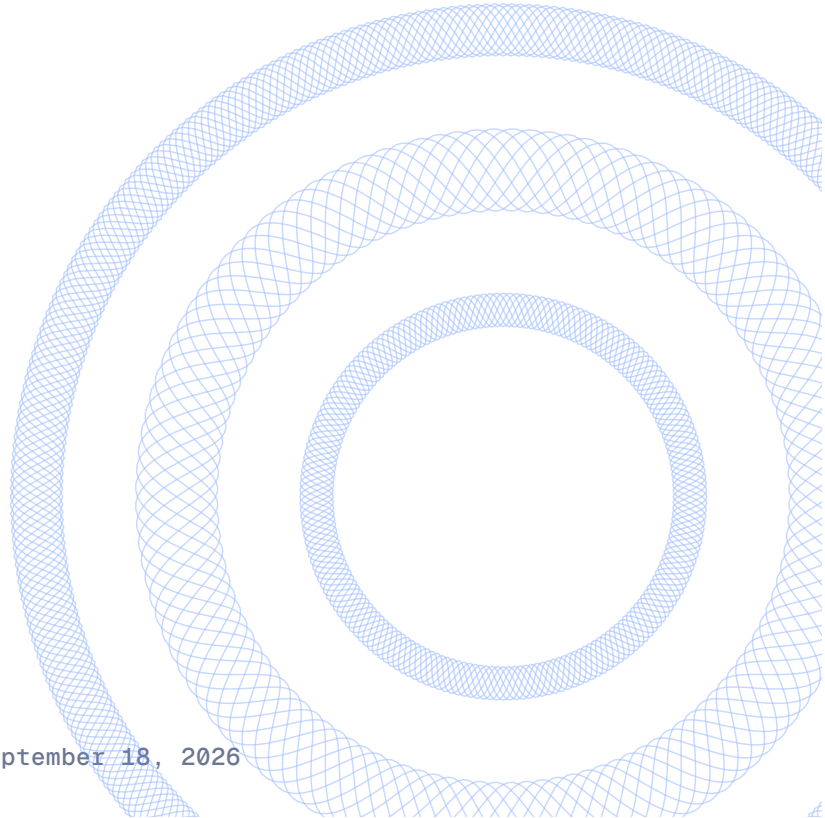
BoundID

Know Your Agent.

Identity and permissions for AI agents

Whitepaper and Technical Specification



Concept release · Version 1.1 · September 18, 2026
useboundid.com · x.com/UseBoundID

Contents

01

Executive summary

BoundID is a proposed identity and permissions service for AI agents operating on Robinhood Chain. Its guiding descriptor, Know Your Agent, expresses the goal: help applications identify who controls an agent, inspect the evidence behind that claim, and decide whether a specific action is authorized.

The proposed product family comprises BoundID Passport for agent identity and credentials, BoundID Verify for verification workflows and issuer checks, and BoundID API for policy evaluation and status access. Owner-signed permissions and public revocation records connect these products. Participating wallets and applications enforce the resulting permissions; identity documents remain off-chain.

The proposed token, provisionally $KYA, would support a later network of bonded verification providers. Its role is collateral for defined provider obligations, not a substitute for evidence or a guarantee of honest behavior. The core service can operate without a fungible token; token issuance should depend on demonstrated customer demand, a defensible security model, and legal review.

Recommended first release

- Build a testnet passport, wallet-control verification, signed permissions, revocation, and one guarded payment demonstration.
- Use named, approved credential issuers initially. Disclose this trust model and do not describe the pilot as permissionless verification.
- Keep sensitive identity evidence off-chain. Require an explicit policy decision for each protected action.
- Evaluate a public token only after the service and dispute process have been tested. Proposed economics appear in Sections 8 and 9.

Document status and scope

This concept release defines a proposed product and implementation plan. It does not announce a deployed service, issued token, completed audit, partnership, fundraising offer, or regulatory approval. Technical requirements and economic parameters require validation before adoption.

BoundID is an independent project concept. No affiliation with or endorsement by Robinhood, Chamath Palihapitiya, Ethereum, NIST, or any referenced organization is claimed.

02

Product purpose and initial use case

An agent may possess a valid wallet key while lacking authority to spend its owner's money. It may also retain an old credential after its operator changes, or present favorable history that says little about its current software. BoundID would expose these distinctions so relying applications can make narrower, better-supported decisions.

What the product establishes

The passport records identifiers and evidence. Authentication establishes control of a key. Authorization defines the actions an owner permits. Enforcement is the wallet, contract, or application control that actually blocks prohibited actions. A credential alone cannot enforce a spending limit, prevent prompt injection, or establish legal entitlement to trade an asset.

The first customers to approach are agent-wallet developers and business payment applications that already need to restrict automated spending. The initial commercial hypothesis is that these teams will pay for integration support, reliable status checks, and an auditable authorization history. Customer demand remains to be tested.

Pilot scenario

A fictional business, Acme Operations, authorizes an invoice-payment agent to send one test token to approved suppliers. The demonstration permits up to 100 test units per transaction and 500 per UTC day. Those values are illustrative testing limits, not a risk recommendation for real funds.

1. The owner registers the agent and proves control of the owner and execution wallets.
2. An approved issuer attests to the operator relationship. The owner signs the asset, recipients, budget, and expiration constraints.
3. The application checks the passport and policy. A guarded wallet rejects a transaction that exceeds the current budget or changes the approved recipient.
4. The owner revokes the delegation. Subsequent protected transactions fail even when the agent still holds its session key.

Why use a blockchain

Shared status, revocation records, and provider bonds may be useful when independently operated applications need the same evidence. A centralized database is simpler when one company controls every participant. The pilot must show that independent verification adds value worth the public metadata, gas costs, and operational complexity.

Outside the initial scope

The pilot does not offer brokerage, lending, custody, autonomous access to securities, a universal trust score, proof of model honesty, or insurance against agent losses. Human and business identity checks would be supplied by qualified providers where required, not replaced by a wallet signature.

03

Network and standards foundation

Robinhood describes its chain as a permissionless, Ethereum-compatible Layer 2. Its connection documentation identifies an Arbitrum-based network using ETH for gas and publishes mainnet and testnet settings. These facts support the proposed deployment target; they do not establish an integration or commercial relationship with BoundID. [1, 2]

NETWORK SETTING	MAINNET	TESTNET
Chain ID	4663	46630
Native gas asset	ETH	ETH
Public RPC	rpc.mainnet.chain.robinhood.com	rpc.testnet.chain.robinhood.com

RPC hosts above use HTTPS. The public endpoints are rate-limited and not recommended for production in the documentation. Recheck network settings, finality behavior, deployed addresses, and infrastructure support before any release. [2]

Standards treatment

REFERENCE	OBSERVED STATUS	PROPOSED BOUNDID USE
ERC-8004 [3]	Draft	Agent identifiers and registry interoperability. Pin a reviewed revision.
ERC-8126 [4]	Final	Later verification-provider compatibility target, not an MVP compliance claim.
W3C VC 2.0 [5]	Recommendation	Evaluate for portable issuer-signed credentials and presentations.
EIP-712 [7]	Final	Typed owner authorizations with application replay controls.
ERC-1271 [8]	Final	Validate signatures for supported smart contract wallets.

ERC-8004 describes identity, reputation, and validation registries; it does not prescribe KYA token economics. Its transferable identity requires careful treatment of ownership changes. BoundID would treat a transfer or relevant key change as a reason to invalidate prior owner-specific permissions. [3]

ERC-8126 includes specialized verification and privacy-proof requirements. Implementing a wallet check or publishing a score is insufficient to claim conformance. Full support would require a separate gap analysis, security review, and interoperability testing. [4]

NIST's February 2026 concept-paper announcement identifies software-agent identity and authorization as an area for further work. It supports the relevance of the problem, not certification of this design or a mandate to use blockchain. [6]

04

BoundID Passport and verification

BoundID Passport would present separate, dated claims with their issuers and limitations. There would be no single VERIFIED badge that implies every claim is current or every action is safe. Applications select the claims and issuers they are prepared to rely on.

PROFILE	REQUIRED EVIDENCE	WHAT IT DOES NOT ESTABLISH
Registered	Agent identifier and declared endpoint metadata.	Real-world identity, authority, or safety.
Control checked	Fresh signatures from relevant keys and an endpoint challenge.	The legal identity of the controller.
Operator attested	Accepted provider credential linking an operator to the current agent binding.	Permission for a particular transaction.
Policy eligible	Current operator claims plus an unexpired delegation that matches the requested action.	Execution unless the receiving system enforces the policy.

Passport fields

The public view would show the network and registry, agent identifier, current key binding, optional display name, claim type, issuer, verification method, issuance time, expiration, and status. It would also display the applicable policy version and any disclosed security assessment. Operator names would be disclosed only when authorized and necessary.

A model name or code hash is a scoped claim. A reviewed repository is not proof that the reviewed build is running. An endpoint response is not proof of exclusive control. Model or runtime verification must identify exactly what was measured, by whom, and when; stronger runtime attestation is a separate feature.

Credential lifecycle

Credential states are pending, active, suspended, revoked, and expired. Suspension blocks reliance while an issue is reviewed. Revocation is permanent for that credential; remediation requires a new issuance. Expiration is evaluated from the signed validity interval even if no transaction updates stored state.

Owner, execution-key, or relevant endpoint changes invalidate dependent claims until rechecked. A changed owner cannot inherit prior identity claims or permissions. Supported registry paths must increment a binding epoch atomically on every owner or key change, including transfers away and back. Event-indexer-only invalidation is insufficient; unsupported paths remain blocked.

Freshness and reputation

Each claim class has a maximum age chosen by the relying application. A proposed pilot default is a 30-day operator claim, a 24-hour delegation, and a decision receipt valid for at most 60 seconds. Revocation always overrides those windows. These are test parameters, not production assurances.

Later reputation features should expose outcomes, sample size, observation period, and dispute status. Missing history remains unknown. A number such as 97 percent must never be presented as a probability of safety without a validated statistical basis.

05

System architecture and contracts

The MVP would connect BoundID Passport, BoundID Verify, and BoundID API through a web console, named issuers, public status contracts, and one guarded execution integration. The following component names are proposed implementation names, not existing deployments.

COMPONENT	RESPONSIBILITY	RELEASE
IdentityAdapter	Resolve a pinned agent registry and its current owner and wallet binding.	MVP
IssuerRegistry	Record issuer keys, permitted claim types, key epochs, and suspension state.	MVP
CredentialStatusRegistry	Anchor opaque credential IDs and public status without identity evidence.	MVP
DelegationRegistry	Record owner-authorized policy hashes, revocations, and binding versions.	MVP
ExecutionGuard	Validate permissions and maintain atomic spend and replay protection.	MVP
ProviderBondVault	Lock provider collateral and reserve amounts for unresolved obligations.	Later
DisputeManager	Apply adjudicated outcomes and auditable, bounded penalties.	Later
KYAToken and Vesting	Implement the separately approved token and allocation schedule.	Conditional

Services outside the blockchain

A verification worker obtains issuer evidence; an encrypted evidence store holds permitted records; an indexer follows status events; and a policy service produces explanations and signed evaluation receipts. A TypeScript SDK would let partners verify supported receipts and retrieve current public status. Infrastructure choices remain replaceable behind versioned interfaces.

A proposed implementation stack is Solidity with invariant testing for contracts, a TypeScript API and SDK, a relational database, encrypted object storage, a durable job queue, and a React console. Exact package versions would be pinned after a dependency and security review.

Deployment and trust boundaries

Use existing registry deployments only after verifying provenance and compatibility. Otherwise deploy a test registry explicitly managed by BoundID; never label it canonical. Publish contract addresses and bytecode hashes only after deployment and verification.

Prefer immutable, versioned core contracts with explicit migration over hidden upgrades. Administrative actions use separately controlled keys. An emergency pause may stop new issuance or execution, but must preserve the ability to publish revocations. The MVP relies on selected issuers and operators; economic decentralization is a later objective.

06

Data model and privacy

BoundID would store only the minimum public information needed for independent status checks. Identity records, biometrics, legal documents, secrets, private prompts, and customer transaction narratives must not be placed in public storage, transaction calldata, or event logs.

RECORD	CORE FIELDS	STORAGE
Agent reference	chainId, registry address, agentId, bindingVersion	Public
Issuer entry	issuer ID, key, keyEpoch, claim scope, status	Public
Credential status	random credentialId, issuer, status, expiry	Public minimum
Signed credential	subject binding, claim, method, evidence reference, validity, signature	Private by default
Delegation	owner, agent key, chain, policy hash, version, expiry, revocation state	Public minimum
Policy constraints	asset, targets, actions, amounts, time window, audience, nonce rules	Public for on-chain enforcement
Evidence record	provider case ID, encrypted evidence, access log, retention class	Off-chain restricted
Decision receipt	request hash, decision, reason codes, policy version, block reference, expiry	Authorized relying party

Binding and integrity

Every credential and delegation must identify its intended subject and current owner/key binding. An agentId alone is insufficient because it can collide across registries and networks. Signed data must include the schema version and relevant network, registry, issuer, and application domains.

Avoid publishing a plain hash of an identity document or other guessable personal data. Prefer a random identifier. If evidence integrity requires a commitment, use a reviewed, high-entropy randomized construction and keep its opening material private. A hash does not make personal information anonymous.

Access and retention

Collect evidence directly through a qualified provider where possible. Disclose public metadata and correlation risks, including transaction constraints submitted to an on-chain guard. Such constraints are not private in this MVP. Restrict private records by role, encrypt them with managed keys, log access, and set appropriate deletion schedules. Revocation does not erase public-chain history.

Portable credentials may use a reviewed W3C Verifiable Credentials profile. That model provides a framework for issuer claims and presentations, not proof that the claims are true. Selective disclosure or zero-knowledge proofs require a separately specified and tested cryptographic implementation. [5]

07

Authorization and BoundID API

BoundID API answers a scoped question: does this request satisfy this relying party's policy using currently acceptable evidence? Its result is an evaluation, not a blanket warranty. On-chain execution must enforce the same constraints at the moment of execution.

Protected transaction flow

1. Resolve the full agent reference and compare current owner, wallet, and binding version with the supplied credentials.
2. Verify issuer and owner signatures, key epochs, claim scope, expiration, and revocation. Use typed signing and support reviewed contract-wallet validation. [7, 8]
3. Apply the relying party's accepted-issuer rules and evaluate the requested target, function, asset, amount, and time window.
4. Return ALLOW, DENY, or INDETERMINATE with reason codes. Missing evidence or unavailable status must not become an implicit ALLOW.
5. At execution, the guard rechecks enforceable current state, binds the authorization to the exact transaction, consumes its nonce, and updates the spend counter atomically.

For private claims, the guard relies on an approved attestation signer and checks the associated status. It does not independently establish the truth of the underlying identity evidence.

Proposed BoundID API surface

OPERATION	PURPOSE
POST /v1/agents	Start registration and return a wallet challenge.
POST /v1/verifications	Request an identified, scoped issuer check.
GET /v1/passports/{agentRef}	Return public claims and current status.
POST /v1/evaluations	Evaluate a signed request against a named policy.
POST /v1/revocations	Submit an authenticated revocation intent.

An evaluation receipt would include the following fields. Identifiers and hashes are omitted in this illustrative shape; the signed production payload must contain them.

```
decision: ALLOW | DENY | INDETERMINATE
reasonCodes: [POLICY_MATCH | EXPIRED | REVOKED | ...]
requestHash, policyVersion, agentBindingVersion
issuerSetVersion, evaluatedBlockHash, evaluatedAt
validUntil, audience, nonce, signerKeyId, signature
```

A proposed receipt lifetime is at most 60 seconds. It must bind the chain, guard, caller, recipient, asset, value, and full call data. Replays, altered parameters, unknown selectors, unsupported batch calls, and failed freshness checks are rejected. Daily budgets use token base units and atomic counters; concurrent requests cannot each spend the same remaining balance.

Proposed token design and allocation

$KYA remains the provisional ticker for BoundID's proposed ERC-20 token on Robinhood Chain. ETH remains the gas asset. The token would primarily bond verification providers and, if adopted, support dispute deposits. Users would not need to buy it merely to read a passport. No token is represented as issued by this document.

Working economic specification

For modeling, propose a maximum supply of 1,000,000,000 KYA with 18 decimals and no discretionary minting after the initial allocation. This is a denomination and governance choice, not a valuation. The name, ticker, supply, distribution, and legal structure require approval before any deployment or offer.

ALLOCATION	SHARE	KYA	PROPOSED RELEASE DISCIPLINE
Ecosystem and provider development	35%	350,000,000	Milestone-based releases over at least 48 months.
Protocol treasury	20%	200,000,000	Published budgets over at least 36 months.
Core contributors	20%	200,000,000	48 months total with a 12-month cliff.
Strategic contributors	10%	100,000,000	48 months total with a 12-month cliff.
Community distribution reserve	10%	100,000,000	Rules and eligibility set only after legal review.
Liquidity reserve	5%	50,000,000	No automatic deployment or price-support promise.
Total	100%	1,000,000,000	All allocations are provisional.

For contributor allocations, the proposed schedule releases 25 percent at month 12 and the remainder monthly through month 48. The approved launch date would start the schedule. Reserves are not assumed to circulate immediately. Publish allocation wallets, unlock schedules, and circulating-supply methodology before distribution.

Fees and economic limits

Charge customers in a stable denomination for completed verification work and managed BoundID API service. Approved settlement assets must be identified by network and contract address; the MVP can use a mock token. Providers may earn those fees for work performed. Holding or staking KYA alone would not entitle holders to dividends or a promised annual yield.

No mandatory burn, automatic buyback, price target, or guaranteed appreciation is proposed. A native token is not technically necessary for identity verification. If a stable-asset bond or ordinary contractual accountability works better, the protocol should retain the option not to launch KYA.

09

Provider bonding and disputes

A bond is money a verification provider can lose for a specifically defined breach. It does not prove an off-chain claim is correct. The network therefore needs evidence standards, independent adjudication, and

enforceable limits before accepting bonds as a meaningful security control.

Proposed provider cycle

1. Admit a provider for particular claim classes after reviewing its methods, key custody, conflicts, and evidence-retention arrangements.
2. Lock collateral and reserve a defined exposure amount before accepting a verification job. One balance cannot secure unlimited simultaneous obligations.
3. Quote the fee and issue a signed, scoped credential after the required checks. Pay for completed work under published terms; do not reward raw issuance volume.
4. Permit evidence-backed challenges. Freeze only the affected reserved collateral while notice, response, and appeal procedures run.
5. Release unencumbered collateral only after its obligations and challenge windows expire. Provider exit must not erase outstanding obligations.

What could justify a penalty

Candidates include cryptographically provable contradictory attestations under a defined rule, falsified required evidence, or documented omission of a mandatory verification step. An agent making a bad trade or later suffering a compromise does not by itself prove provider misconduct. Vague standards such as negligent verification must be made specific before funds are slashable.

For a pilot dispute process, propose a seven-day challenge period, seven-day provider response period, and fourteen-day appeal period. A challenged job's collateral stays reserved until final resolution. Unchallenged collateral remains locked through the claim's validity plus the challenge period. These windows require operational and legal testing.

Who decides and where funds go

The initial process would use a disclosed panel of three independent reviewers, with two required for a finding and conflicted reviewers recused. An appeal uses a separate panel. Publish evidence-handling rules, maximum penalties, decision reasons with sensitive details removed, and the treatment of unresolved cases.

Return successful challengers' deposits and pay only capped investigation costs under the published schedule. Route remaining penalties to a separately governed remediation reserve. That reserve is not insurance and does not guarantee reimbursement of downstream losses. A rejected good-faith challenge is not automatically punishable.

Collateral risk

A falling KYA price could weaken the bond precisely when misconduct increases. Stress-test severe price declines, correlated claims, provider concentration, and illiquid markets. Set conservative workload caps, reviewed valuation rules, and top-up or suspension procedures. Do not launch bonding if adequate collateral cannot be measured or maintained; stable collateral or a tokenless service remains an alternative.

Application screens and commercial model

The interface should make evidence and permissions understandable to a developer or business operator. It must consistently distinguish declared information, verified claims, active delegation, and actual execution controls.

SCREEN	PRIMARY ACTION	REQUIRED INFORMATION
Project overview	Register or inspect an agent.	Credential status, approaching expirations, current bindings.
BoundID Passport	Inspect individual claims.	Issuer, method, scope, last check, expiry, limitations.
Permissions	Create or revoke a delegation.	Asset, destinations, actions, limits, duration, enforcement path.
BoundID API	Integrate and test in the developer console.	Scoped keys, usage, examples, reason codes, webhook delivery.
BoundID Verify	Review evidence and issue claims in the issuer workspace.	Assigned scope, evidence checklist, key identity, review history.
Incident console	Suspend or revoke safely.	Affected subjects, authorized actor, audit trail, recovery steps.

Public website

Lead with BoundID and the line “Identity and permissions for AI agents,” with Know Your Agent as the descriptor. Explain the products, label testnet examples, document issuer trust assumptions, and publish verified contract addresses when available. Publish only substantiated partners, audits, usage figures, token prices, and exchange listings; enable purchases only after an approved launch.

Who pays and for what

Agent developers and platforms would pay for verification workflows, managed status access, monitoring, and support. Identity providers could charge their underlying human or business verification fees separately. The public registry remains independently readable; BoundID would charge for convenience, reliability, private workflows, and contracted service levels rather than claiming exclusive ownership of public data.

Test three commercial structures with prospective design partners: a per-verification charge, a monthly BoundID API plan with metered usage, and an enterprise integration agreement. Do not publish prices until provider costs, support effort, and retention requirements have been measured.

Commercial evidence to gather

- Which transaction is currently blocked or too risky without this evidence?
- Which issuer claims will the customer actually rely on, and what liability terms are acceptable?
- Does reusable public status reduce integration work compared with a private API?
- Would the customer pay without token incentives or a token listing?

The competitive advantage to prove is usable delegation and revocation across independent applications. A familiar ticker or a passport graphic is not evidence of product demand.

11

Implementation roadmap and acceptance tests

The roadmap is milestone-based. An indicative MVP effort is 12 to 16 weeks after staffing and scope approval, assuming an experienced small engineering team and access to verification providers. This is a planning assumption, not a delivery commitment. External audits, legal review, and procurement may take longer.

Phase 1 · Define the pilot

Confirm one customer workflow, obtain design-partner feedback, choose the first issuer and claim class, and write the threat model. Finalize the public/private data boundary, policy schema, revocation semantics, and incident ownership. Exit when a partner accepts the proposed evidence and integration requirements.

Phase 2 · Build on testnet

Implement the registry adapters, issuer and credential status, owner delegation, API, SDK, console, and one guarded wallet demonstration. Use test assets only. Exit when registration, verification, permitted execution, denied execution, expiration, and revocation all work end to end.

Phase 3 · Run a restricted service pilot

Obtain independent security review, close material findings, test recovery and provider outages, and secure applicable legal and privacy approvals. Start with approved users, limited exposure, and an explicit support process. Measure real operational costs before setting prices.

Phase 4 · Decide whether to add a token

Evaluate whether multiple independent providers need bonded participation. Complete economic simulation, dispute rules, token review, vesting contracts, and public disclosures. Launch only if the evidence supports it. A functioning service with no public token remains a valid outcome.

ACCEPTANCE TEST	REQUIRED RESULT
Binding changes	Old owner or key credentials cannot authorize new protected actions.
Replay and mutation	Reused nonces, other chains, altered recipients, and changed call data fail.
Concurrent spending	Two simultaneous requests cannot overspend one shared budget.
Revocation and outages	Revoked or stale evidence blocks execution; unavailable evidence never defaults to approval.
Privacy and access	No prohibited personal evidence reaches public state or unauthorized API callers.

Proposed pilot measurements: three active design partners, at least 100 end-to-end test transactions, all negative-path tests passing, and no unresolved critical or high-severity security findings. Measure decision latency and revocation propagation at the 95th percentile. An initial target is under 500 ms for cached API

evaluation and under 60 seconds for index updates after the configured confirmation threshold; these are not chain-finality or uptime guarantees.

12

Security and operational assurance

BoundID should assume that agents can be compromised, issuers can make mistakes, and infrastructure can become unavailable. The design must limit the consequences of those events and expose uncertainty to relying applications.

RISK	REQUIRED CONTROL	RESIDUAL LIMITATION
Issuer key compromise	Protected signing keys, key epochs, rapid issuer suspension, reissuance plan.	A false claim can be accepted before detection.
Prompt injection or malicious agent	Least privilege, guarded calls, exact intent binding, independent budgets.	Identity checks cannot make model behavior safe.
Guard bypass	Restrict approvals and alternate execution paths; reject unsupported call forms.	Controls cover only integrated execution paths.
Stale data or chain outage	Freshness ceilings, current-state checks, RPC redundancy, fail-closed policy.	Availability decreases when evidence cannot be trusted.
Sybil feedback or colluding providers	Evidence-linked outcomes, issuer diversity, concentration limits, review.	Reputation remains imperfect and gameable.
Contract or admin failure	Invariant tests, independent review, bounded roles, migration and pause drills.	Audits and multisignatures cannot eliminate all risk.

Transaction safety

Guard all relevant transaction routes, including batch calls, approval changes, delegate calls, and session-key operations. Restrict the pilot to a reviewed allowlist of simple calls. Any contract upgrade or alternative signing route that can bypass the guard must be explicitly excluded or separately controlled.

Use per-asset base-unit budgets in the MVP. A general dollar-denominated limit would require reviewed price sources, staleness checks, and manipulation defenses. Test unusual token behavior and fee-on-transfer assets before allowing them; unsupported assets remain blocked.

Operations and incident response

Separate verification signing, application administration, and treasury controls. Record privileged actions, rehearse key rotation and issuer suspension, maintain backups, and assign an on-call owner. A detected compromise should trigger containment, dependent-credential suspension, customer notification, evidence preservation, and reviewed recovery.

Document the distinction between transaction inclusion and the confirmation level accepted by the relying application. Store block references, reconcile chain reorganizations, invalidate affected cached decisions, and stop protected actions when freshness cannot be established. A signed API receipt does not override later revocation.

13

Governance and launch requirements

Accountability before token governance

The initial operator would publish who can admit issuers, change policy schemas, pause issuance, deploy new contract versions, and move treasury funds. A proposed starting control is a three-of-five administrator multisignature with a 48-hour delay for ordinary changes. Emergency suspension and revocation require a narrower, documented path. These arrangements are proposed, not established.

Token voting, if introduced, should not reveal private identity records or adjudicate individual verification disputes by popularity. Publish conflicts policies and change notices, and allow integrators to reject new policy versions. Material migrations require explicit customer review and a clear treatment of outstanding credentials and bonds.

Required approvals before public release

- Confirm the operating entity, responsible individuals, ownership of intellectual property, and name and ticker availability.
- Have qualified counsel assess token issuance, marketing, distribution, applicable financial-services obligations, sanctions controls, and each intended jurisdiction.
- Complete privacy review covering public metadata, provider contracts, retention, deletion, cross-border handling, and customer disclosures.
- Publish the threat model, independent security-review scope and results, unresolved limitations, deployed addresses, administrator powers, and incident contact.
- Approve commercial terms covering evidence reliance, provider duties, challenges, remedies, service availability, and limits of responsibility.
- If a token is approved, publish supply controls, allocations, unlocks, conflicts, treasury authority, and distribution rules before any sale or promotion.

Open decisions

Before engineering commitments, select the first buyer, claim provider, supported wallet, and protected action. Before bonding, decide collateral valuation, exposure caps, objective penalty rules, and reviewer independence. Before token issuance, approve the economic case and distribution structure. None of these decisions should be inferred from the illustrative allocation table.

Important notice

This paper describes a proposed technology project. It is not an offer to sell, a solicitation to buy, a prospectus, or investment, legal, or tax advice. No return, liquidity, exchange listing, asset eligibility, regulatory status, or future token value is promised. Calling an asset a utility token does not determine its legal treatment.

BoundID credentials would provide scoped evidence for a relying party's decision. They would not replace that party's legal duties, guarantee the conduct of an agent or operator, or authorize regulated activity by themselves. Any third-party relationship must be independently agreed and announced. Publication of this paper does not establish one.

14

References and technical source notes

Primary sources reviewed September 18, 2026. Numbered citations identify external facts; uncited product requirements, economic terms, timelines, and targets are BoundID proposals. Standards and network documentation may change. Pin reviewed versions during implementation and recheck them before release.

[1] **About Robinhood Chain** · Robinhood Chain documentation
Supports the description of the network and its stated focus on financial infrastructure.

[2] **Connecting to Robinhood Chain** · Robinhood Chain documentation
Supports chain IDs, ETH gas, public RPC hosts, and the public-endpoint production caveat.

[3] **ERC 8004 Trustless Agents** · Ethereum Improvement Proposals
Displayed status: Draft. Supports registry concepts and identity transfer behavior; not proof of a BoundID deployment or audit.

[4] **ERC 8126 AI Agent Verification** · Ethereum Improvement Proposals
Displayed status: Final. A later compatibility target with broader requirements than the proposed MVP.

[5] **Verifiable Credentials Data Model v2.0** · World Wide Web Consortium
W3C Recommendation dated May 15, 2025. Reference for the credential data model, not a verification service.

[6] **New Concept Paper on Identity and Authority of Software Agents** · National Institute of Standards and Technology
Announcement dated February 5, 2026. Context for agent identity and authorization research; no BoundID endorsement.

[7] **EIP 712 Typed structured data hashing and signing** · Ethereum Improvement Proposals
Reference for typed signatures. Nonces, expiration, audience binding, and application replay prevention remain implementation duties.

[8] **ERC 1271 Standard Signature Validation Method for Contracts** · Ethereum Improvement Proposals
Reference for signature validation by supported smart contract wallets.