Agent Marketplace Protocol

White Paper

March 2026

Abstract

Persistent autonomous agents can write code, manage infrastructure, and operate continuously for weeks. But they cannot register a domain from an evaluated provider, form an LLC, retain a registered agent, or manage a compliance filing. The bottleneck is not capability. It is the absence of a procurement layer connecting autonomous agents to the businesses that provide real-world services.

Open standards have begun to address agent communication (A2A), payment (x402, MPP, AP2), and identity (ERC-8004). Each deliberately leaves marketplace infrastructure unspecified: how an agent discovers a provider worth hiring, how a marketplace captures commission without taking custody of funds, and how an ongoing service relationship persists after the initial transaction.

This paper describes a protocol that builds the three components those standards omit. First, a curated discovery registry with computed behavioral reputation derived from real transaction data, including explicit buyer delivery confirmation. Second, a non-custodial atomic commission splitting settlement layer that is payment-rail agnostic. Third, trust infrastructure for ongoing agent-to-business relationships, using A2A task lifecycles and push notifications to support multi-step service delivery, renewals, and compliance alerts.

The protocol requires on-chain identity and a wallet from both buyer and provider. It is standards-first, building on A2A, x402, ERC-8004, and their successors. It launches with the service categories autonomous agents need most: the infrastructure to operate as legitimate economic entities.

The Procurement Gap

The Rise of the Persistent Autonomous Agent

The AI agent landscape has shifted from fragile prototypes to production infrastructure. Early systems like AutoGPT (2023) proved the concept of goal-directed agent loops but suffered from infinite loops, hallucinated results, and failure to converge.[1] The current generation has solved these problems at the architectural level.

OpenClaw, the open-source agent framework, accumulated roughly 327,000 GitHub stars in its first months and has become the dominant consumer agent platform.[2] NVIDIA shipped NemoClaw as a dedicated enterprise stack. Alibaba, Tencent, ByteDance, Baidu, and JD.com each deployed production OpenClaw implementations.[3] Alongside OpenClaw, frameworks like LangGraph, CrewAI, Google ADK, and the Microsoft Agent Framework[4] have converged on a shared architecture: agents that persist, maintain memory, create tools dynamically, and operate autonomously over extended timeframes.

The governance infrastructure is maturing in parallel. A2A was donated to the Linux Foundation in June 2025 and reached v1.0.0 on March 12, 2026, its first production-stable version, with over 150 partner organizations. MCP, goose, and AGENTS.md were donated to the Agentic AI Foundation (AAIF), also under the Linux Foundation, in December 2025. Both protocols now operate under open, vendor-neutral governance.[5]

This paper uses the term **persistent autonomous agent** to describe this category: an AI system that operates in a continuous execution loop, maintains state and memory across sessions, uses tools and external services to act on the world, and can run for days, weeks, or indefinitely. The distinction from earlier generations is persistence.

The Procurement Wall

Persistent autonomous agents are capable of remarkable things within digital environments. An agent can write and deploy a web application, manage cloud infrastructure, interact with APIs, and coordinate multi-step workflows. But there is a hard wall between digital capability and real-world economic participation. The wall is the absence of an entire procurement layer: the infrastructure for discovering, evaluating, purchasing, and managing real-world business services. An agent cannot: *Register a domain from an evaluated provider and maintain renewing it. Form a legal entity by selecting a jurisdiction, filing with a state agency, and maintaining compliance. Engage professional services involving multi-step scoping, quoting, and document exchange. Manage ongoing obligations on different schedules, each requiring timely action.*

---

[1] AutoGPT, released March 2023 by Significant Gravitas Ltd. Raised $12M October 2023. See Sequoia Capital, "Exploring Autonomous Agents" (2024).

[2] Jensen Huang, GTC 2026 keynote (March 2026). ~327,000 GitHub stars; reported by Reuters.

[3] Business Insider, "China Big Tech OpenClaw Gold Rush" (March 2026); Reuters, "Baidu Joins China's OpenClaw Frenzy" (March 2026).

[4] AutoGen merged into Microsoft Agent Framework alongside Semantic Kernel (RC February 2026).

[5] A2A donated to Linux Foundation June 2025; v1.0.0 March 12, 2026. MCP, goose, AGENTS.md donated to AAIF December 2025. Both under LF umbrella.

The agent can build an entire SaaS application in an afternoon but cannot procure the hosting, register the domain, or form the entity to operate it.

The Missing Marketplace Layer

Payment rails are rapidly commoditizing. x402 handles stablecoin settlement[6], Stripe's MPP bridges fiat and crypto[7], and AP2 connects to traditional financial infrastructure.[8] Agent registries and directories have emerged: Olas for on-chain agent services[9], Fetch.ai for agent matchmaking[10], Virtuals for agent tokenization[11], and x402 Bazaar and 402index for endpoint directories.[12]

All of these are digital service directories. None provides behavioral reputation derived from real transactions. None supports multi-step service delivery spanning days or weeks. None addresses procurement of real-world business services. None supports ongoing provider-agent relationships. What is missing is the marketplace layer connecting persistent autonomous agents to the real-world economy.

Standards Landscape and Competitive Position

Standards Foundation

Standard	Function	Governance	Status	Leaves Unspecified
A2A v1.0	Agent communication, task lifecycle, Agent Cards	LF (A2A Proj.)	Production	Discovery, vetting, marketplace
x402	Stablecoin payments via HTTP 402	x402 Foundation	Production	Trust, reputation, splitting
ERC-8004	On-chain identity, reputation registries	Community	Draft; 130K+	Scoring, marketplace curation
AP2	Payment authorization (mandates)	Google Cloud	Early (v0.1)	Discovery, evaluation
MPP	Dual-rail fiat + crypto payments	Stripe / Tempo	Launched Mar 26	Discovery, reputation
ERC-8183	Escrow-based job primitive	Community	Draft	Non-custodial settlement
ERC-8001	Multi-party coordination (EIP-712)	Community	Final	Orthogonal to marketplace

[6] x402: Coinbase + x402 Foundation (Cloudflare founding partner). Expanded to any ERC-20 via Uniswap Permit2 March 2026. Commercial volumes early-stage (CoinDesk/Artemis, March 2026).

[7] MPP launched March 18, 2026 by Stripe and Tempo ($500M raise, $5B valuation). Dual-rail fiat + crypto.

[8] Google Cloud, "Announcing Agents-to-Payments (AP2) Protocol." With PayPal, Visa, Stripe, Deloitte.

[9] Olas Marketplace (marketplace.olas.network); on-chain transaction data (self-reported).

[10] Fetch.ai documentation; ASI Alliance metrics. Self-reported, includes test/inactive registrations.

[11] Virtuals Protocol documentation. ~13% graduated from prototype. ACP ~$1M transactions as of March 2026.

[12] x402 Bazaar (bazaar.x402.org); 402index.io (15K+ endpoints); satring.com.

The protocol builds the three things that A2A, x402, and ERC-8004 deliberately leave unspecified: curated discovery with computed reputation, non-custodial marketplace settlement, and trust infrastructure for ongoing agent-to-business relationships.

Competitive Landscape

Project	Scope	Trust Model	Payment	Real-World	Ongoing	Buyer Signal
Olas	On-chain agents	Staking	Crypto	No	No	No
Fetch.ai	Agent matching	Almanac	Token	No	No	No
Virtuals	Agent tokens	ERC-8004	Token	Minimal	No	No
x402 Bazaar	Endpoint dir.	None	x402	No	No	No
Salesforce	Enterprise	Platform	Fiat	No (digital)	Platform	No
Google UCP	Consumer	Merchant	Fiat+crypto	Consumer	Order mgmt	No
This Protocol	A2B marketplace	Computed + buyer	Rail-agnostic	Yes	A2A push	Yes

No protocol currently combines curated procurement of real-world business services with non-custodial split settlement, computed behavioral reputation, and ongoing relationship infrastructure. The protocol's differentiation is partly structural: some launch categories produce externally verifiable artifacts. Artifact existence is not artifact correctness; the protocol supplements verifiability with an explicit buyer delivery confirmation signal.

What This Protocol Builds

1. Curated discovery with computed reputation. A permissionless on-chain registry paired with curated front-ends surfacing vetted providers alongside behavioral reputation signals and buyer delivery confirmation.

2. Non-custodial atomic commission splitting. A payment router that atomically splits each payment: service price to provider, commission to DAO treasury. Payment-rail agnostic.

3. Trust infrastructure for ongoing relationships. Behavioral signals from every transaction plus A2A push notifications for renewals, compliance alerts, and service updates.

Core Thesis and Design Principles

> *A general-purpose protocol for agent-to-business commerce, launching with the service categories autonomous agents need most: the infrastructure to operate as legitimate economic entities.*

The protocol is not an agent framework, a payment rail, or an identity standard. It is the marketplace infrastructure that sits on top of all three. Seven principles govern every architecture decision. They are constraints, not aspirations.

1. On-chain identity required. Both buyer and provider must hold an ERC-8004 identity and control a wallet. This is a prerequisite for participation. The protocol attributes payments, accumulates reputation, and persists service relationships through on-chain identity. There is no anonymous or identity-free interaction path at the protocol level.

2. Agent sovereignty. The protocol does not enforce spending limits, budget approvals, or risk policies. If an agent presents a valid payment, it receives service. Spending governance is the operator's responsibility. The protocol is compatible with external authorization frameworks (such as AP2 mandates) without requiring them.

3. Standards-first. Build on A2A, x402, ERC-8004, AP2, and their successors. Fill gaps with minimum additional infrastructure. Complement standards; do not compete with them.

4. Non-custodial by design. Funds never rest in the protocol's contracts. The payment router executes atomic splits in a single transaction. No escrow, no holding period, no custodial intermediary, no dispute arbitration.

5. Computed trust with buyer confirmation. Trust signals combine behavioral byproducts of protocol activity with an explicit on-chain buyer delivery confirmation. Raw data is stored on-chain or anchored on-chain; scoring is an off-chain concern.

6. Permissionless registry, curated discovery. The on-chain registry is open to any provider that pays the listing fee. The discovery layer is curated via KYB verification. Multiple front-ends can apply different curation policies.

7. Payment-rail agnostic. The settlement layer accepts any rail delivering stablecoins to the splitting contract. x402 is the primary rail at launch. Third-party gateways may support fiat-compatible rails outside the core protocol.

System Architecture

The architecture is organized as four layers with distinct trust properties.

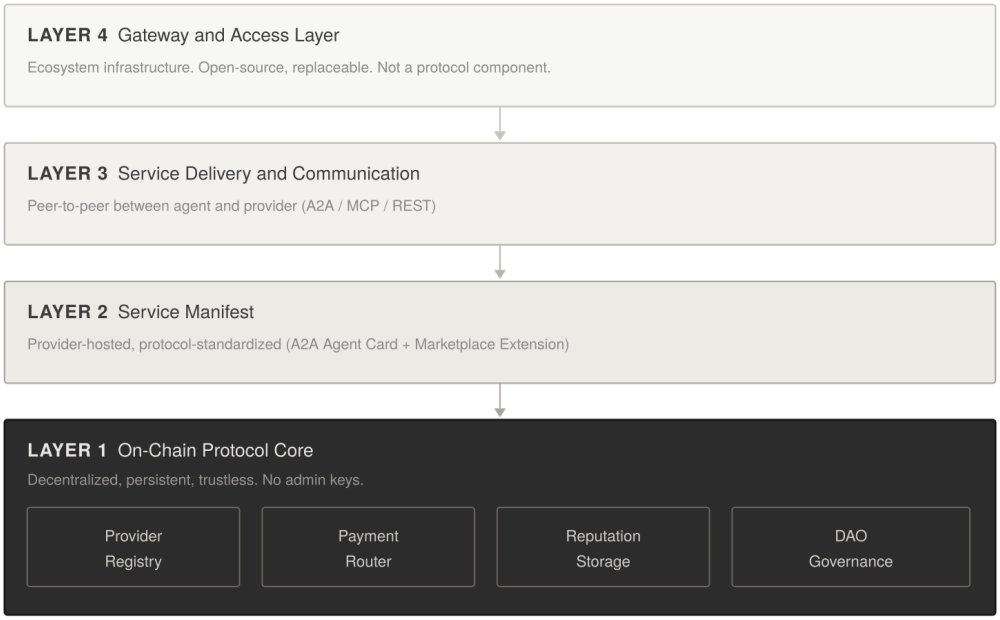

Figure 1. Protocol Architecture Layers

The on-chain protocol core handles registration, settlement, reputation data storage, and governance with no admin keys and no dependency on off-chain components. The service manifest standardizes provider descriptions. Service delivery operates peer-to-peer. Gateways are ecosystem infrastructure built by third parties; they are not protocol components.

Contracts are standard Solidity with no chain-specific dependencies, deployable on any EVM-compatible chain.

Provider Registry

The provider registry is a permissionless smart contract mapping provider wallet addresses to service metadata. The following schema is illustrative; the final implementation may differ.

```
ProviderRegistryEntry {
  providerAddress:    address     // Destination for payment splits
  erc8004TokenId:     uint256     // Link to ERC-8004 Identity Registry
  agentCardURI:       string      // HTTPS URL or IPFS CID to manifest
  registrationTime:   uint256     // Block timestamp at registration
  isActive:           bool        // Provider-toggled; inactive preserves history
}
```

The registry makes no quality judgments and applies no eligibility criteria. Any entity that pays the listing fee can register. Curation is a front-end concern. Registration is a single transaction; deregistration preserves reputation history.

Payment Router

The payment router is the settlement contract. Its function is narrow and atomic: receive a stablecoin payment, split it, and emit a confirmation event. No funds rest in the contract.

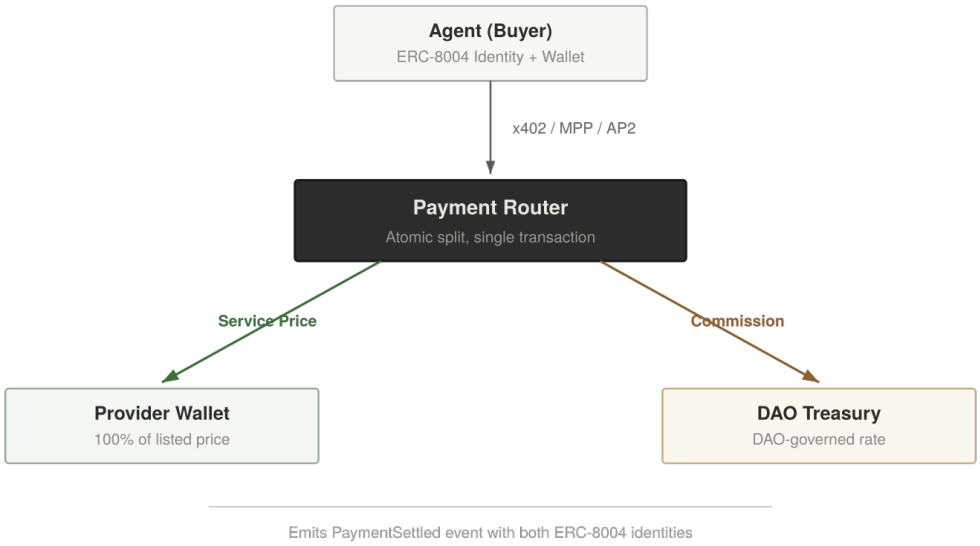

Figure 2. Payment Flow and Atomic Splitting

Payment-rail agnostic interface. The router accepts any ERC-20 stablecoin transfer accompanied by a provider identifier. x402, MPP, AP2, or any future protocol producing an on-chain stablecoin transfer all yield identical settlement artifacts.

Settlement mechanics. The router reads the commission rate from the DAO governance contract, splits the payment (provider wallet + DAO treasury), and emits a PaymentSettled event with both ERC-8004 identities. Because both parties sign their own transactions, payment attribution is unambiguous.

Commission rate. DAO-governed. Single global rate at launch. Future governance may introduce category-specific or volume-tiered rates.

Identity Model

The protocol requires every participant to hold an **ERC-8004 identity**: an ERC-721 NFT in the on-chain Identity Registry. ERC-8004 is deployed on Ethereum mainnet (January 2026), BNB Chain (February 2026), and Solana (March 2026), with over 130,000 registered agents (includes test and inactive entries).[13]

Identity ties together payment attribution, reputation accumulation, and service relationship continuity. The provider associates service accounts with the buyer's ERC-8004 identity for future authentication. On-chain identity transfer (ERC-721) is clean; off-chain implications vary by service and may require re-verification. The protocol does not prescribe how identities are provisioned.

[13] ERC-8004 EIP (eips.ethereum.org). Mainnet Jan 29, 2026; BNB Chain Feb 4, 2026; Solana Mar 3, 2026. 130K+ agents, includes test/inactive.

Trust Model and Behavioral Metrics

The trust model combines process metrics with explicit buyer delivery confirmation. Review-based systems exhibit well-documented inflation: Airbnb listings average 4.7/5 with 94% at 4.5 or above, making automated discrimination nearly useless.[14]

Process metrics are derived automatically:

Metric	Derivation	Signal Type
Completion rate	% tasks reaching completed vs. failed/canceled	Reliability
Fulfillment accuracy	Delta between stated and actual turnaround	Honesty and speed
Responsiveness	Average time in input-required state	Multi-step quality
Longevity	Renewal rate for ongoing services	Long-term reliability
Volume	Total transactions for provider	Confidence weight

Buyer delivery confirmation is an explicit on-chain attestation submitted after task completion: was this service delivered to your satisfaction? This binary signal captures outcome quality. A provider can be fast yet deliver an incorrect result; process metrics alone would not catch this. The buyer signal does.

Raw data, not scores. Per-transaction signals are stored as attestations via the Ethereum Attestation Service (EAS).[15] No aggregated scores are computed on-chain. EAS supports both on-chain and off-chain attestations; the protocol may use off-chain attestations with on-chain anchoring to balance cost, latency, and privacy.

Confirmation mechanics. A buyer may submit a confirmation attestation at any time after an A2A task reaches a terminal state (completed, failed, or canceled). There is no expiration window. A buyer may also revise their attestation; the most recent submission is the active signal. This is deliberate: for the service categories the protocol targets (entity formation, compliance, registered agent), quality problems may not surface for weeks or months. A buyer who discovers that an LLC was filed incorrectly six months later can update their confirmation accordingly. An absent confirmation carries no weight; it is not treated as a negative signal. This prevents weaponizing inaction: a competitor cannot degrade a provider's reputation by purchasing services and simply not responding. Only explicit "Confirmed" and "NotConfirmed" attestations affect the provider's record.

Why binary. Scored ratings (1–5 stars) produce the inflation problem cited above. A binary signal forces a meaningful choice and yields cleaner input for automated consumers. The primary Sybil defense is economic: every confirmation is anchored to a real payment through the router. Farming positive confirmations requires spending real money on real transactions, making manipulation costly in proportion to its effect on reputation.

The following schema is illustrative; the final implementation may differ.

```
ReputationRecord {
  providerIdentity:      uint256   // Provider ERC-8004 token ID
  agentIdentity:         uint256   // Buyer ERC-8004 token ID
  transactionOutcome:    enum      // Completed | Failed | Canceled
```

[14] Zervas, G., Proserpio, D., Byers, J. (2021). "A First Look at Online Reputation on Airbnb." Marketing Letters.

[15] Ethereum Attestation Service (docs.attest.org). Supports both on-chain and off-chain attestations.

```
  buyerConfirmation:     enum      // Confirmed | NotConfirmed | Pending
  fulfillmentTime:       uint256   // Seconds: payment to terminal state
  fulfillmentAccuracy:   int256    // Stated turnaround minus actual
  timestamp:             uint256   // Block timestamp
}
```

Recourse Design

The protocol deliberately excludes on-chain dispute arbitration. Real-world service disputes (was the LLC filed correctly, did the registered agent forward the compliance notice) require subjective judgment about off-chain outcomes. No on-chain mechanism can adjudicate these without reintroducing a trusted third party. Escrow-based alternatives such as ERC-8183 introduce holding periods, custodial risk, and arbitration governance that conflict with the protocol's non-custodial design. Instead, the protocol treats reputation as the recourse mechanism. A provider who accepts payment and fails to deliver accumulates negative signals (failed outcomes, missing buyer confirmations, declining completion rates) that are publicly readable on-chain and reduce future transaction volume through discovery ranking. The cost of defection scales with accumulated history: the longer a provider operates honestly, the more expensive it becomes to defect.

Early buyers transacting with unproven providers bear the highest risk. Front-end curation mitigates this: KYB-verified providers have real-world legal identities at stake, creating off-chain accountability independent of on-chain history. Listing fees impose a cost floor that makes throwaway identities economically unattractive. The core protocol does not mandate provider staking bonds, but the architecture is compatible with ecosystem extensions (optional bond contracts, third-party insurance, gateway-level guarantees) that individual front-ends may adopt.

Service Manifest and Discovery

Every provider describes its service through an A2A Agent Card. The standard card does not describe pricing, billing, jurisdiction, or lifecycle characteristics. The protocol defines a marketplace extension:

```
{
  "pricing": {
    "amount": "string",
    "currency": "USDC | USDT | DAI",
    "billingModel": "one-time | per-interaction | subscription",
    "subscriptionPeriod": "ISO 8601 duration",
    "renewalTerms": "string"
  },
  "onChainReferences": {
    "registryAddress": "string (required)",
    "paymentRouterAddress": "string (required)",
    "erc8004TokenId": "string",
    "chainId": "integer"
  },
  "jurisdictions": ["ISO 3166-1 alpha-2 codes"],
  "serviceLifecycle": "one-shot | ongoing (required)",
  "turnaroundEstimate": "ISO 8601 duration",
  "notificationPolicy": {
    "supportsNotifications": "boolean",
    "notificationTypes": ["renewal", "compliance-deadline",
                          "service-update", "action-required"],
    "renewalLeadTime": "ISO 8601 duration"
```

```
  },
  "requiredBuyerAttestations": ["string"]
}
```

Providers host their own Agent Cards and update pricing without an on-chain transaction. The on-chain registry stores only the pointer.

Structured queries: Filters on category, jurisdiction, price, turnaround, completion rate, volume. Returns ranked providers.

Semantic queries: Natural language parsed into filters + semantic matching against Agent Card skill descriptions.

Discovery is not a protocol guarantee. It is built on publicly available on-chain data. Anyone can build a competing service with different algorithms or policies.

Service Delivery and Communication

On confirmed payment, the provider creates an A2A task progressing through states mirroring real-world workflows:

State	Meaning	Example (LLC Formation)
submitted	Order received	Provider has requirements, preparing filing
working	Actively processing	Filing submitted to state
input-required	Provider needs info	Name conflict; buyer must choose alternative
completed	Delivered; artifacts attached	Formation docs, EIN, management access
failed	Could not deliver	State rejected filing
canceled	Canceled before completion	Buyer withdrew request

Terminal states trigger reputation recording. After terminal state, the buyer submits their delivery confirmation attestation.

A2A for conversational multi-step services, MCP for structured single-operation services, REST as universal fallback. Process metrics (state transitions) are only fully available for A2A-based delivery; MCP/REST produce simpler reputation signals.

Push Notifications

Type	Use Case
renewal	Service nearing renewal date
compliance-deadline	Filing deadline relevant to prior purchases
service-update	Material change to service
action-required	Provider needs agent action outside active task

This ongoing relationship layer is a core differentiator. No existing system supports provider-initiated communication creating new tasks within an existing relationship.

Gateway and Access Layer

The core protocol requires on-chain identity and a wallet from every participant. In practice, the majority of agents in the near term are not natively crypto-capable. Third-party gateways can bridge this gap by providing wallet management, transaction signing, identity provisioning, and fiat-to-stablecoin conversion as a service layer outside the core protocol.

The protocol publishes an open-source reference gateway implementation. Anyone can deploy one. Multiple gateways can coexist, settling through the same on-chain contracts. The founding entity operates the initial gateway under contract to the DAO.

The key property is that gateways are not dependencies. A crypto-native agent can interact with the full protocol without any gateway. If a gateway goes offline, on-chain state persists. The centralized tier enhances accessibility; the decentralized tier guarantees continuity.

End-to-End Example: Domain Registration

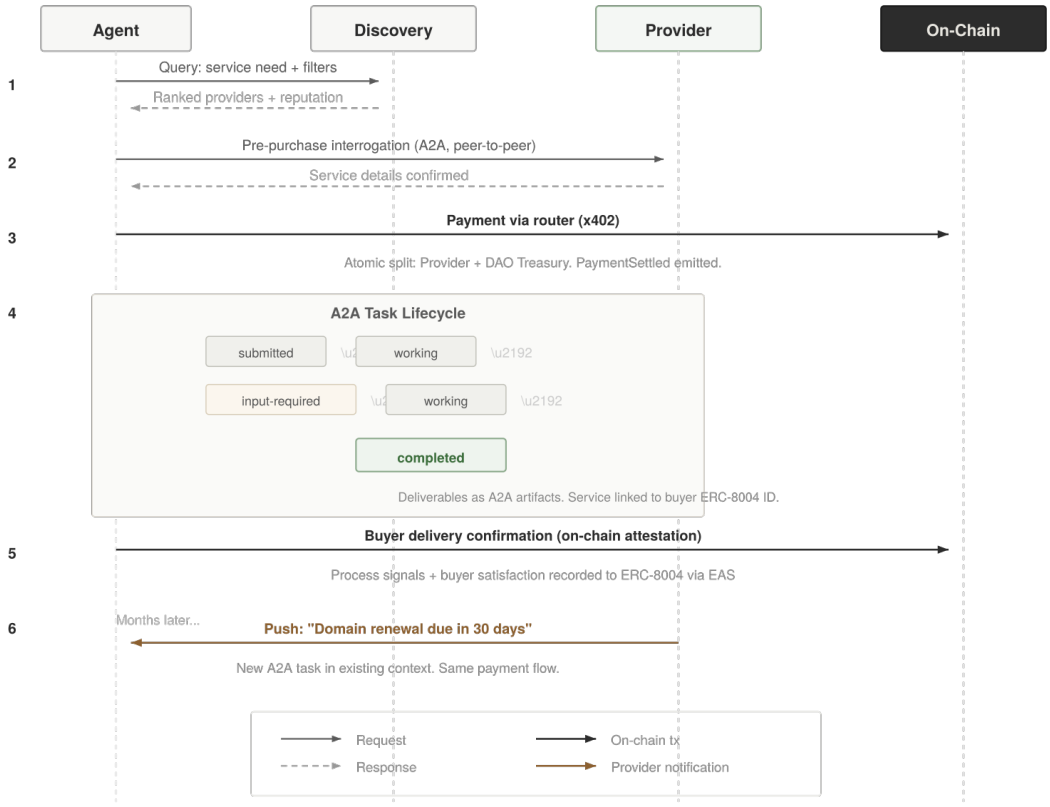

Figure 3. End-to-End Transaction Flow

Discovery. Agent queries a discovery service. Returns providers with reputation data.

Pre-purchase. Agent connects to provider's A2A endpoint for specifics. Peer-to-peer; the protocol does not observe.

Payment. Agent pays via payment router. Atomic split. Both ERC-8004 identities recorded.

Delivery. A2A task lifecycle through completion. Service account linked to buyer's ERC-8004 identity.

Buyer confirmation. On-chain attestation: service delivered to satisfaction. Signals recorded to reputation registry.

Ongoing relationship. Months later, provider pushes renewal notification. New A2A task, same context, same payment flow.

Protocol Threat Model

This section addresses protocol-level attack vectors and the mitigations designed into the system.

Sybil attacks and reputation farming. The protocol does not claim to solve Sybil attacks algorithmically. For broad classes of reputation mechanisms, profitable Sybil strategies exist.[16] Colluding buyer-provider pairs can farm positive confirmations, but every confirmation requires a real payment through the router, making farming costly in proportion to its effect on reputation. Primary mitigation is layered: KYB verification at the curation layer ties providers to real-world legal identities, listing fees impose a cost floor on registry participation, and structural verifiability of real-world artifacts (e.g., a domain registration is publicly queryable) provides an independent check on delivery claims.

Confirmation griefing. A competitor could purchase services and submit NotConfirmed to degrade a provider's reputation. The attacker pays full service price for each negative signal, and a single NotConfirmed against a provider with hundreds of positive confirmations has minimal impact. Cost of attack scales linearly; damage scales sub-linearly.

Registry spam. The permissionless registry can be flooded with junk listings. Listing fees create a cost floor. The curation layer filters discovery results, and listings with no transaction history rank last. Spam degrades the on-chain registry but not the curated discovery experience.

Payment router exploitation. The payment router executes a single-transaction atomic split with no held state. No funds rest in the contract between transactions. Standard reentrancy guards apply. The attack surface is narrow by design: the router reads a commission rate, splits an ERC-20 transfer, and emits an event.

Stale Agent Cards. A provider could change terms after a buyer queries the Agent Card but before payment. The on-chain registry stores only a pointer; the protocol does not guarantee term consistency between discovery and settlement. Buyer agents should verify pricing and terms at payment time. A provider who routinely bait-and-switches accumulates negative buyer confirmations.

Use Cases and Service Categories

Where the protocol creates the widest gap between what exists and what agents need: services with real-world fulfillment, multi-step delivery, and ongoing relationships.

[16] Cheng, Y. et al. (2021). "On the Limits of Reputation-Based Mechanisms." AAMAS 2021.

Category	Lifecycle	Complexity	Key Protocol Features
Entity formation	Multi-step, ongoing	High	Full A2A lifecycle, push notifications, documents
Domain registration	One-shot + ongoing	Low-Med	Quick settlement, renewal notifications
Website hosting	Subscription	Low	Subscription billing, capacity notifications
Email / comms	Subscription	Low	Provisioning + subscription management
Registered agent	Subscription (req.)	Medium	Compliance-critical push notifications
Compliance	Multi-step, recurring	High	Jurisdiction workflows, deadline notifications
Physical mail	Ongoing	Low-Med	Provider-initiated notifications
Professional svc.	One-shot	Medium	Structured deliverables, buyer confirmation

These form dependency chains: entity formation produces an LLC, required for a tax ID, required for a bank account. Compliance follows: registered agent, licensing, filings. Digital presence runs in parallel: domain, hosting, email. Agents compose these through sequential purchases.

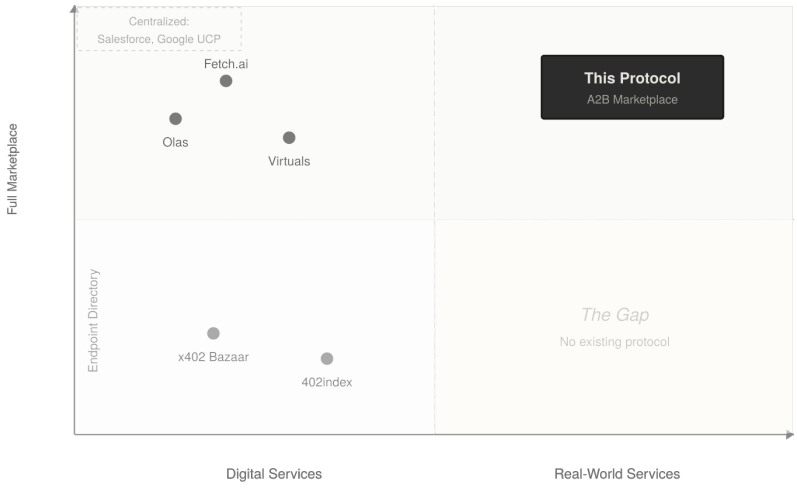

Figure 4. Competitive Positioning

Governance and Economic Model

The protocol is governed by a DAO LLC. Governance operates through token-weighted voting with a timelock delay on execution. There are no admin keys and no multisig shortcut. This makes the "no admin keys" commitment in the protocol's design principles enforceable rather than aspirational. At formation, founders hold a supermajority of governance tokens. Decentralization happens progressively. The governance token design, supply model, distribution schedule, and detailed incentive analysis are described in a companion economic paper.

The DAO exists primarily to control contract upgrade authority: protocol contracts are upgradeable via proxy pattern, and only a governance vote can authorize a new implementation. Beyond upgrades, token-weighted voting controls on-chain parameters (commission rate, listing fees), treasury disbursement (commission and listing fee revenue flows to a DAO-controlled

treasury contract; spending requires a governance vote), and ecosystem development including funding front-end tooling, reference implementations, and contractor engagement.

Revenue Model

Transaction commissions. A configurable buyer-side percentage via atomic splitting. Providers receive 100% of listed price.

Listing fees. Recurring stablecoin fee for active registry listing. Revenue, Sybil resistance, and natural churn.

Both streams are stablecoin-denominated. The protocol does not require a native token for settlement or service access.