

COVARA

Launch openly. Move privately.

A privacy focused launchpad
and capital workspace

Designed around Sigilo. Targeting Robinhood Chain.

TECHNICAL & PRODUCT WHITEPAPER

Version 0.1 / 23 October 2026

Proposed architecture with a local simulated interface.

No COVARA contracts or token deployed. No audit or endorsement implied.

Working brand; clearance pending. No platform token required initially.

Abstract

COVARA proposes a common workspace for shielded ETH transfers, token launches with inspectable economics, trading through bounded public routes, and privately owned fee entitlements. It starts from the payment primitives developed in Sigilo and treats every extension as a separate protocol change requiring its own review.

The central design principle is selective visibility. A participant should be able to avoid publishing an entire financial history simply to receive an allocation or a contributor payment. A market should still be able to inspect token supply, issuer powers, sale terms, public liquidity and fee policy. COVARA does not equate private ownership with private market execution. Public deposits, withdrawals, escrow movements and DEX swaps remain observable.

The target network is Robinhood Chain. The initial product does not require a native COVARA token. Delivery starts with recoverable shielded ETH transfers and proceeds through launch settlement, constrained trading and private fee claims.

1. Thesis

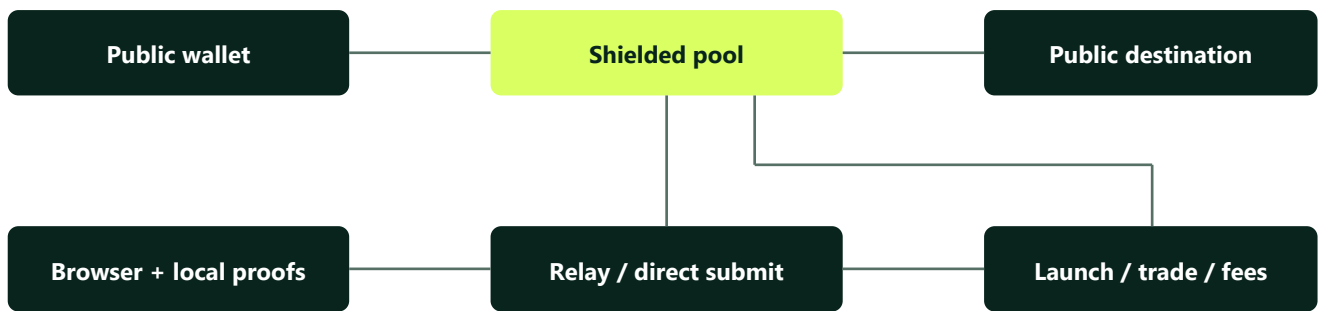
Launch infrastructure connects founders, participants, liquidity and ongoing revenue. Today, each step can create a public identity link. A user's source wallet, allocation wallet, trading activity and compensation destination can be correlated even when the application does not need those relationships.

COVARA's intended contribution is a consistent permission and disclosure model across those workflows. A private account is controlled by local keys. An action specifies what value moves, what authority is exercised, what the public chain sees and what any service learns. A protocol proof enforces the hidden part; an ordinary contract enforces the public market rules. Neither the website nor an operator's promise can replace those guarantees.

"Privacy launches" means improving privacy for launch participants and beneficiaries. It does not mean hiding issuer controls, supply, reserves, conflicts of interest or the terms buyers need to understand.

2. System and actors

The system consists of a browser wallet/prover, supported public wallets, versioned shielded pools, launch escrow/factory contracts, constrained trade adapters, fee-epoch contracts, an indexer, quote service and gas-paying relay. Recipients scan encrypted note data and prove ownership locally. Founders publish launch terms. Relayers submit approved transactions but do not receive users' spending keys. Indexers reconstruct public state and are independently replaceable.



Public inputs and outputs remain visible. New modules need independent review.

Indexer reconstructs public events; private keys and witnesses stay on the user device.

The diagram is a target design. In particular, launch, trading and private fee adapters are not present merely because Sigilo has a transfer circuit.

3. Shielded value

A shielded note represents a value owned by a key. A conceptual new note commitment can be written as:

$$C = H(\text{protocolDomain}, \text{assetId}, \text{value}, \text{ownerKeyCommitment}, \text{randomness})$$

A spend reveals a deterministic nullifier derived from the note and the owner's secret. A valid proof establishes authorization, membership in a recognized commitment tree, value range, conservation and binding to the intended action. The contract rejects already used nullifiers. Output commitments and encrypted note payloads allow recipients to discover new value without publishing the plaintext owner/value relationship.

For each asset in an ordinary transfer:

$$\text{sum}(\text{input notes}) + \text{public deposit} = \text{sum}(\text{output notes}) + \text{public withdrawal} + \text{explicit fees}$$

All values use bounded integers in the asset's smallest unit. The proof must prevent field wraparound and cross-asset balancing. A multi-asset pool requires asset identifiers and corresponding contract custody accounting; adding an ERC-20 selector to a single-asset pool is insufficient.

An action authorization must bind network, pool address/version, operation, destination, amount limits, fee, deadline/nonce where used, and encrypted output payloads. Reusing a proof across a chain, pool, operation, or changed recipient must fail. The precise binding schema is a circuit/contract design task, pinned by hashes and reproducible artifacts.

Sigilo's EVM source offers useful reference mechanisms for local proving, encrypted notes, nullifiers, external-context binding and relay quotes. Its code and security documents explicitly describe unaudited/development assumptions, and deployment documentation is inconsistent. COVARA therefore imports neither a security certification nor a claim of current production readiness. Pin and review the actual source and proving artifacts before reuse. See architecture.

4. Transfer lifecycle and recovery

Shielding is public: a public wallet transfers ETH to a pool and creates a commitment. Within the shielded system, a transfer spends owned notes and creates receiver/change notes. Withdrawing exposes a public destination and amount. The public transaction sender may be a relayer, but the service can still observe timing and IP information unless the transport and operations design mitigate it.

The user must be able to reconstruct state from encrypted note data and canonical chain events after device loss. An indexer cache is not the authoritative ledger. Backups must identify network, protocol version and scan start point as well as key material. Loss of the spending/recovery secret can permanently prevent spending; the backend has no hidden recovery key.

An ambiguous submission creates a reconciliation task, not a fresh payment. The client tracks the intent, nullifiers and transaction identifier, and reconciles after disconnects or reorgs. Pending outputs are not presented as final spendable balance. Finality is a configured chain-specific policy distinguishing sequencer inclusion from stronger settlement assurance.

5. Launch design

Public market terms

The first launch template is a fixed-price ETH sale of a supported standard token. Creation commits supply, sale allocation, issuer permissions, vesting, price, caps, deadlines, liquidity policy, fee schedule and refund rules. The factory emits a terms hash so the frontend can verify that the displayed terms are those being executed. A mutable website must not change an immutable sale.

Contributions cannot exceed the hard cap. A race for the remaining allocation results in one accepted transaction and a reverted stale contribution; initial scope avoids partial fills. A successful close permits settlement under the fixed terms. A failed soft-cap close or valid pre-contribution cancellation follows explicit refund rules. Once funds exist, the issuer cannot invent new cancellation penalties or replace recipients. Settlement and refund execution should be permissionless after the required state transition.

The invariant is that accepted principal is completely accounted for as refundable liability, sale proceeds, disclosed liquidity reserve and successful-sale fee. Allocation inventory must similarly reconcile against supply and sale terms. Funds are never available for general treasury spending while still owed as refunds or claims.

Participant privacy

Phase 2A can publish a conventional escrow sale with explicit public amounts. Shielded funding does not hide the amount of the visible pool-to-escrow transfer. A receipt commitment can obscure who controls an allocation, but a unique contribution amount or timing may still correlate it.

Phase 2B requires a new protocol for privately owned allocation/refund entitlements. Proofs must bind a receipt to its launch, accepted principal, allocation formula, asset, owner and one-time claim nullifier. Success-token claims and failed-sale refunds must be mutually exclusive for the same entitlement. Receiving the token into a shielded pool additionally requires reviewed supported-token custody and proofs. A conventional ERC-20 claim to a public address is explicitly public.

Hiding individual contribution amounts while publishing only aggregate funding requires batching/aggregation and an additional confidentiality construction. That is research beyond the baseline private-beneficiary design. COVARA will not describe a public escrow event as an amount-private contribution.

6. Trading design

The initial trade path uses one allowlisted venue and a small supported-asset set. Exact-input orders bind the adapter, route, input, minimum output, receiver, deadline and service fee. Contracts reject unapproved routers, stale authorization, unsupported token behavior and arbitrary callbacks.

The proposed atomic path spends shielded input, deducts the disclosed input service fee, exposes the net amount to a public swap, and returns measured output into supported shielded custody. A conventional precomputed fixed-output proof cannot simply predict a DEX's actual output. A new reviewed construction must bind that measured value to valid output notes before the route can ship. Any adapter step that cannot complete must revert the entire onchain operation, including protocol fee collection and note consumption. An offchain database status cannot provide that atomicity. Multi-transaction fallback routes require a separate design with explicit custody and failure handling; they are not silently used in the private flow.

The public DEX still reveals the pair, executed amounts, price, route and transaction timing. The objective is to reduce direct linkage to the user's complete shielded balance, not to conceal the trade from the exchange or chain. MEV, sequencer visibility, ordering and thin liquidity remain material. A private relay does not establish confidential execution or a guaranteed price.

7. Private fee beneficiaries

Fees can originate from COVARA launches, routed trades or separately integrated creator-fee systems. A source can be redirected only by its authorized controller and only within that source's rules. COVARA cannot redirect arbitrary third-party fees by software convention.

The proposed model uses funded epochs. An epoch commits its source, asset, received total, policy version and recipient-entitlement root. A recipient proves entitlement and authorizes a shielded destination without publishing the private beneficiary key. A claim nullifier prevents repeated withdrawal. Accounting requires:

epoch funding = claimed value + unclaimed liability + explicitly tracked rounding remainder

Publishing an arbitrary operator-created Merkle root is not sufficient. Root construction must either be verified by a new proof against the committed policy and funded total, or clearly treated as a trusted distributor model with different guarantees. COVARA's target is proof-enforced conservation and policy adherence. A proposed policy construction partitions 10,000 basis points into committed contiguous recipient intervals, with entitlement equal to the difference of rounded cumulative shares. This allocates the exact total independently of claim order. Recipients need available encrypted policy/witness data before activation; a missing policy owner must not prevent claims on a funded, sealed epoch. The construction and its data-availability and recovery properties require dedicated review.

An onchain deposit for each individual recipient reveals that payout amount. To keep entitlements confidential, the design must fund an aggregate epoch and convert hidden entitlement liabilities into notes within shared custody using a verified conservation relation. A public vault-to-pool token transfer per claim

would disclose the amount. Public weights can allow observers to calculate recipient shares from total funding; a one-recipient epoch offers no amount ambiguity. Source revenue and aggregate deposits remain visible in all cases.

Policy changes apply prospectively. Past accrued rights cannot be redirected by changing the current recipient list. Recipients need authenticated receive-address registration, replay-resistant control proofs and an explicit recovery/rotation policy. "Anyone privately" means eligible recipients with compatible private addresses, not removal of source authorization or operating obligations.

8. Threat model and disclosure

Adversaries include public-chain analysts, dishonest or compromised relayers/indexers, malicious founders/tokens/routers, hostile browser extensions, compromised dependencies, network observers and colluding participants. COVARA seeks to prevent unauthorized spending, forged balances, duplicate claims, arbitrary route execution and unnecessary public owner linkage. It cannot promise privacy on a compromised device, secrecy of public events, or immunity to timing/amount inference.

Observer	Can normally observe	Intended hidden information / limitation
Public chain	Deposits/withdrawals, commitments/nullifiers, public launch events, DEX execution, fee totals	Note plaintext ownership/value, only where proof/encryption construction supports it
Relayer/RPC	Requests, timing, IP/transport metadata, transaction public data	Spending secret and private witness should remain local
Counterparty	Its received amount and relationship with sender	No automatic access to sender's other notes
User's device	Keys, witnesses, plaintext balances, selected routes	Device compromise defeats local confidentiality
Launch operator	Public terms/state and operational metadata	Private beneficiary secrets should not be needed
Optional auditor	User-selected receipts/disclosures	Scope depends on export; full viewing keys can reveal much more

No master viewing key is proposed. Selective disclosure begins with narrow receipts generated by the user and previewed before export. A receipt may itself disclose a relationship; its scope must be clear. Calling a viewing key "read only" does not make the revealed history harmless.

Operations must minimize logging, avoid app session replay, separate public analytics from financial data, and make indexer reconstruction possible. A small shielded pool can have weak practical anonymity even when its cryptography is correct. Do not advertise an "anonymity score" without a reviewed measurement model.

9. Governance and economics

Use versioned immutable pool deployments where practical, with explicit opt-in migrations. Upgradeable launch/adaptor components, if chosen, must publish who can change them, delay periods and user exit rights. Administrative pause authority must be specific and testable; there is no generic promise that all funds can always be safely frozen or recovered. Changes to cryptography or custody are new releases.

Proposed service fees are 50 bps on successful launch proceeds and 20 bps on routed trade input. Transfer and fee-distribution protocol fees start at zero, with explicit network/relay costs. These are planning hypotheses. No platform token is needed initially, and this paper creates no holder revenue right. See economics.

10. Deployment and roadmap

Robinhood's primary documentation lists mainnet chain ID 4663 and testnet 46630, with ETH as gas currency. COVARA must independently verify network identity, deployed bytecode and supported venue behavior at release time. Robinhood Wallet compatibility is an integration target; deployment on Chain does not establish a retail exchange listing or endorsement. [Official network configuration](#).

Delivery order is transfer MVP, launch escrow/claims, public-execution trade adapters, and private fee claims. Gates include reproducible artifacts, cryptographic review, contract invariants, restoration and reorg tests, monitored capped beta and a signed deployment manifest. Dates are staffing estimates contingent on these gates, not promises.

11. Prior work and evidence

Commitment trees, nullifiers and shielded note ownership are established privacy-system concepts; the [Zcash protocol specification](#) provides one primary reference. COVARA does not claim protocol compatibility with Zcash or inherit its security analysis. Existing Sigilo source is the concrete local starting point. The source register records the reviewed files, external sources and uncertainty.

The next technical publication should contain the actual circuit statement, typed public inputs, custody ledger, exact contract state machine, proving-artifact provenance, adversary assumptions and independent review results. Until then, the strongest honest description is **a specified product and proposed architecture with a simulated interface**.

Source document: docs/WHITEPAPER.md. The complete blueprint includes product requirements, engineering design, privacy/security review criteria, economics, user and developer guides, and a staged launch roadmap.