SHADE: A Privacy-Preserving AI Agent for Token Discovery on Base

The SHADE Project

shadeonbase.com · dapp.shadeonbase.com · docs.shadeonbase.com
x.com/shadeonbase · t.me/shadeonbase · github.com/shadeonbase

Abstract

A privacy-preserving artificial intelligence agent for token discovery on Base would allow traders to receive real-time launch intelligence without exposing their wallet identity, research history, or trade attribution to the public ledger or to the agent operator itself. The current generation of Base agents—Bankr, Aixbt, Clanker bots—logs every wallet, broadcasts every query, and links every trade to a known address, making whales easy to copytrade and alpha hunters trivial to front-run. The proposed solution scores every new token launch on Clanker, Flaunch, Bankr, and Zora using on-chain and social signals, then delivers ranked alpha through a four-primitive privacy stack: zero-knowledge proofs of token ownership (Semaphore over BN254), end-to-end encrypted messaging (XMTP using X3DH and Double Ratchet), trusted-execution-environment inference (Intel SGX / AMD SEV via Phala) with remote attestation, and privacy-routed trade execution (Railgun shielded pools). The protocol intentionally hides only what can honestly be hidden on a public chain—identity, queries, and trade attribution—while leaving on-chain settlement public. The native token, $SHADE, gates access via a Merkle-tree commitment scheme; subscription fees auto-buy $SHADE on the open market and fund a weekly reward pool for top performers.

1. Introduction

The Base ecosystem has become the dominant venue for permissionless token issuance in crypto. Clanker alone has processed more than $7 billion in cumulative trading volume since launch, and the Farcaster acquisition of Clanker in late 2025 sent the $CLANKER token up over 360 percent in a single week. Bankr, the conversational trading agent for X and Farcaster, generates millions of dollars in fees each month. Hundreds of new tokens are deployed every day across Clanker, Flaunch, Bankr, and Zora.

Two problems remain unsolved.

First, no human can read 500 launches per day. Traders either miss alpha entirely or buy into rugs because the speed of deployment exceeds the speed of human due diligence.

Second, every existing AI agent on Base operates as a public surveillance layer. When a user queries Bankr, that query and the originating wallet are logged. When a user follows Aixbt, their identity is linked to the calls they act on. When a whale researches a token, the research itself becomes a market signal that copytraders exploit.

These problems compound. The more public Base trading becomes, the more urgently sophisticated traders need privacy—and the less any of them are getting it.

SHADE addresses both problems with a single product: an AI agent that scores every Base launch in real time and delivers the result through a privacy stack that hides who the user is, what they research, and where their trades come from—while remaining honest that on-chain trades themselves cannot be hidden on a public chain.

2. The Problem

2.1 Signal overload

The Base launchpad ecosystem produces more discoverable assets per day than any prior environment in crypto. A retail trader monitoring Farcaster, X, and three launchpad feeds simultaneously will see a new candidate token roughly every two to three minutes during peak hours. Each candidate requires the same baseline diligence: liquidity check, holder distribution check, dev wallet history check, contract bytecode check, and social-context check.

In aggregate this work is impossible for an unaided human. The trader either skips diligence and gets rugged, or skips opportunities and watches others print.

2.2 Privacy failures in current agents

Every Base AI agent in production today exposes user data by design.

Bankr. A user issues trade commands by tagging the bot in a public Farcaster cast or X reply. The cast is permanent, the wallet is on-chain, and the link between the human persona and the trading wallet is publicly indexed.

Aixbt. Calls and analyses are posted to the public X timeline. Users who act on a call have their resulting trade visible to anyone watching the wallet.

Clanker bots. Token deployments are tagged on Farcaster, linking the deploying account to the originating wallet permanently.

Generic alpha groups. Telegram or Discord groups gate access via wallet verification, exposing every member's wallet address to the group operator and frequently to other members.

The aggregate effect is that any trader large enough to matter is also large enough to be tracked, copied, and front-run.

3. Threat Model

We define the protocol's adversarial assumptions before describing the construction. SHADE is designed to be secure against the following adversaries:

Definition 1 (Network adversary $\mathcal{A}_N$)**.** *A passive observer of all network traffic between user and SHADE infrastructure. $\mathcal{A}_N$ can record every packet but cannot break standard cryptographic primitives (TLS 1.3, AES-256, Curve25519).*

Definition 2 (On-chain adversary $\mathcal{A}_C$)**.** *An observer with full read access to Base, all connected L2s, and all public mempool data. $\mathcal{A}_C$ can correlate timestamps, addresses, transaction patterns, and gas fingerprints across chains.*

Definition 3 (Operator adversary $\mathcal{A}_O$)**.** *The SHADE service operator itself, assumed to be honest-but-curious. $\mathcal{A}_O$ controls the application servers, ingestion pipeline, and front-end—but cannot break TEE attestation or zero-knowledge soundness.*

Definition 4 (Subscriber adversary $\mathcal{A}_S$)**.** *A legitimate paying subscriber who attempts to deanonymize other subscribers, harvest signals for resale, or correlate group membership with on-chain activity.*

The protocol guarantees the following properties under the standard cryptographic assumptions of each underlying primitive (DDH for Curve25519, knowledge-of-exponent and AGM for Groth16, SGX/SEV hardware integrity for TEE attestation):

Property 1 (Subscription unlinkability)**.** *For any wallet w holding $\geq T$ \$SHADE that authenticates to the service via a Semaphore proof π, no adversary in $\{\mathcal{A}_N, \mathcal{A}_O, \mathcal{A}_S\}$ can link π to w with non-negligible advantage.*

Property 2 (Query confidentiality)**.** *For any research query q submitted by an authenticated session, no adversary in $\{\mathcal{A}_N, \mathcal{A}_O, \mathcal{A}_S\}$ outside the TEE enclave can recover q or its result r.*

Property 3 (Trade-attribution unlinkability)**.** *For any executed trade t routed through Railgun or fresh-wallet rotation, $\mathcal{A}_C$ cannot link t to the user's primary identity wallet w with probability better than the anonymity-set size $|S|$ allows:* $\Pr[\mathsf{link}(t, w)] \leq 1/|S| + \mathsf{negl}(\lambda)$.

The protocol explicitly does *not* protect against:

- A global active adversary controlling all network paths and the TEE hardware vendor simultaneously.
- Side-channel attacks on the user's local device (keylogger, screen recorder).
- Out-of-band identity leaks (the user posting their trade publicly on Farcaster).
- On-chain settlement visibility—this is a property of Base, not SHADE.

4. Architecture

SHADE is structured as three layers: ingestion and scoring, privacy, and delivery and action.

4.1 Layer 1: Ingestion and scoring

The ingestion layer maintains a real-time event stream over Base. The stack consists of:

- An Erigon archive node with a custom `eth_subscribe` filter for the deployment factories of Clanker, Flaunch, Bankr, and Zora.
- A Kafka event bus partitioned by launchpad, with a 7-day retention window for replay and backfill.
- A Postgres-backed enrichment service that joins each new deployment against indexed historical state within 200ms p95.
- A Farcaster Hub subscriber consuming the cast firehose, filtered for token mentions and tags.

On-chain inputs:

- Liquidity lock status and depth (Uniswap v3/v4 pool state, lock contract verification).
- Holder concentration (top-10 share, Gini coefficient, Herfindahl-Hirschman index).
- Dev wallet history (prior deployments via the same EOA, prior rugs by graph traversal of funding paths, ENS/Farcaster ID linkage).
- Contract bytecode pattern matching against known templates and known bad actors using a Bloom-filter index of malicious opcodes.

Off-chain inputs:

- Farcaster cast volume and engagement velocity in the first 60 seconds after deploy.
- Casting account's Neynar reputation score and historical hit-rate.
- X mentions weighted by follower-quality score.

The scoring engine outputs a 0–100 composite plus a discrete set of risk flags.

[DIAGRAM: Ingestion flow—launchpad event → Kafka → on-chain enrichment → social enrichment → scoring engine → output (score, flags).]

4.2 Layer 2: Privacy primitives

The privacy layer composes four production-ready primitives, each addressing a different leak point.

4.2.1 Semaphore: zero-knowledge group membership

Semaphore is a zk-SNARK protocol over BN254 for anonymous group membership proofs. SHADE maintains an on-chain Merkle tree $\mathcal{T}$ of identity commitments, where each commitment is:

$$\mathsf{commit} = \mathsf{Poseidon}(\mathsf{identityNullifier}, \mathsf{identityTrapdoor})$$

A user holding $\geq T$ \$SHADE generates a Groth16 proof π attesting to four facts:

1. Knowledge of pre-image to a leaf in $\mathcal{T}$.
2. Knowledge of a wallet w such that $\mathsf{balanceOf}(w) \geq T$ at a recent Base block.
3. A nullifier $\mathsf{null} = \mathsf{Poseidon}(\mathsf{identityNullifier}, \mathsf{externalNullifier})$.
4. A signal payload s bound to the proof.

The verifier accepts π iff Groth16 verification succeeds and null has not been previously used within the current external nullifier scope (e.g., the current epoch). The wallet address w is never transmitted; only the proof π.

Proof generation runs entirely client-side in WebAssembly, with circuit complexity of approximately 2^{20} constraints and proving time of 3–5 seconds on a modern laptop. Verification is constant-time, on the order of 2 ms.

Listing 1: Client-side proof generation (illustrative).

```
import { Identity } from "@semaphore-protocol/identity";
import { generateProof } from "@semaphore-protocol/proof";

const identity = new Identity(walletSignature);
const group = await fetchSubscriberGroup();
const externalNullifier = currentEpochId();
const signal = sessionRequestPayload();

const proof = await generateProof(
    identity,
    group,
    externalNullifier,
    signal
);
// Send only `proof` to SHADE. Wallet never leaves the device.
```

4.2.2 XMTP: end-to-end encrypted delivery

Once authenticated, signals are pushed via XMTP. XMTP implements the Signal Protocol stack adapted for wallet-keyed identities:

- Identity keys are derived deterministically from a wallet signature over a canonical XMTP key-bundle message.
- Initial key agreement uses X3DH (Extended Triple Diffie-Hellman) over Curve25519.
- Ongoing message keys are derived via the Double Ratchet algorithm, providing forward secrecy and post-compromise security.
- Payloads are encrypted with AES-256-GCM under per-message keys.

Crucially, XMTP nodes (including any operated by SHADE) store only ciphertext. The SHADE operator $\mathcal{A}_O$ cannot read its own outbound messages without the recipient's private key.

4.2.3 Phala TEE: confidential AI inference

Ad-hoc research queries ("score this arbitrary token") run inside a trusted execution environment. SHADE uses Phala Network, which orchestrates Intel SGX and AMD SEV-SNP enclaves under a unified attestation API.

The protocol works as follows:

1. The user fetches a remote attestation α from the enclave proving that the running code matches a published, audited binary hash h.
2. The user encrypts the query under the enclave's ephemeral public key epk, which is bound to α.
3. The enclave decrypts, executes the inference, and returns the result encrypted under the user's public key.
4. Neither plaintext query nor result ever appears outside enclave memory; nothing is logged.

Property 4 (Inference confidentiality)**.** *Under the assumption of TEE hardware integrity, no party outside the enclave—including the SHADE operator—can recover the query plaintext, the result plaintext, or correlate them with a session identity beyond the granularity exposed by network metadata.*

4.2.4 Railgun: shielded trade execution

Railgun provides shielded UTXOs on top of Base. A user deposits ETH or USDC into the Railgun shielded pool, and subsequent withdrawals are unlinkable to the original deposit beyond the pool's anonymity set.

For trade execution:

1. The user deposits v ETH into Railgun from their primary wallet w, producing a shielded note n_1.
2. After a delay Δ chosen from a stochastic distribution to avoid timing correlation, the user withdraws to a fresh wallet w'.
3. w' executes the trade. From $\mathcal{A}_C$'s perspective, w' is unlinkable to w within the anonymity set $|S|$ of the pool at time of withdrawal.

The size and freshness of $|S|$ directly determine the privacy guarantee. SHADE surfaces the current anonymity set size to the user before recommending a route.

[DIAGRAM: Privacy layer—wallet → Semaphore proof → session; query → TEE → encrypted result; signal → Railgun routing → on-chain settlement.]

4.3 Layer 3: Delivery and action

Delivery is multi-channel: XMTP inbox, zk-gated Telegram bot, and optional encrypted Farcaster DM. Builders with a premium tier credential receive an authenticated API stream over WebSocket, with rate limits enforced via per-credential token buckets.

The action layer is optional. Users can integrate SHADE as a pre-trade risk check on Bankr; a tagged trade is intercepted, scored, and either confirmed or warned against before execution. Users may also route execution through Railgun or through SHADE's fresh-wallet rotation service.

4.4 End-to-end protocol flow

Algorithm 1 SHADE subscription, query, and execution

User wallet w holds $\geq T$ \$SHADE
$\mathsf{commit} \leftarrow \mathsf{Poseidon}(\mathsf{nullifier}, \mathsf{trapdoor})$
Submit commit to Merkle tree $\mathcal{T}$ on-chain (one-time)
Per session:
Generate Groth16 proof $\pi \leftarrow \mathsf{Prove}(\mathcal{T}, w, \mathsf{epoch})$
Submit π to SHADE; receive session token σ
Subscribe to XMTP topic keyed by σ
For each new launch event e:
    Score e, encrypt under user's XMTP key, push to inbox
On user query q for token τ:
    Encrypt q under enclave key epk
    Receive encrypted result r
On user action (buy):
    Deposit v ETH to Railgun from w
    Wait $\Delta \sim \mathcal{D}$
    Withdraw to fresh w'
    w' executes trade for τ on Base

5. AI Scoring Methodology

The scoring engine produces a single 0–100 composite from a weighted blend of sub-scores. Let $S(\tau)$ denote the composite score for token τ:

$$S(\tau) = w_d \cdot D(\tau) + w_h \cdot H(\tau) + w_c \cdot C(\tau) + w_v \cdot V(\tau) + w_b \cdot B(\tau)$$

with initial weights:

- $w_d = 0.30$ (deployer reputation)
- $w_h = 0.25$ (holder distribution and liquidity health)
- $w_c = 0.25$ (caster reputation and hit-rate)
- $w_v = 0.15$ (engagement velocity)
- $w_b = 0.05$ (contract pattern analysis)

$D(\tau)$ is the historical hit-rate of the deploying wallet across all prior launches, weighted by recency and volume. Formally:

$$D(\tau) = \frac{\sum_{i \in \mathsf{prior}(w_d)} \mathbb{1}[\mathsf{peak}_i \geq 5 \times \mathsf{init}_i] \cdot e^{-\lambda(t_0 - t_i)}}{\sum_{i \in \mathsf{prior}(w_d)} e^{-\lambda(t_0 - t_i)}}$$

with decay constant $\lambda = \ln(2)/30$ days (30-day half-life).

$H(\tau)$ combines top-10 holder share h_{10}, liquidity-to-FDV ratio ρ, and lock status:

$$H(\tau) = (1 - h_{10}) \cdot \min(\rho/0.05, 1) \cdot \mathbb{1}[\mathsf{locked}]$$

$C(\tau)$ scores caster mentions weighted by Neynar reputation and historical hit-rate. $V(\tau)$ measures the slope of replies, recasts, and unique inflows in the first 60 seconds. $B(\tau)$ is a binary discount for known-bad bytecode patterns (honeypot, mint backdoor, fee-on-transfer trap).

Weights are recalibrated weekly via a gradient-boosted regression against realized 7-day performance of scored tokens. The methodology, weights, and historical performance are published openly on a public dashboard.

6. Privacy Model

6.1 The honest scope

SHADE is explicit about what it can and cannot hide on a public chain.

What SHADE hides:

- User identity at the application layer (Semaphore-gated subscription, Property 1).
- Research queries and their results (TEE inference, Property 2).
- The link between user identity and executed trades (Railgun routing, Property 3).

What SHADE does not hide:

- The trades themselves—Base is a public chain.
- On-chain state and settlement—visible to all observers.
- Aggregate timing patterns of message bursts (mitigated by cover traffic; not eliminated).

This boundary is not a limitation; it is a property of building privacy infrastructure on a transparent ledger. Any project that claims on-chain trade privacy on Base without using a dedicated privacy chain is either lying or misinformed.

6.2 Anonymity-set analysis

The privacy guarantee for trade-attribution unlinkability is parameterized by the Railgun anonymity set $|S|$ at withdrawal time. We define the effective anonymity set as the number of deposits of comparable magnitude within a rolling time window W:

$$|S_{\mathsf{eff}}|(t, v) = |\{d_i : |v_i - v| \leq \epsilon \cdot v \ \wedge \ t_i \in [t - W, t]\}|$$

with $\epsilon = 0.10$ (10% magnitude tolerance) and $W = 24$ hours by default. SHADE displays $|S_{\mathsf{eff}}|$ to the user and recommends delaying execution if the set is below a configurable threshold.

6.3 Comparison to common patterns

Most projects marketed as "privacy AI" fall into one of two failure modes. The first is fake privacy: a wallet-gated Discord that stores every member's wallet address. The second is impossible privacy: claims of fully private on-chain execution on a public chain, which collapse on inspection.

SHADE rejects both. The privacy claims are scoped, verifiable under stated cryptographic assumptions, and built on primitives that have been in production for at least one year each.

7. Tokenomics

7.1 Supply

Initial supply is 100 billion $SHADE. The full supply is paired into the launch liquidity pool. A minor protocol-owned reserve may be acquired post-launch via open-market buybacks; this reserve is fully on-chain and trackable.

7.2 Utility tiers

Holding $SHADE above defined thresholds unlocks tiered access:

- **Free tier (no holding required):** delayed signals, public caster leaderboard, daily digest.
- **Subscriber tier ($\geq T_1$):** real-time signals, encrypted research, sniper auto-feed.
- **Builder tier ($\geq T_2$):** authenticated API access, higher rate limits, webhook delivery.

Threshold quantities T_1, T_2 are denominated in $SHADE and adjusted by governance to maintain a target subscriber base.

7.3 Value accrual

Premium access can be paid in ETH or USDC. The fee flow is:

$$\mathsf{revenue} \rightarrow \mathsf{operational\ cost} \rightarrow \mathsf{buyback}(\$\mathsf{SHADE}) \rightarrow \mathsf{reward\ pool}$$

All non-operational fee revenue is used to market-buy $SHADE on Base via a programmatic TWAP executor. The acquired tokens fund a weekly reward pool distributed to the top-performing public snipers as ranked by realized PnL on a public leaderboard.

All accrual mechanics are on-chain and verifiable. The buyback contract emits events; the reward pool distributes via Merkle drops with public root commitments.

This design intentionally avoids the most common tokenomics failure modes: there is no insider unlock cliff, no synthetic utility ("governance"), and no off-chain promise.

8. Go-to-market

Phase 1 (Launch). Fair launch directly on Uniswap v3, with the full LP position time-locked until 2100. The free public caster leaderboard is published as a Farcaster channel; this leaderboard generates shareable content that markets the product organically. Success metric: 5,000 unique channel followers within 30 days.

Phase 2 (Growth). Direct partnerships with high-reputation Farcaster casters who become early subscribers and amplifiers. Integration as a pre-trade risk check inside Bankr. Integration with Flaunch revenue-split tooling. Success metric: 1,000 paying subscribers within 90 days.

Phase 3 (Scale). Open the builder API. Position SHADE as the privacy-preserving signal layer for the broader Base agent economy. Success metric: 10 third-party products integrating the SHADE API within 180 days.

9. Competitive Landscape

	Bankr	Aixbt	Clanker	Virtuals	**SHADE**
Base-native	Yes	No	Yes	Partial	Yes
Real-time launch signals	No	Partial	No	No	**Yes**
Multi-launchpad coverage	No	No	No	No	**Yes**
Wallet-anonymous access	No	No	No	No	**Yes**
Encrypted research	No	No	No	No	**Yes**
Private trade routing	No	No	No	No	**Yes**

The competitive position is the only project on Base that combines real-time multi-launchpad intelligence with a four-primitive privacy stack. SHADE is positioned as a layer above the existing agents, not a substitute for them.

10. Roadmap

Q1. MVP scoring engine. Public caster leaderboard live as a Farcaster channel. Free daily digest.

Q2. Fair launch directly on Uniswap v3 with LP locked until 2100. Semaphore subscriber tier live. XMTP delivery in production.

Q3. Phala TEE-based encrypted research. Bankr pre-trade co-pilot integration.

Q4. Railgun-based private routing. Builder API in public beta. First third-party integrations live.

11. Risks and Mitigations

Regulatory risk. Privacy primitives may attract regulatory scrutiny. SHADE is designed around pseudonymity, not full anonymity; subscription is gated by token ownership rather than by anonymity-preserving cash, and the protocol does not custody user funds. The deliberate honest scope (no on-chain trade hiding) materially reduces exposure relative to mixer-style protocols.

Technical risk. TEEs make hardware trust assumptions; documented side-channels exist. SHADE mitigates by publishing remote attestation, supporting multi-vendor TEEs (SGX, SEV-SNP, future TDX) over time, never holding user funds inside the enclave, and auditing the published binary against the running code.

Adoption risk. XMTP adoption is still building. SHADE mitigates by supporting parallel delivery channels (Telegram, Farcaster DM) until XMTP reaches critical mass.

Market risk. The Base agent narrative could cool. The underlying utility—faster, safer launch evaluation—persists regardless of narrative.

Competitive risk. Bankr or Clanker could add privacy features. SHADE mitigates with first-mover focus, open-source proofs, and the builder API—becoming part of the substrate rather than a competitor for it.

12. Team and Governance

[Placeholder—team biographies and governance model to be inserted prior to publication.]

13. Conclusion

The Base agent ecosystem has solved deployment, execution, and discovery—in public. What it has not solved is the privacy of the participants. As the ecosystem grows, this gap becomes the binding constraint on serious capital: a whale who cannot research without leaking, cannot subscribe without doxxing, and cannot trade without being copied is a whale who eventually leaves.

SHADE closes that gap with a scoped, honest, and shippable privacy stack: Semaphore-based zero-knowledge subscription, XMTP encrypted delivery, TEE-backed private inference, and Railgun-routed clean execution. It does not promise to hide what cannot be hidden on a public chain. It hides exactly what can be hidden, using primitives that are already in production today, with stated cryptographic assumptions and a defined adversarial model.

The token model aligns access, incentives, and value capture: subscribers pay in ETH or USDC, fees buy back $SHADE on the open market, and top performers earn from a transparent on-chain reward pool. The launch is fair, the locks are real, and the supply is visible.

SHADE is positioned at the intersection of two of the strongest narratives in crypto—Base AI agents and on-chain privacy—with a working product, an honest scope, and a tokenomics design that survives scrutiny. The opportunity is not to build a louder agent. It is to build the quiet one.

Appendix A: Glossary of Technical Terms

AES-256-GCM. Authenticated encryption with associated data using AES with a 256-bit key in Galois/-Counter Mode. Provides both confidentiality and integrity.

BN254. A pairing-friendly elliptic curve commonly used for zk-SNARK constructions including Groth16. Provides approximately 100 bits of security.

Curve25519. A Montgomery elliptic curve designed for efficient Diffie-Hellman, used by XMTP's X3DH key agreement.

DDH (Decisional Diffie-Hellman). A standard cryptographic hardness assumption underlying many privacy primitives.

Double Ratchet. A symmetric-key ratcheting algorithm used by Signal and XMTP to provide forward secrecy and post-compromise security in messaging.

Forward secrecy. A property of a key-agreement protocol such that compromise of long-term keys does not compromise past session keys.

Gini coefficient. A measure of distributional inequality between 0 (perfect equality) and 1 (perfect inequality), applied here to token holder distribution.

Groth16. A zk-SNARK proving system with constant-size proofs (around 200 bytes) and constant-time verification.

Merkle tree. A binary tree of hashes allowing efficient proof of inclusion of a leaf with $O(\log n)$ proof size.

Nullifier. A unique value derived from a private input that prevents double-use of a credential without revealing the credential itself.

Phala Network. A TEE orchestration platform supporting Intel SGX and AMD SEV-SNP enclaves with on-chain attestation.

Poseidon. A zk-friendly hash function optimized for use inside arithmetic circuits, dramatically cheaper than SHA-256 in zk-proof contexts.

Post-compromise security. A property such that even if a session key is compromised, future session keys recover security after the next ratchet step.

Railgun. A privacy protocol providing shielded UTXOs over Ethereum, Base, and other EVM chains via zk-SNARK-based proofs of valid balance transitions.

Remote attestation. A cryptographic proof produced by a TEE that the running code matches a specific binary hash, allowing a remote verifier to trust the enclave.

Semaphore. A zk-SNARK protocol for anonymous group membership proofs over BN254, supporting nullifier-based replay prevention.

Signal Protocol. The cryptographic protocol underlying Signal Messenger, combining X3DH and Double Ratchet for end-to-end encrypted messaging.

TEE (Trusted Execution Environment). A hardware-isolated execution context (e.g., Intel SGX, AMD SEV-SNP) that protects code and data from the host operating system.

TWAP (Time-Weighted Average Price). An execution algorithm that splits a large order into many smaller orders over time to minimize price impact.

X3DH (Extended Triple Diffie-Hellman). An asynchronous key-agreement protocol used by Signal and XMTP for initial session establishment.

XMTP. An end-to-end encrypted messaging protocol with native Base integration, used by Coinbase Wallet and other Base-native applications.

zk-SNARK. Zero-Knowledge Succinct Non-interactive Argument of Knowledge—a class of proofs that are short, fast to verify, and reveal nothing about the witness.