs. The p402 proposal mirrors the registry pattern with a Service Record Registry that retains interaction history and performance metrics under selective disclosure, addressing the institutional concern that a public reputation record itself leaks competitive intelligence. For institutional agents, the reputation layer should additionally accommodate weighted attestations from regulated counterparties, time-decayed reliability scores, and tags that segment performance by capability domain. Reputation must be portable across chains; one of the more interesting open problems, addressed in Section 10, is how to aggregate reputation when the same agent identity operates simultaneously on Ethereum, Solana, and Canton.

5. Policy-Bounded Payments and Intents

The conventional model in which an agent constructs and signs a raw transaction is unsuited to bounded-authority operation for two reasons. First, transaction construction couples the agent to a specific chain and gas regime; a multi-chain agent must implement chain-specific signing logic. Second, raw transactions are difficult to constrain ex ante: a policy enforcer that inspects a serialized transaction must reason about the transaction's semantic effect, which in turn requires re-executing the transaction in simulation. The intent abstraction inverts this. The agent expresses what should happen — a target end-state, plus constraints — and a solver, operating under explicit policy, determines how to achieve it. The intent-centric formulation is most fully developed in Anoma's vision paper [16] (foundational research paper, 2021; protocol implementations are in active development but not yet at production scale), which formalizes versatile commitments to value, counterparty discovery, and solver settlement as primitives distinct from transaction execution.

5.1 The x402 model and AP2

The x402 protocol [9] provides the most concrete instantiation of agent-native payments to date. A resource server replies to an unpaid request with HTTP 402 and a structured payment requirement specifying network, asset, recipient, and amount. The client retries with a signed payment payload in a header; a facilitator verifies and settles. The protocol is asset-agnostic but in practice is dominated by USDC on Base and Solana for retail micropayments. Its limitation for institutional use is the assumption of immediate settlement: a corporate treasury cannot reasonably settle thousands of micropayments on-chain in real time. The deferred-payment scheme jointly proposed by Coinbase and Cloudflare in late 2025 [3] addresses this by decoupling cryptographic handshake from financial settlement, permitting batch and subscription patterns while preserving the one-line developer integration that made x402 viable. Google's AP2 [4] sits at a complementary layer of abstraction: where x402 specifies the wire-level handshake, AP2 specifies portable mandates that authorize an agent to spend up to declared limits at declared destinations under declared references — the authorization analog to the payment-rail handshake. Cross-chain intent settlement is addressed by EIP-7683 [17] (draft EIP, with reference implementations in production at multiple cross-chain solver networks), which standardizes the intent payload format across EVM environments.

5.2 Spending limits, allowances, and stream payments

Three programmable patterns recur across institutional use cases. The first is the allowance — a pre-authorized maximum that the agent may draw down without further principal approval, typically scoped to counterparty, asset, and time window. The second is the streaming payment — a continuous flow at a specified rate, used for subscriptions, salaries, and SaaS-like agent services. The third is the conditional payment — settlement contingent on an oracle-verified outcome, such as successful task validation under ERC-8004 [5] or delivery confirmation in a tokenized supply-chain transaction. All three patterns are expressible as intents at L4, encoded as AP2 mandates at the authorization layer, and enforced by L2 policy without exposing the underlying chain mechanics to the agent.

5.3 Solver governance

Intent-based architectures introduce a new actor — the solver — whose governance is non-trivial. A solver with discretion over execution path can extract value from the principal through poor routing,

frontrunning, or collusion — the foundational analysis of this class of adversarial reordering is Daian et al.'s "Flash Boys 2.0" [18], which coined the term miner extractable value and remains the canonical reference for adversarial extraction in solver-mediated execution. The architecture treats solvers as a constrained class of counterparties, governed by the same L1 identity, L2 policy, and validation mechanisms as any other agent. Solver selection is itself an L4 concern: the principal's policy may specify a whitelist, a competitive auction, or fallback to direct execution. Production systems such as NEAR Intents, CowSwap, and UniswapX provide reference implementations of solver-mediated execution at scale on the EVM and adjacent chains.

6. Constrained-Authority Custody

If identity binds an agent to a principal and policy bounds what the agent may do, custody bounds what the agent's keys can reach. The traditional answer — give the agent its own wallet with a small balance — is operationally fragile and accounting-unfriendly. The architecture proposed here treats agent custody as a constrained sub-account of principal custody, instantiated through one of three mechanisms depending on the settlement environment.

6.1 Mechanisms across environments

On EVM chains, the principal operates a smart account (ERC-4337 [7] or ERC-7702 [8]) and issues session keys to agents. A session key is a subordinate signing credential whose authority is enforced by the smart account's own code: it may sign transactions only of declared types, only to declared targets, only up to declared limits, and only during a declared window. The session key cannot exceed its scope even if compromised, because the smart account itself rejects out-of-scope transactions. Revocation is on-chain and immediate. Session keys are the closest existing analog to capability-based security in operating systems, and they are the EVM-native instantiation of L2 policy enforcement.

On Solana, custody of agent funds is most cleanly expressed through a program-derived address controlled by a multi-step program that enforces policy in its instruction handlers. The PDA holds funds; the agent submits instructions; the program enforces constraints before authorizing any transfer. Solana's account model and parallel execution make this pattern particularly efficient for high-throughput agent operations such as DeFi rebalancing or x402 micropayment streams.

On Canton, the agent and the principal are distinct parties, and the principal grants the agent rights through Daml choices on shared contracts. This model differs fundamentally from the smart-account pattern in that Daml's permissioning is contractual rather than custodial: the agent never holds funds in its own wallet; rather, it holds the right to exercise specific choices on contracts that move funds between principal-held positions. Sub-transaction privacy ensures that the agent's actions are visible only to parties explicitly designated as observers, which for institutional applications typically includes the principal's compliance function and a regulator. The Daml choice model is, in the author's view, the most expressive natively-supported instantiation of constrained-authority custody currently available, though it carries the overhead of a less mature developer ecosystem and a smaller validator network than the EVM.

6.2 Hardware roots of trust

Across all three custody models, the question of where the agent's signing keys actually live is increasingly answered by Trusted Execution Environments. The use of TEEs as an authenticated bridge

between smart contracts and off-chain logic was formalized by Zhang et al. in the Town Crier system [19] (foundational research paper, ACM CCS 2016), which provides the canonical academic foundation for TEE-attested off-chain execution. An agent whose keys are sealed in a TEE — and whose attestation can be verified on-chain — provides a much stronger assurance against operator compromise than an agent whose keys sit in a cloud key management system. Frameworks such as ROFL on Oasis [20] (live deployed framework) demonstrate how TEE-backed agent execution composes with on-chain identity standards to produce an end-to-end verifiable agent operation. For high-value institutional applications, hardware-rooted attestation should be considered a baseline rather than an enhancement.

7. Mapping the Architecture Across Settlement Environments

A reference architecture earns its name only if it can be instantiated across heterogeneous infrastructure. Table 3 summarizes how each of the five layers maps onto four settlement environments commonly encountered in institutional deployments. Note that XRPL's permissioned-domain framework [21] became active on mainnet in February 2026, providing an institutional-grade access-control primitive analogous to Canton's party model in functional terms.

Table 3. Layer Instantiation Across Environments

Layer	EVM (Ethereum, L2s)	Solana	Canton	XRPL
L5 Agent	MCP, A2A frameworks	MCP, A2A frameworks	MCP, A2A; Daml-aware agents	MCP, A2A frameworks
L4 Payments	x402 (USDC), EIP-7683	x402 (USDC), Jupiter intents	x402 facilitator (proposed); USDCx; CC	Native XRP, IOUs, RLUSD
L3 Custody	Smart account + MPC + TEE	PDA + MPC + TEE	Validator-hosted party + HSM	Multi-sign + regular keys
L2 Policy	4337 session keys; AP2 mandates	Program enforcement on PDA	Daml controller, observer	Hooks, deep-freeze, regular keys
L1 Identity	ERC-8004 NFT + DID	ERC-8004 mirror; SPL token	p402 (Daml registries) + party	Account + permissioned domain

Three observations emerge from this mapping. First, every layer admits a workable instantiation across all four environments, but the ergonomic match varies considerably. The EVM offers the richest agent-native tooling because the standards bodies have prioritized it; Canton offers the strongest native privacy and compliance posture; Solana offers the highest throughput and lowest latency; XRPL offers a payments-first design that makes L4 trivially cheap but provides weaker general-purpose smart-contract expressiveness. Second, identity portability across these environments remains an open problem: ERC-8004 [5] is technically deployable on any EVM-compatible chain and has been mirrored to BNB Chain, Base, and Hedera (via HCS-14 Universal Agent IDs), but no canonical mechanism currently exists to bind a single agent identity to actions on Solana, Canton, and XRPL simultaneously. Third, the choice of

settlement environment is increasingly driven less by technical capability than by the regulatory and counterparty profile of the application. Tokenized real-world assets with institutional counterparties gravitate to Canton; consumer-facing agent commerce gravitates to Base and Solana; cross-border wholesale payments gravitate to XRPL or Stellar. A mature agent architecture must accommodate all of these without forcing the agent logic to be rewritten for each.

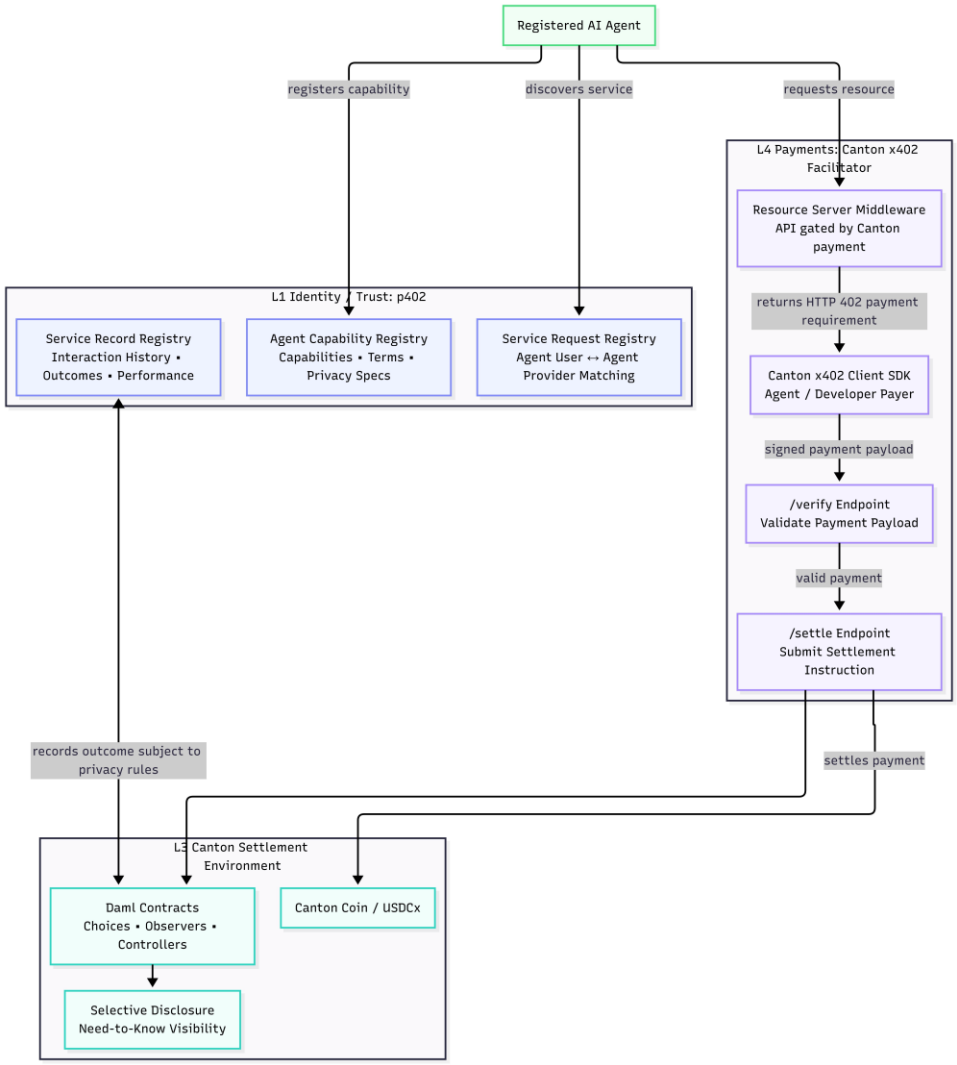
7.1 Case Study: Canton's Two-Track Approach to Agentic Infrastructure

Two parallel proposals announced in early 2026 instantiate the L4 payments and L1 identity layers of the architecture against the Canton substrate, and their separation is a useful empirical validation that those layers are genuinely distinct concerns rather than artifacts of the model. The first is the Canton x402 facilitator proposal submitted to the Canton Foundation Development Fund in March 2026 [22] by team FTP. The second is p402, a Canton-native trust-layer proposal funded through Cantor8's research program with Cambridge University researcher Yash Bharti [23], with a milestone transaction settled on Canton mainnet on April 23, 2026.

The FTP proposal [22] addresses L4. It delivers a Canton x402 facilitator implementing the standardized /verify and /settle endpoints, a Canton x402 client SDK for payers (developers and autonomous agents), and resource-server middleware allowing any Canton builder to gate an API endpoint behind Canton Coin payment with a one-line middleware import. The proposal is explicit that the facilitator is additive: it requires no changes to Canton core, Splice, or Daml. It accompanies the implementation with a formal scheme specification (scheme_exact_canton.md) submitted upstream to the x402 repository, following the structural pattern established by the prior SVM (Solana Virtual Machine) integration. The architectural significance is that x402's facilitator pattern is explicitly designed for new-chain integration without protocol modification: each chain contributes its own facilitator, the facilitator speaks the chain-specific settlement layer, and the upstream protocol remains unchanged. CanTrustAI, an inference-routing platform already deployed on Canton mainnet, serves as the production reference implementation.

The p402 proposal addresses L1. It is positioned by its sponsors as the Canton-native analog to ERC-8004 and consists of three Daml-based registries: an Agent Capability Registry through which institutional parties register agents and their service capabilities, terms of service, and privacy specifications; a Service Request Registry that matches agent users with agent providers under selective disclosure; and a Service Record Registry that maintains interaction history, outcomes, and performance metrics in a privacy-preserving form. The April 23, 2026 milestone transaction was initiated and executed by an Anthropic Claude agent, transferred USDCx (a 1:1 USDC-backed stablecoin developed with Circle's xReserve to provide confidential dollar settlement on Canton), and settled atomically with selective disclosure under the same privacy model that supports Canton's institutional repo and Treasury volume [23].

Figure 3. Canton-Native Agent Stack: p402 Trust Layer and Canton x402 Facilitator



How a registered AI Agent composes the two Canton initiatives into an end-to-end stack. The agent registers capabilities, discovers services, and transacts through the L1 p402 trust-layer registries (Agent Capability Registry, Service Request Registry, Service Record Registry). Resource requests pass through the L4 Canton x402 facilitator pipeline (Resource Server Middleware → Client SDK → /verify → /settle), which then settles to the L3 Canton Settlement Environment (Daml Contracts with Choices, Observers, and Controllers; Canton Coin or USDCx as the asset). Selective disclosure on the underlying ledger ensures that interaction outcomes recorded back to the Service Record Registry are visible only on a need-to-know basis. The diagram makes explicit how p402 (L1) and the FTP x402 facilitator (L4) compose without protocol modification on either side, validating the architecture's L1 / L4 separation thesis empirically.

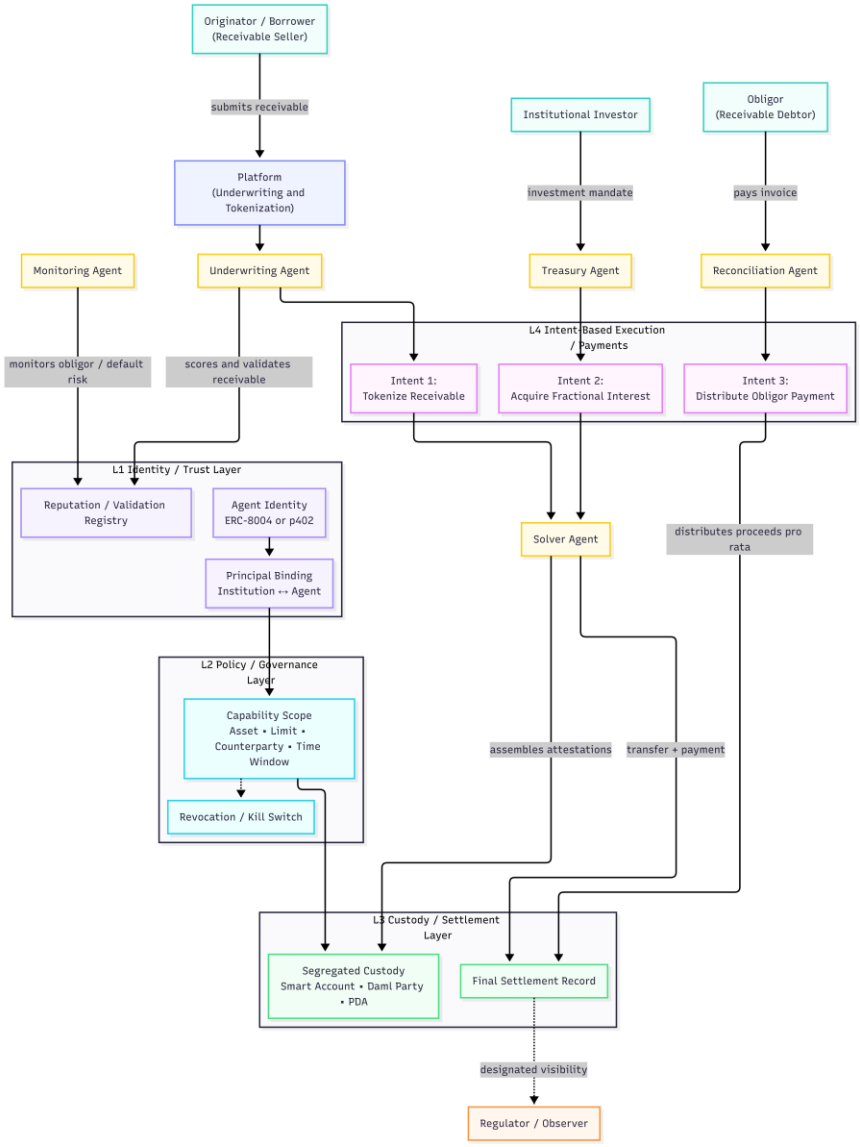
The two proposals are complementary, not competing. As Figure 3 makes explicit, FTP's facilitator instantiates L4 (the payment handshake and settlement) while p402 instantiates L1 (agent identity, capability advertisement, and reputation). A complete Canton-native agent stack composes both: an agent registered through p402's Capability Registry, transacting via the FTP-deployed x402 facilitator, settling in USDCx or Canton Coin, with all of identity, capability, and settlement subject to Canton's selective-disclosure visibility model. The case validates two architectural claims of this paper. First, the L1 / L4 separation is real: independent teams partitioned the agentic-infrastructure problem along precisely this

line, with one initiative addressing identity and trust as the foundation and the other addressing payments as the high-abstraction infrastructure. Second, privacy-preserving institutional ledgers reframe the agentic identity problem in ways that public-chain analogs cannot. ERC-8004's reputation model assumes interaction history is publicly readable; p402's Service Record Registry assumes the opposite. The architectural primitive is the same — a registry of bounded authority and accumulated trust — but visibility semantics inherited from the underlying chain force materially different design choices. Both proposals remain in proposed status as of this writing: FTP's Dev Fund #78 was submitted in March 2026 and is open; p402's milestone transaction has been demonstrated on mainnet but the registries are in active development. Treating either as deployed institutional infrastructure would be premature.

8. Worked Example: Tokenized Accounts Receivable

Consider an asset-based lending platform that tokenizes accounts receivable originated by middle-market commercial borrowers and sells fractional interests to institutional investors. Tokenization of trade-finance instruments has been the subject of recent authoritative analysis at the central-banking level — the Bank for International Settlements' 2024 report on tokenization concepts and implications for central banks [24] provides the relevant policy and infrastructure framing. The lifecycle of a single receivable involves at least the following actors: the obligor (the receivable's debtor), the originator (the platform's borrower), the platform (which underwrites and tokenizes), the lead lender, the syndicate participants, and the regulator. Several of these actors will, in practice, be assisted by AI agents — underwriting agents that score receivables, monitoring agents that track obligor financial health, treasury agents that rebalance investor positions, and reconciliation agents that match incoming obligor payments to the appropriate token holders. Figure 4 illustrates the complete lifecycle, showing how the four agent roles interact with the four layers of the architecture across three intent phases.

Figure 4. Tokenized Accounts Receivable Lifecycle Across the Architecture



End-to-end flow of a tokenized receivable from origination through obligor payment, mapped onto the proposed layered reference architecture. External actors (Originator, Institutional Investor, Obligor) interact with the Platform, which delegates work to four AI agents (Underwriting, Monitoring, Treasury, Reconciliation), each registered at L1 (Identity / Trust) via ERC-8004 or p402 with principal binding to the institution. Capability scoping at L2 (Policy / Governance) bounds each agent's authority by asset, limit, counterparty, and time window, with revocation and kill-switch as first-class L2 primitives. Three intent phases at L4 (Intent-Based Execution / Payments) — tokenize the receivable, acquire a fractional interest, distribute the obligor's eventual payment pro rata — are resolved by a Solver Agent that assembles attestations and orchestrates atomic transfer-and-payment. All settlement, segregated custody, and the final settlement record live at L3 (Custody / Settlement), with selective disclosure to a designated Regulator/Observer.

Map this onto the architecture. At L1, every agent is registered through an ERC-8004-style identity (or, on Canton, a p402 Capability Registry entry) anchored on the registry of whichever chain the platform settles on, with principal binding asserting which institution the agent acts for and what scope of authority it carries. The platform's underwriting agent, for example, would carry capability statements asserting that it may post draft underwriting decisions but may not alone authorize disbursement; an authorized human

or a separate compliance agent must counter-sign. At L2, policy is enforced through smart-account modules (if EVM) or Daml controllers (if Canton). The treasury agent's session key permits transfers only to the segregated settlement account and only against an invoice token whose validation status is "verified" in the relevant Validation Registry.

At L4, settlement of a receivable proceeds in three intent phases. The originator expresses an intent to tokenize a receivable; a solver — itself an agent — assembles the necessary attestations (KYC of the obligor, confirmation of invoice authenticity, valuation against historical performance) and constructs the mint transaction. The investor expresses an intent to acquire a fractional interest at or below a specified yield; a solver matches investor intents against newly minted tokens and executes the transfer-and-payment leg atomically. The obligor's eventual payment, when received, triggers a settlement intent that distributes proceeds pro rata among token holders; on Canton this is a single Daml choice, on EVM a smart-contract method call, and on Solana a program instruction. At L3, custody is constrained: investor funds sit in a segregated smart account or party whose only permitted outflows are to receivable-mint operations or to the investor's own withdrawal address.

The architecture's value in this example is most visible in the audit and exception paths. When a receivable defaults, every action taken by every agent against it — underwriting decisions, monitoring alerts, attempted collections — is recorded on-chain (or in the privacy-preserving Canton sub-ledger) under the relevant agent's verifiable identity, with policy enforcement preserving the regulator's ability to reconstruct the chain of authority. This is the operational realization of what governance literature [25] calls the audit trail without the audit committee — a continuous, cryptographically verifiable record of agentic action that does not depend on after-the-fact reconstruction by humans. On Canton specifically, the regulator may be a designated observer on the relevant contracts without becoming a counterparty, achieving regulatory observability without a transparent ledger.

9. Audit, Revocation, and Failure Modes

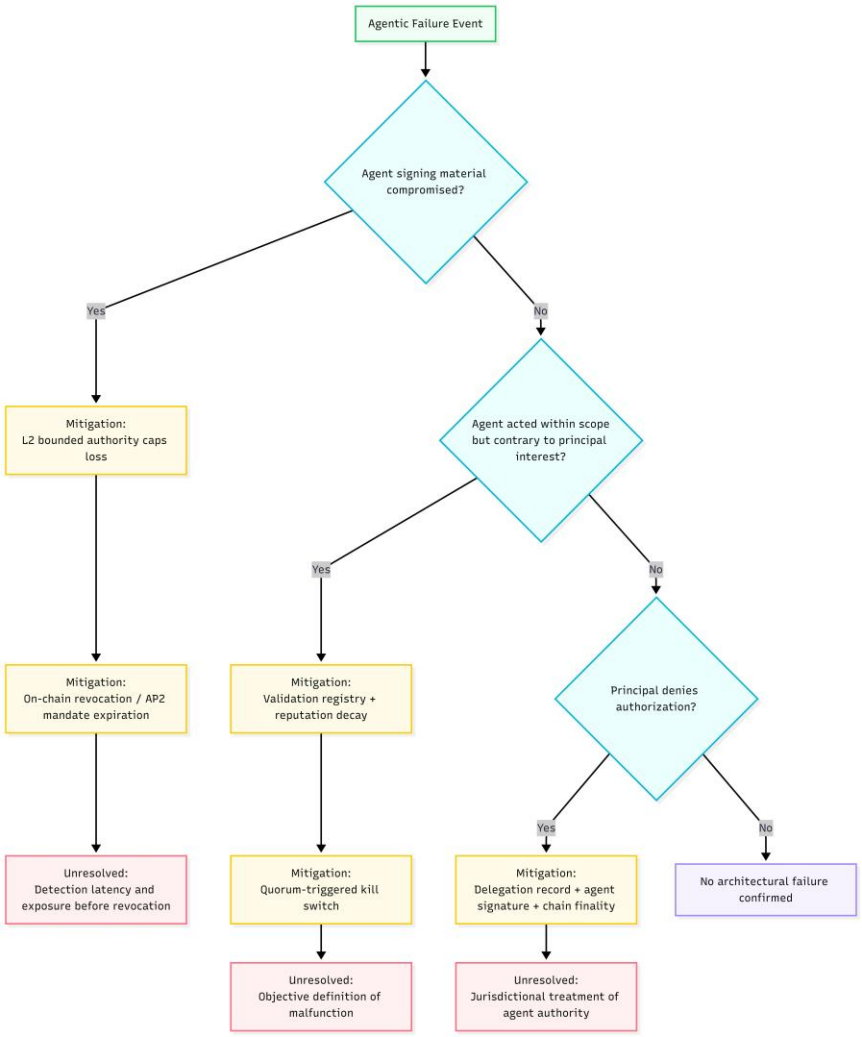
An architecture that does not specify how it fails is an architecture that has not been thought through. Three failure modes deserve explicit treatment, summarized in Table 4 and elaborated below. Figure 5 then renders the same content as a decision tree showing, given an agentic failure event, how to determine which failure mode is at play and which mitigation applies.

Table 4. Failure Modes, Mitigations, and Unresolved Aspects

Failure mode	Architectural mitigation	Unresolved
Agent compromise	L2 bounded authority caps loss to declared scope; on-chain revocation; AP2 mandates carry portable expiration semantics	Detection latency; worst-case loss before revocation can be material if limits or windows are large
Agent malfunction	L1/L2 validation for independent verification; reputation decay on detection; quorum-triggered kill-switch as L2 capability	Defining “contrary to interest” objectively; cross-chain reputation aggregation across multiple ledgers

Failure mode	Architectural mitigation	Unresolved
Principal repudiation	Cryptographic record of delegation, agent's signed actions, and chain finality establish a non-repudiable technical predicate	Whether jurisdictional contract law treats the record as binding; the architecture is necessary but not sufficient

Figure 5. Failure-Mode Decision Tree for Agentic Failure Events



Decision tree for diagnosing and responding to an agentic failure event. The first branch establishes whether agent signing material has been compromised; if so, L2 bounded authority caps the loss and on-chain revocation (or AP2 mandate expiration) terminates further exposure, with the residual unresolved aspect being detection latency before revocation takes effect. If signing material is not compromised, the second branch tests whether the agent acted within its declared scope but contrary to the principal's interest; if so, the Validation Registry plus reputation decay are the first-line mitigations, escalating to a quorum-triggered kill switch for high-value applications, with the residual unresolved aspect being objective definition of malfunction. If the agent's actions were in-scope and not contrary to interest, the third branch tests whether the principal is now denying authorization; if so, the technical predicate (delegation record + agent signature + chain finality) is necessary but not sufficient, with jurisdictional treatment of agent authority remaining unresolved as a matter of legal scholarship. If none of the three apply, no architectural failure is confirmed.

Agent compromise — a third party gaining control of the agent's signing material — is mitigated by the L2 policy layer's bounded authority. A compromised session key cannot exceed its declared scope; the worst-case loss is bounded by the spending limit and the time window before revocation. Revocation must be implementable on-chain in a single transaction by the principal and must take effect within the finality window of the underlying chain. The Canton revocation pattern, in which the principal exercises a Daml choice that retracts the agent's controller rights on relevant contracts, is exemplary in its cleanliness; analogous patterns exist for ERC-4337 modules and Solana programs but require more careful implementation. AP2 mandates [4] carry explicit expiration and revocation semantics at the authorization layer, providing a portable revocation primitive that travels across rails.

Agent malfunction — the agent acting within its scope but in a manner contrary to the principal's interest — is mitigated by the validation layer (independent verification of agent outputs before they are consumed by counterparties) and by the reputation layer (decay or revocation of trust scores when malfunction is detected). The April 23, 2026 Canton settlement [23] is illustrative: an agent transaction visible only to its counterparties and a designated regulator-observer permits ex-post forensic reconstruction without ex-ante public disclosure of the trade. For high-value applications, a kill-switch oracle that any of a quorum of trusted observers may trigger is appropriate. The kill-switch should be a first-class L2 capability, not a procedural override.

Principal repudiation — the principal denying that the agent acted on its behalf — is the hardest of the three because it is partly a legal rather than a technical problem. The architecture's contribution is to make the technical predicate verifiable: the principal's signature on the original delegation, the agent's signed actions within scope, and the chain's record of finality together establish a record that no party can plausibly deny. Whether a particular jurisdiction's contract law treats that record as binding remains a question for legal scholarship beyond this paper's scope; the architecture is a necessary but not sufficient condition for non-repudiation.

10. Open Problems and Conclusion

Three problems remain unresolved by the architecture proposed here. Cross-domain reputation portability is the most acute: ERC-8004 reputation is chain-specific, p402 reputation is Canton-specific, and an agent operating across both accumulates parallel records that are not directly comparable. Cryptographically attested cross-chain reputation aggregation, likely involving zero-knowledge proofs of reputation summaries, is an active research area but not yet standardized. Solver governance and value extraction is the second: intent-based payment architectures depend on solvers whose incentives are not perfectly aligned with principals'; the literature on order-flow auctions and MEV mitigation in DeFi [18] is the closest extant body of work, but agentic intents introduce dimensions (reputation effects, principal-specific policy) that complicate direct application. Agent-to-agent dispute resolution is the third: when two agents disagree about whether a task was completed or a payment was earned, the current state of the art is human escalation or simple oracle mechanisms, and neither scales.

These problems notwithstanding, the architecture is implementable today using standards either ratified or in late-stage draft. The principal contribution is integrative: by specifying how identity, policy, custody, and payments compose under a shared intent-based execution model, and by demonstrating that the composition is portable across heterogeneous environments — including the privacy-preserving institutional ledgers the public-chain literature has largely ignored — the paper offers a foundation on which

institutional applications of agentic AI can be designed without reinventing each layer per deployment. The natural integration points the practitioner first reaches for — identity, payments, wallets — are indeed the right primitives, but they are primitives within a control plane, not the architecture itself. The Canton case study demonstrates that independent industry teams converge on precisely the L1 / L4 separation the architecture predicts.

References

[1] Anthropic, "Model Context Protocol Specification," Nov. 25, 2024. [Online]. Available: https://modelcontextprotocol.io/specification

[2] Google LLC and Linux Foundation, "Agent2Agent (A2A) Protocol Specification," 2024–2026. [Online]. Available: https://github.com/a2aproject/A2A

[3] W. Allen, C. Whiteside, R. Lohe, and S. James, "Launching the x402 Foundation with Coinbase, and support for x402 transactions," Cloudflare Blog, Sep. 23, 2025. [Online]. Available: https://blog.cloudflare.com/x402/

[4] Google LLC, "Agent Payments Protocol (AP2) Specification," Apache 2.0 license, 2025. [Online]. Available: https://github.com/google-agentic-commerce/AP2

[5] M. De Rossi, D. Crapis, J. Ellis, and E. Reppel, "ERC-8004: Trustless Agents [Draft]," Ethereum Improvement Proposals, no. 8004, Aug. 2025. [Online]. Available: https://eips.ethereum.org/EIPS/eip-8004

[6] Andreessen Horowitz crypto research team, "The missing infrastructure for AI agents: 5 ways blockchains can help," a16z crypto, Apr. 2026. [Online]. Available: https://a16zcrypto.com/posts/article/5-ways-blockchains-help-ai-agents/

[7] V. Buterin, Y. Weiss, D. Tirosh, S. Nacson, A. Forshtat, K. Gazso, and T. Hess, "ERC-4337: Account Abstraction Using Alt Mempool [Draft]," Ethereum Improvement Proposals, 2021. [Online]. Available: https://eips.ethereum.org/EIPS/eip-4337

[8] V. Buterin, S. Wilson, A. Dietrichs, and M. Garnett, "EIP-7702: Set Code for EOAs," Ethereum Improvement Proposals (Final; activated in Pectra, May 2025), 2024. [Online]. Available: https://eips.ethereum.org/EIPS/eip-7702

[9] E. Reppel, R. Caspers, K. Leffew, D. Organ, D. Kim, and N. Dalal, "x402: An open standard for internet-native payments," Coinbase Developer Platform white paper, May 2025. [Online]. Available: https://www.x402.org/x402-whitepaper.pdf

[10] Digital Asset, "Daml Reference Documentation; Canton platform documentation," 2024–2025. [Online]. Available: https://docs.daml.com; https://docs.digitalasset.com

[11] Global Synchronizer Foundation (Canton Foundation), "Global Synchronizer governance under the Linux Foundation," 2025. [Online]. Available: https://sync.global

[12] M. Sporny, A. Guy, M. Sabadello, and D. Reed, "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, Jul. 19, 2022. [Online]. Available: https://www.w3.org/TR/did-core/

[13] M. Sporny, T. Thibodeau Jr., I. Herman, G. Cohen, and M. Jones, "Verifiable Credentials Data Model v2.0," W3C Recommendation, May 15, 2025. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/

[14] S. Rodriguez Garzon, A. Vaziry, E. M. Kuzu, D. E. Gehrmann, B. Varkan, A. Gaballa, and A. Küpper, "AI agents with decentralized identifiers and verifiable credentials," arXiv:2511.02841, 2025. [Online]. Available: https://arxiv.org/abs/2511.02841

[15] K. Huang, V. S. Narajala, J. Yeoh, J. Ross, M. Lambe, R. Raskar, Y. Harkati, J. Huang, I. Habler, and C. Hughes, "A novel zero-trust identity framework for agentic AI: Decentralized authentication and fine-grained access control," arXiv:2505.19301, 2025. [Online]. Available: https://arxiv.org/abs/2505.19301

[16] C. Goes, A. Sun Yin, and A. Brink, "Anoma: Undefining money — Versatile commitments to value," Anoma Foundation vision paper, 2021. [Online]. Available: https://anoma.net/vision-paper.pdf

[17] M. Toda, M. Rice, and N. Pai, "ERC-7683: Cross Chain Intents [Draft]," Ethereum Improvement Proposals, 2024. [Online]. Available: https://eips.ethereum.org/EIPS/eip-7683

[18] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in Proc. 2020 IEEE Symp. Security and Privacy, 2020, pp. 910–927. doi: 10.1109/SP40000.2020.00040.

[19] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town Crier: An authenticated data feed for smart contracts," in Proc. ACM CCS '16, 2016, pp. 270–282. doi: 10.1145/2976749.2978326.

[20] Oasis Protocol Foundation, "ROFL: Runtime Off-chain Logic for trustless TEE-backed computation," 2024–2025. [Online]. Available: https://docs.oasis.io/build/rofl/

[21] M. Vadari, "XLS-0080: Permissioned Domains (Final; activated on XRPL mainnet 4 February 2026)," XRP Ledger Foundation, 2024. [Online]. Available: https://xls.xrpl.org/xls/XLS-0080-permissioned-domains.html

[22] Canton Foundation, "Funding Proposal by FTP: x402 Protocol Integration for Canton (PR #78)," Mar. 12, 2026. [Online]. Available: https://github.com/canton-foundation/canton-dev-fund/pull/78

[23] BSCN, "Private AI agents have arrived on Canton Network: First fully private AI agent payment executed via Cantor8 university research program," BSCN News, Apr. 25, 2026. [Online]. Available: https://x.com/BSCNews/status/2047999864408031393

[24] Bank for International Settlements, Committee on Payments and Market Infrastructures, "Tokenisation in the context of money and other assets: concepts and implications for central banks," Oct. 2024. [Online]. Available: https://www.bis.org/cpmi/publ/d225.pdf

[25] K. Werbach, The Blockchain and the New Architecture of Trust. Cambridge, MA, USA: MIT Press, 2018. ISBN 978-0-262-03893-5.