Stellalpha

Autonomous Non-Custodial Copy-Trading on Solana

Whitepaper

Version 1.2
February 2026

www.stellalpha.xyz

Contents

Abstract

Stellalpha is a permissionless, non-custodial, on-chain copy-trading protocol designed to democratize automated trading strategies for everyday users. While originally conceived as an EVM-based MVP, the protocol has migrated to a fully native Solana architecture to leverage high throughput, near-instant block times (400ms), and superior composability.

Stellalpha allows users to follow high-performing wallets ("Star Traders") and automatically mirror their trades with full transparency. The system utilizes a Vault-Based Non-Custodial Model, where user funds are secured in Program-Derived Addresses (PDAs) governed by an Anchor-based smart contract.

Unlike traditional botting tools that require sharing private keys, Stellalpha users retain full ownership of their assets, delegating only specific trading permissions to the protocol.

The architecture employs an off-chain Watcher Agent that monitors Star Trader activity. Upon detecting a trade, the agent calculates optimal routing via the Jupiter Aggregator API and passes instructions to the on-chain program. The program validates constraints and executes deterministic swaps via Jupiter CPI. A custom Relayer Service acts as the fee payer, enabling a seamless gasless experience.

1 Introduction

Decentralized finance (DeFi) has evolved rapidly, yet sophisticated trading remains inaccessible to the average user. Newcomers must navigate complex wallet management, liquidity routing, slippage, and MEV risks.

Stellalpha bridges this gap by enabling:

- **Permissionless Copy-Trading**: Anyone can follow high-performing wallets ("Star Traders") instantly.
- **Trustless Execution**: Trades are executed automatically via smart contracts without shared private keys.
- **Non-Custodial Security**: Users retain full control over their funds in segregated Program-Derived Addresses (PDAs).
- **Public Utility Automation**: Institutional-grade trading automation made accessible as a public good.

Open Source Repositories:
Vault Contract: https://github.com/akm2006/stellalpha_vault
dApp Frontend: https://github.com/akm2006/stellalpha

2 Problem Statement

1. **Manual Monitoring is Impractical**: Cryptocurrency markets operate 24/7 with high volatility. For retail traders, manually tracking and mirroring the trades of professional "alpha" wallets is physically impossible without automated tools, leading to missed opportunities and suboptimal entry prices.
2. **Custodial Risk in Centralized Platforms**: Traditional copy-trading platforms (e.g., eToro, Bybit) require users to deposit funds into centralized exchange wallets. This exposes users to counterparty risks, insolvency events (e.g., FTX), and potential withdrawal freezes.
3. **Key-Sharing Risks in Botting**: Existing on-chain automation tools often require users to export their private keys or API keys to third-party servers. This creates a massive security vulnerability, where a single server breach could result in total loss of user funds.
4. **High Costs on EVM Chains**: Ethereum-based solutions suffer from high gas fees and slow finality. This makes high-frequency copy-trading prohibitively expensive for smaller portfolios, as transaction costs erode potential profits.
5. **Lack of Trustless Standards**: There is currently no standardized, trustless protocol for on-chain copy trading that guarantees execution transparency without relying on opaque off-chain logic or custodial bridges.

3 Protocol / Architecture Overview

Stellalpha is built on the Anchor Framework and consists of five primary layers.

3.1 Vault-Based Non-Custodial System

The core of Stellalpha is the User Vault, a sovereign on-chain account acting as a smart wallet. It is derived via a Program Derived Address (PDA) deterministically seeded by the user's public key:

$$\text{UserVault} = \text{seeds}[b\text{"user\_vault\_v1"}, \text{user\_pubkey}]$$

This architecture ensures:

- **Sovereignty**: The vault is program-owned but user-governed. Only the user can authorize withdrawals or strategy allocations.
- **Security**: Funds remain within the Solana Token Program's safety guarantees and never leave the vault system until withdrawn by the owner.
- **Constraint Enforcement**: The smart contract enforces strict rules on how funds can be utilized, preventing unauthorized transfers or malicious drains.

3.2 Trader State Allocation

To participate in copy-trading, users do not send funds to the trader. Instead, they allocate capital to a dedicated 'TraderState' PDA. This account mirrors the Star Trader's actions but remains legally and technically owned by the user.

$$\text{TraderState} = \text{seeds}[b\text{"trader\_state"}, \text{owner}, \text{trader}]$$

This segregation allows for:

- **Isolated Risk**: Capital is isolated per strategy. A drawdown in one copy-trade does not affect the balance of others.
- **Granular Accounting**: Performance fees and high-water marks are calculated independently for each strategy.
- **Asset Separation**: Users can manage multiple copy-trades simultaneously without commingling assets.

ATA Segregation: Each `TraderState` PDA owns its own set of Associated Token Accounts (ATAs). This ensures that funds allocated to one strategy are cryptographically isolated from the user's main vault and other active strategies. Trading in one strategy cannot affect the balance or security of another.

3.3 Trader Watcher Agent

- Real-time monitoring via WebSockets or Geyser plugins
- Filters active Star Trader transactions
- Computes optimal routes off-chain via Jupiter API to minimize price impact

3.4 Equity-Based Signal Normalization

Stellalpha utilizes an "Equity Model" to determine trade sizing. Instead of mirroring raw token movements, the protocol replicates the *trade intent* (portfolio weight) of the Star Trader.

- **Buy Orders**: The ratio is calculated as the trade value divided by the trader's total buying power (available equity).

$$\text{Ratio} = \frac{\text{Trade Value}_{\text{USD}}}{\text{Total Buying Power}_{\text{USD}}}$$

- **Sell Orders**: The ratio is the percentage of the specific asset holding being sold.

$$\text{Ratio} = \frac{\text{Token Amount Sold}}{\text{Total Token Holdings}}$$

This ensures that followers replicate the conviction of the trader proportionally, scaling effectively across different portfolio sizes.

3.5 Jupiter CPI Execution Engine

The execution layer utilizes Jupiter's Cross-Program Invocation (CPI) interface for optimal trade settlement.

1. **Route Submission**: The Watcher Agent submits the swap instruction to the Stellalpha program.
2. **Constraint Validation**: The on-chain program verifies that the swap adheres to the user's constraints (slippage, allowed assets).
3. **Fee Collection**: Platform fees are deducted before execution.
4. **CPI Execution**: The program invokes the Jupiter CPI to execute the swap, passing the validated data. The `TraderState` PDA signs for the funds via `invoke_signed`.
5. **Settlement**: The resulting assets are returned to the `TraderState` token accounts.

3.6 Relayer Service

- Acts as the Solana fee payer for transactions

- Provides a gasless user experience
- Does not take custody of user assets

4 Tokenomics

4.1 Token Supply

The native Stellalpha token will feature a fixed-supply economic model to ensure scarcity and long-term value alignment.

- **Total Supply**: To be announced (TBA) - Determined based on network launch parameters
- **Supply Model**: Fixed supply with no inflationary mechanisms

4.2 Token Utility

- **Staking & Revenue Share**: Stake tokens to earn a 30% share of protocol performance fees generated from profitable copy trades.
- **Governance Rights**: Vote on new Star Trader whitelisting, risk parameters, and treasury allocations.
- **Fee Reduction**: Hold tokens to receive up to 50% discount on protocol fees and priority relay processing.
- **Exclusive Strategy Access**: Hold tokens to gain early entry to capped high-performance Star Trader vaults.

5 Roadmap

5.1 Operational Roadmap

- **Phase I: Foundation (Current)**: Testing and achieving correctness and speed in demo vault environment. Verifying Helius integration.
- **Phase II: Mainnet Beta**: Controlled launch with limited users and whitelist access. Pending security audit completion.
- **Phase III: Public Launch**: Full Mainnet release with permissionless vault creation and increased cap limits.
- **Phase IV: Evolution**: Advanced features based on user feedback. Cross-chain expansion research and DAO governance.

6 Competitive Analysis

The following table compares Stellalpha's non-custodial copy-trading approach with traditional centralized trading platforms and API-based trading bots:

Feature	**Stellalpha**	**Centralized Platforms**	**API Bots**
Custody	Non-Custodial (PDA)	Custodial (Exchange)	Exchange Funds
Security Model	PDA Vaults	Trust-Based	API Key Risk
Execution	On-Chain (Jupiter CPI)	Opaque Off-Chain	API Triggered
Transparency	100% On-Chain	Low	Low
Latency	Near Real-Time	High	Variable

Table 1: Comparison of Stellalpha with existing solutions

7 Team & Advisors

- **Aakash Mandal** (Founder, Developer): Founder & builder working on blockchain, automation, and decentralized systems. Lead developer of Stellalpha, focused on non-custodial trading infrastructure.
- **Manobendra Mandal** (Co-founder, Developer): Co-founder & developer with strong Web3 and full-stack experience. Hackathon-driven builder focused on smart contracts, dApps, and scalable systems.

8 Legal / Compliance Notes

This document is informational only and does not constitute an investment offer.