AIDOS FI

The First Privacy-Native Neo Bank Protocol on Solana

Write your own financial myth. Own your privacy.

Whitepaper v0.1 • June 2026

AIDOS is Solana's first privacy-native neo bank. It ZK-shields your entire financial life, deploys TEE-guarded AI agents that trade and manage your portfolio autonomously, and routes every swap through an on-chain darkpool — invisible to MEV bots, copy-traders, and surveillance. It ships with a developer SDK so anyone can fork a private bank in five lines. Spend. Save. Earn. Delegate. Nothing hits the chain in plaintext. Fair launch via PumpFun — no pre-sale, no VC, community float from day zero.

1. Introduction

Every neo bank today is a panopticon. Revolut, Wise, Monzo, and their peers know your balance, your salary, your spending patterns, your physical location at the moment of every tap, and every counterparty you have ever transacted with. They mine that data to underwrite products you did not ask for, they surrender it to governments and law-enforcement on request, they sell inferences about you to partners, and — with grim regularity — they lose it in breaches that expose millions of customers at once. The bargain you strike in exchange for a clean mobile app and instant notifications is the quiet forfeiture of your entire financial life. You do not own your data; you rent access to a record that someone else controls, reads, and can revoke.

On-chain finance promised an escape from that arrangement and, for privacy, delivered something worse: a panopticon with no walls at all. Every balance, every transfer, and every open position is permanently, globally, and freely public. The moment you receive your first paycheck on-chain, your salary, your savings, and your spending are legible to anyone with a block explorer. MEV bots front-run your swaps in the seconds before they confirm, extracting value that should have been yours. Copy-traders clone your strategy the instant it becomes visible, crowding the very trades that made it work. Chain-analytics firms de-anonymize a fresh wallet within hours by clustering its behavior, and once your address is linked to your identity, that link is permanent and retroactive — it exposes not just what you do next, but everything you have ever done. Transparency, the property that makes blockchains trustworthy, is the same property that makes them hostile to anyone who simply wants to be left alone.

The industry's response has been to bolt privacy on as an afterthought: a mixer here, a shielded pool there, a single private transfer primitive that users must remember to invoke and that breaks the moment they touch a transparent contract. These are tools, not banks. They privatize one action while the surrounding financial life stays fully exposed, and they offer no path to compliance, which is why they keep colliding with regulators. Privacy bolted on after the fact is privacy that leaks.

AIDOS resolves the contradiction. It is not a privacy feature attached to a wallet; it is a complete neo bank — debit cards, IBAN rails, shielded yield, recurring payments, and autonomous AI portfolio management — designed from the first line so that *the protocol itself cannot see your money*. Balances live behind zero-knowledge proofs rather than in a readable ledger. AI agents that manage your portfolio execute inside hardware secure enclaves where even the operator cannot observe them. Swaps settle through a darkpool that never exposes order intent to a public mempool. And compliance, rather than being abandoned, is achieved through selective disclosure: you prove a specific fact to a specific party without surrendering the underlying data to everyone forever. Privacy is the default, disclosure is the exception, and surveillance is structurally impossible rather than merely discouraged by policy.

The result is a banking experience that feels familiar — a card in your wallet, a balance on your screen, money that earns while it sits — but rests on a foundation where the bank is blind by construction. You hold the keys. You choose what to reveal. Nobody, including AIDOS, can reassemble your financial life from what the chain records.

This document specifies the four products that compose the protocol (§3–§6), the privacy architecture that binds them (§7), the doctrine of what the protocol structurally cannot do (§8), the token and launch model (§9), and the verification guarantees that let anyone check these claims rather than trust them (§10).

2. Design Principles

Five principles govern every design decision in AIDOS. They are not aspirations to be balanced against convenience; they are constraints that the architecture is built to satisfy, and any feature that would violate one is not shipped.

- **Privacy by default** — Shielding is not an opt-in mode that a cautious user must remember to enable. Every balance and every transfer is private from the very first deposit, with no plaintext path to fall back into. Disclosure is the exception — always explicit, always scoped, always chosen by the user — not the default state that privacy must be carved out of.
- **Self-custody always** — AIDOS never holds your keys and can never sign a transaction on your behalf. It operates pools and verifies proofs; it does not take

custody of funds. This is what makes it a protocol rather than a bank in the legal sense: there is no account for an operator to freeze, no balance for a court to seize from AIDOS, because AIDOS never holds it.

- **Compliance without surveillance** — Regulation and privacy are usually framed as opposites; AIDOS treats them as orthogonal. Selective disclosure lets a user prove a single fact — that KYC was passed, that funds are sufficient, that they are not a sanctioned entity — to exactly one counterparty, without exposing the underlying documents, amounts, or history to anyone else. You can satisfy a regulator without building a database for an attacker.
- **Verifiable, not trusted** — Promises are not a security model. Open-source clients let anyone audit the circuits and code, client-side proof verification lets every user confirm a transaction's validity locally, and TEE remote attestation lets you prove that an agent ran exactly the code you deployed. Wherever the design would otherwise ask for trust, it instead provides a way to check.
- **Protocol over product** — The Aidos app is the first client, not the whole system. The same primitives that power it — shielded accounts, cards, TEE agents, darkpool routing — are exposed through an open SDK so that anyone can build a private bank on top, embed shielded payments into an existing product, or fork the entire stack. Infrastructure outlives any single application.

3. Product 01 — Aidos App

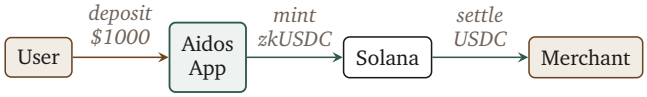
Status: v0.1 — in build. The Aidos App is the bank that cannot see your money. It is the flagship client of the protocol and the surface most users will touch first: a familiar mobile and web banking experience where you deposit funds, hold a balance, spend with a card, earn yield, and pay your bills — except that every figure on the screen is yours alone. Each balance lives behind a zero-knowledge proof rather than in a readable ledger entry, so the amount you hold is mathematically opaque to AIDOS, to chain observers, and to anyone who might later try to reconstruct your finances. You spend with a debit card that reveals the single payment in front of it and nothing about the portfolio behind it. The experience is ordinary; the privacy underneath it is total.

3.1 Capabilities

- **Shielded balance** — Balances sit behind ZK commitments; nobody can read how much you hold.
- **Aidos Card** — Spend USDC directly from a shielded balance — virtual and physical, anywhere Mastercard is accepted, via Apple/Google Pay.
- **IBAN rails** — SEPA/SWIFT on-ramp through a licensed EMI partner. Fiat in, shielded USDC out, and back.
- **Shielded yield** — Earn on shielded deposits. Your APY is visible; your yield amount is not.
- **Recurring payments** — Subscriptions, bills, and payroll, all shielded.
- **Multi-currency** — USDC, USDT, EURC, and future private stablecoins — one shielded balance per asset.
- **Selective disclosure** — Prove KYC without revealing documents; prove balance sufficiency without revealing the amount.
- **Multi-sig shielded accounts** — Shared business, DAO, or family treasuries that stay private.
- **Full personal history** — You see your complete transaction history; nobody else can reassemble it.

3.2 Spend Flow

The mechanics are deliberately invisible to the user but worth making explicit. When you deposit funds, the protocol mints shielded `zkUSDC` — a balance represented as a cryptographic commitment, not a public number. When you swipe the card, the app unshields exactly the purchase amount and no more, generating a zero-knowledge proof that you hold sufficient funds without ever revealing how much you actually hold. The merchant settles in ordinary USDC through the same Mastercard rails any business already uses, so nothing changes on their side. What the chain records is a single line: USDC moving from the Aidos pool to a merchant. The total, the history, the rest of the portfolio — none of it touches the ledger.



A $5 swipe unshields exactly $5, backed by a ZK proof of funds; the merchant sees the payment, the chain sees one settlement line.

3.3 What the Chain Sees

Hidden

Total portfolio value, transaction count, counterparties, yield earned, spending patterns, salary deposits, savings.

Visible

A valid ZK proof that funds exist (no amount), and a single settlement line moving USDC from the Aidos pool to a merchant.

3.4 Representative Use Cases

Scenario	Why AIDOS
Remote worker paid in USDC	Salary invisible to landlord, ex, or competitors.
On-chain founder	Treasury shielded from rivals.
DeFi trader	Portfolio hidden from MEV and copy-traders.
DAO contributor	Private pay, tax-reportable via disclosure.
HNWI holder	Holdings invisible; card for daily spend.

4. Product 02 — Aidos Agent

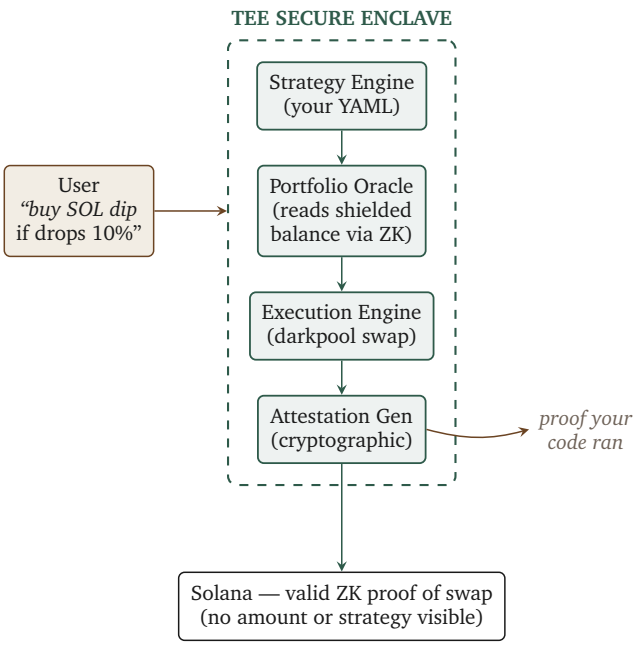
Status: v0.1 — in build. The Aidos Agent is a banker that lives inside a cryptographic fortress. Most people do not want to babysit a portfolio; they want it managed well, continuously, and privately. The Agent does

exactly that. You deploy an AI agent into a Trusted Execution Environment (TEE) — a hardware-isolated secure enclave — where it reads your portfolio, executes your chosen strategy, earns yield, exits losing positions, and rebalances around the clock. The crucial property is the asymmetry of visibility: the agent sees everything because it must in order to act on your behalf, yet *Aidos sees nothing, the chain sees nothing, and your government sees nothing*. The enclave is sealed; even the machine's operator cannot peer inside it. And you are not asked to take this on faith — cryptographic remote attestation lets you verify that the agent running in the enclave is the exact code you deployed, unaltered, every time it acts.

4.1 Capabilities

- **TEE-guarded execution** — The agent runs in an Intel TDX enclave. Keys are generated and stored inside and never leave.
- **Autonomous management** — Rebalance, compound yield, and exit losing positions, 24/7.
- **Strategy library** — 15+ pre-built strategies: DCA, grid, yield maximizer, risk parity, momentum, mean reversion.
- **Custom strategy DSL** — Express your own logic in a simple YAML config; the agent executes it faithfully.
- **Natural language** — "Sell 20% of my SOL at $200." The agent watches, waits, executes.
- **Cryptographic attestation** — Every action produces a remote attestation proving *your* code ran, not someone else's.
- **Multi-agent swarms** — Run several agents with different strategies across asset slices.
- **Agent marketplace** — Community strategies, each verified by TEE attestation before you run them.

4.2 Enclave Pipeline



4.3 Visibility Matrix

	AI (TEE)	Aidos Team	Chain Obs.	Gov.
Balance	✓	×	×	×
Strategy	✓	×	×	×
Trades	✓	×	×	×
P&L	✓	×	×	×
Attestation	✓	✓	✓	✓

The agent sees all to act; everyone else verifies integrity through attestation and sees nothing else.

5. Product 03 — Aidos Darkpool

Status: v0.2 — planned. The Aidos Darkpool lets you swap without leaving a trace. On a transparent blockchain, the act of trading is itself a leak: the moment your order intent reaches a public mempool, automated adversaries can read it, front-run it, and replicate it before it even confirms. A large trade announces itself to the entire market and pays for the privilege through slippage and sandwich attacks. The darkpool closes that window entirely. Orders never touch a public mempool, so there is nothing to front-run, nothing to copy, and no MEV to extract. Instead, orders are collected and matched in batch auctions that settle at fair midpoint prices, and only a single zero-knowledge proof — carrying no amount, no pair, and no trader address — is posted to Solana. The chain learns that a valid swap occurred and learns nothing else. This is the same architectural idea that institutional dark pools have used for decades to move size without moving the market, rebuilt to be self-custodial, on-chain, and consumer-native.

5.1 Capabilities

- **Pre-trade privacy** — Order intent is never broadcast to a public mempool, so adversaries cannot see what you intend to do before you do it.
- **Midpoint pricing** — Trades execute at the fair market midpoint rather than a price bots have already moved against you.
- **Low-impact large orders** — Direct buyer-to-seller matching avoids the AMM price impact that makes large on-chain trades expensive.
- **ZK settlement** — Only a valid proof reaches the chain — no amount, no pair, and no trader address is ever revealed.
- **MEV-resistant by design** — With no transparent mempool there is no surface for sandwich attacks; the resistance is structural, not heuristic.
- **Batch auctions** — Orders are pooled, matched, and cleared together in fair batches, so no single participant can be picked off in isolation.

5.2 With vs. Without Darkpool

Public mempool	Aidos darkpool
Swap $50K SOL→USDC	Swap $50K SOL→USDC
MEV bot sees the order	Order enters darkpool
Bot front-runs: lose ~2.3%	Batch auction matches
Copy-traders clone strategy	ZK proof settles on chain
47 wallets follow your moves	Chain sees only "a swap happened"

6. Product 04 — Aidos SDK

Status: v0.3 — planned. The Aidos SDK is how the protocol escapes the app. Everything the Aidos App does

— spawning shielded accounts, issuing cards, deploying TEE agents, routing through the darkpool — is exposed as an open developer surface so that anyone can build a private bank of their own. The ambition is deliberate: AIDOS is not trying to be the one bank everyone uses, it is trying to be the infrastructure that a thousand private banks are built on. A fintech can white-label the entire stack and ship a privacy-preserving neobank under its own brand. A DAO can run private payroll and treasury for its contributors. A developer can stand up a full banking bot inside Telegram in an afternoon. A game studio can give players a shielded in-game economy. The same five lines of TypeScript that create an account, fund it, issue a card, and deploy an agent are available to every builder, in every major language, with compliance tooling included rather than bolted on.

6.1 Surface

- **REST + WebSocket API** — Spawn accounts, issue cards, deploy agents programmatically.
- **Multi-language SDKs** — TypeScript, Python, Rust, and Go.
- **React hooks** — `<AidosProvider>`, `useShieldedBalance()`, `useAidosCard()`.
- **Bot templates** — Telegram relay — a full banking bot in ∼50 lines.
- **White-label cards** — Issue Aidos-powered debit cards under your own brand.
- **Compliance module** — Pluggable KYC/AML, selective disclosure, audit reports.

6.2 Build in Five Lines

```
import { AidosClient } from '@aidosfi/sdk';
const aidos = new AidosClient({ apiKey: KEY });

// shielded account + deposit
const acct = await aidos.createAccount({ label: '
    payroll' });
await acct.deposit('USDC', 5000);

// virtual card
const card = await acct.issueCard({ type: 'virtual',
    limit: 1000 });

// autonomous agent
await acct.deployAgent({ strategy: 'dca', asset: 'SOL
    ',
                         amount: 100, interval: '1w'
                             });
await card.spend({ merchant: 'starbucks', amount:
    5.50 });
```

6.3 Who Builds on Aidos

Builder	What they ship
Fintech startup	Fiat-crypto neobank with privacy.
DAO	Private payroll + treasury.
Telegram dev	A `/bank` bot for a community.
Remittance service	Private cross-border transfers.
Enterprise	Shielded employee expense cards.
Game studio	In-game shielded economy.

7. Architecture

AIDOS is a four-layer stack. Applications sit on top, the SDK exposes protocol calls, a privacy layer enforces the guarantees, and Solana provides settlement. The full stack is shown in Figure 1.

7.1 Stack Specification

Layer	Technology
ZK proving	Light Protocol / Helius ZK; Groth16, sub-second client-side.
TEE runtime	Phala Network (Intel TDX); GPU TEE for AI inference.
Darkpool	Custom batch auction + ZK settlement, Solana-native.
Card issuance	Mastercard partner (Bleap / Gnosis Pay model).
IBAN rails	Licensed EMI partner; SEPA + SWIFT.
Chain	Solana; 400 ms finality, $0.00025 fees.
Client	React Native (iOS/Android) + Web PWA; self-custody via Solana Mobile SDK.

8. Doctrine — What We Cannot Do

Most privacy promises are policies: a company tells you it will not look at your data, and you are asked to believe it. Policies can change, be quietly revised in a terms-of-service update, be overridden by a subpoena, or be ignored by a rogue employee. AIDOS is built so that the promises below are not policies at all — they are structural impossibilities. They hold because of cryptography and hardware, not because of corporate goodwill, and they would continue to hold even if AIDOS wanted to break them. The distinction matters: a bank that *chooses* not to read your account is one regulation away from reading it, while a protocol that *cannot* read your account has nothing to surrender.

- **Cannot see your balance** — Shielded balances are cryptographically opaque to Aidos. We operate the pool and verify proofs against it, but the individual holdings inside are commitments we have no key to open.
- **Cannot see your transactions** — Each transfer produces a zero-knowledge proof of validity, never a plaintext record. We verify that the proof is sound; we never read what it conceals.
- **Cannot see your strategy** — Your agent runs inside a TEE. Its code, its state, and its decisions are encrypted within the enclave; we confirm its integrity through attestation and never observe its logic.
- **Cannot freeze your funds** — Self-custody is the design, not a setting. AIDOS never holds your keys, so there is no account for us to freeze and no balance for anyone to seize from us.
- **Cannot sell your data** — There is no data to sell. It is never collected, never stored, and never monetized, so there is nothing to leak, subpoena, or auction.

What we *can* do is just as important. We provide selective disclosure: when you genuinely need to prove something — to a regulator, an auditor, a lender, or a counterparty — you generate a single cryptographic proof of exactly that fact rather than handing over a data dump. You can prove you passed KYC, that your funds are sufficient, or that you are not a sanctioned entity, all without exposing your documents, your balance, or your history to anyone beyond the party who needs that one assurance. Compliance and privacy stop being a trade-off.

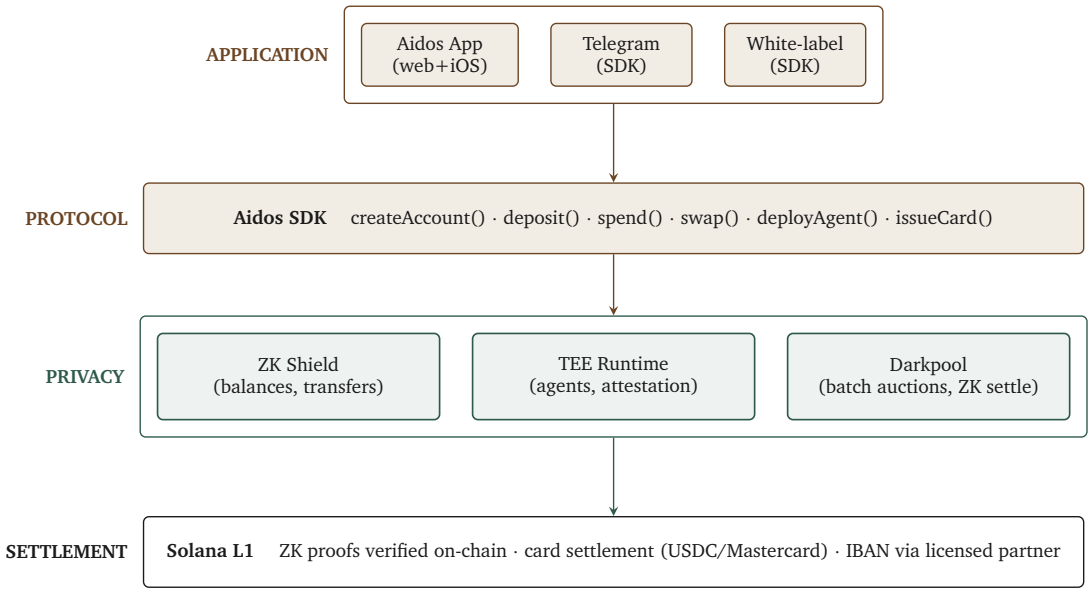


Figure 1: The AIDOS four-layer protocol stack.

9. Token & Launch

Token: $AIDOS • **Network:** Solana • **Launch:** fair launch on PumpFun.

$AIDOS launches fairly on PumpFun. There is no pre-sale and no private round — the token enters the market through the public bonding curve, open to everyone on the same terms from the first moment. The launch model is chosen deliberately: it puts the community and the builders on the same footing, participating in the same market at the same time, so that ownership of the protocol is shared from the start rather than concentrated before anyone else can take part. Liquidity is community-owned from block zero.

- **Fair launch on PumpFun** — The token is launched publicly on the PumpFun bonding curve, with no pre-sale and no private round. Everyone participates on equal terms from the same starting point.
- **Community-owned liquidity** — When the curve graduates, its liquidity migrates to a Solana DEX and the LP position is burned, leaving a permanent, community-owned market that no single party can withdraw.
- **Utility-aligned** — The token's role is tied to actual protocol usage — governance over parameters and access to the privacy stack — and is earned through the product rather than promised against a roadmap.
- **Self-custodial** — $AIDOS is held in your own wallet on Solana, consistent with the rest of the protocol: your keys, your tokens, your control.

10. Verification — Trust but Verify

A privacy protocol that asks to be trusted has already failed. The entire point of AIDOS is that its guarantees can be checked rather than believed, and the verification surface grows over the protocol's lifetime. What follows is what a user, an auditor, or a skeptic can independently confirm today, and what stronger guarantees arrive at v1.0.

10.1 Today (client-side)

- **Open-source client** — Every ZK circuit, every TEE payload, and every SDK call is open to inspection. You do not have to take our word for what the software does — you can read it, build it, and run your own.
- **Self-custody** — You hold the keys at all times; AIDOS has no ability to sign on your behalf. The most fundamental guarantee is the one you can verify simply by noting that we never possess what would be required to move your funds.
- **Client-side proof verification** — Every shielded transaction carries a zero-knowledge proof that you can verify locally, on your own device, without trusting any server to tell you it was valid.

10.2 v1.0 (cryptographic attestation)

- **TEE remote attestation** — Confirm that the agent running in the enclave is the exact code you deployed — not a modified, malicious, or substituted version. Tampering changes the attestation and is detectable.
- **Receipt chaining** — Every agent action emits a cryptographic receipt; chain them together and you have a tamper-evident proof-of-execution history that shows precisely what the agent did and when.
- **Warrant canary** — A cryptographically signed statement, refreshed on a fixed 30-day cadence. Its continued presence affirms the protocol's integrity; its sudden silence is itself a signal that something has gone wrong.

11. Competitive Landscape

The market AIDOS enters is not empty, but it is fragmented. There are excellent neobanks that offer cards and IBAN rails but operate as full surveillance custodians. There are self-custodial card products that return control of funds to the user but expose every transaction on a transparent chain. There are privacy protocols that shield transfers beautifully but offer no banking layer, no card, and no path to compliance. And there are

agentic-wallet frameworks that let AI manage funds but do nothing to hide what that AI is doing. Each solves a slice of the problem and leaves the rest exposed. The table below maps the landscape; the pattern it reveals is that no competitor ships banking, privacy, AI agents, and darkpool execution together in a single protocol. AIDOS is built precisely to occupy that gap.

Feature	Revolut	Gnosis Pay	Railgun	AgentKit	**AIDOS**
Debit card	✓	✓	×	×	✓
Self-custody	×	✓	✓	✓	✓
Shielded balance	×	×	✓	×	✓
ZK transactions	×	×	✓	×	✓
AI agent (TEE)	×	×	×	✓	✓
Darkpool swaps	×	×	×	×	✓
Shielded yield	×	×	×	×	✓
Selective discl.	×	×	×	×	✓
White-label SDK	×	✓	×	✓	✓
Telegram bot	×	×	×	×	✓

Banking. Privacy. AI. Protocol. Only AIDOS ships all four.

12. FAQ

- **Is AIDOS a bank?** — No. It is a protocol offering banking-like features through a self-custodial wallet and licensed partners. AIDOS never touches your funds.
- **Privacy with a debit card?** — The card sees only the merchant, amount, and that funds exist. Balance, history, and portfolio stay shielded.
- **Can the team see my agent's strategy?** — No. It runs in an Intel TDX TEE; code, state, and keys never leave the enclave, verifiable by attestation.
- **Is this Tornado Cash 2.0?** — No. Tornado Cash was a mixer. AIDOS is a full banking protocol with privacy by default *and* compliance tools that work with regulation.
- **Why Solana?** — 400 ms finality, $0.00025 fees, and a maturing privacy stack (Helius/Light ZK, Arcium MPC, PrivacyCash).
- **What if my agent goes rogue?** — It cannot. The TEE guarantees it runs your exact code; tampering changes the attestation and alerts you. You can also set hard limits and drawdown pauses.

A Living Document

This whitepaper describes AIDOS as currently designed and is a snapshot, not a contract. Features, scope, timelines, and technical details are subject to change as the protocol is built, as the surrounding ecosystem evolves, and as audits, testing, and community feedback inform the design. Everything here — the product capabilities, the architecture, and the launch mechanics — may be revised. When it is, this document will be updated and reissued with a new version number. Always refer to the latest published version for the current state of the protocol.

Changelog

Version	Date	Changes
v0.1	June 2026	Initial public draft. Introduces the four products (App, Agent, Darkpool, SDK), the four-layer architecture, the privacy doctrine, the verification model, and the PumpFun fair-launch token model.

Future revisions will be recorded here as new rows as the protocol and this document evolve.

AIDOS FI — Write your own financial myth. Own your privacy.