StellAlpha: Autonomous Non-Custodial Copy-Trading on Solana

Whitepaper

Version 1.1

November 2025

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, sub-second finality, 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 via real-time WebSockets (or Geyser plugins). Upon detecting a trade, the agent calculates the optimal routing path via the **Jupiter Aggregator API** and passes these instructions to the on-chain program. The program then cryptographically validates the trade against user-defined constraints (slippage, daily limits) and executes the swap deterministically via **Jupiter CPI (Cross-Program Invocation)**. A custom **Relayer Service** acts as the transaction fee payer, ensuring a seamless "gasless" experience for the user while maintaining a strict non-custodial boundary.

Table of Contents

1. Introduction
2. Problem Statement
3. Protocol / Architecture Overview
4. Tokenomics
5. Feature Rollout Plan
6. Roadmap (Visionary & Operational)
7. Competitive Analysis
8. Legal / Compliance Notes
9. Team & Advisors
10. References

(Note: The Table of Contents above is for navigation. Ensure that each section header in the document corresponds to the anchor links so the TOC is accurate.)

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. Meanwhile, "alpha wallets" and experienced traders capitalize on information asymmetry and automation to generate consistent returns.

StellAlpha bridges this gap by building an ecosystem where:

- **Anyone can follow high-performing wallets.**
- **Trades are executed automatically, transparently, and trustlessly.**
- **Users remain in full control of their funds.**
- **Automation becomes a public utility, not a luxury.**

While our initial proof-of-concept on EVM demonstrated the viability of agent-driven trading, the migration to Solana addresses critical bottlenecks in gas costs and latency. By leveraging Solana's parallel execution environment and PDA architecture, StellAlpha redefines copy-trading as a transparent, decentralized, and truly non-custodial experience.

2. Problem Statement

Users who wish to replicate the success of on-chain traders face significant hurdles:

1. **Manual Monitoring is Impractical:** Crypto markets operate 24/7 with sub-second volatility. Manually tracking profitable wallets is inefficient, and missing a single trade entry or exit can ruin a strategy.
2. **Centralized Risks:** Traditional copy-trading platforms (e.g., centralized exchanges) require users to deposit funds into custodial wallets, exposing them to withdrawal freezes, insolvency risks, and opaque execution engines.
3. **Security Vulnerabilities in Current Automation:** Existing trading bots often require users to input their **Private Keys** or API keys into off-chain software. This practice is fundamentally insecure and has led to millions in losses from hacks and data leaks.
4. **EVM Limitations:** On Ethereum and L2s, high gas fees and slower block times make high-frequency copy-trading cost-prohibitive for smaller portfolios.

5. **Lack of Trustless Execution:** There is currently no mainstream protocol that combines safe fund custody (via smart contracts) with real-time, programmatic strategy replication.

3. Protocol / Architecture Overview

StellAlpha leverages a high-performance architecture built on the **Anchor Framework**, designed for speed, safety, and non-custodial control. The system is composed of four primary layers.

[Insert Technical Diagram – e.g., a system architecture diagram illustrating the User Vault, Watcher Agent, Relayer, and Jupiter CPI interaction.]

Figure 1: StellAlpha Architecture Flow

3.1 Vault-Based Non-Custodial System (PDA Model)

Instead of a standard wallet, each user initializes a **User Vault**—a Program Derived Address (PDA) derived from seeds [b"vault", user_pubkey].

- **Ownership:** The vault is owned by the StellAlpha program but governed strictly by the user's wallet.
- **Security:** Funds stored in the vault cannot be withdrawn by the developers or the trading agent. They can only be used for swapping tokens as authorized by the user's settings.
- **Constraint Enforcement:** All trades are validated on-chain against user-defined rules (e.g., Max Trade Size, Daily Volume Limit, Whitelisted Tokens).

3.2 Trader Watcher Agent (Real-Time Monitoring)

To handle Solana's 400ms block times, simple polling is insufficient. StellAlpha employs a sophisticated **Transaction Monitor**:

- **Technology:** Uses **RPC WebSockets** (or Geyser Plugins in production) to subscribe to transaction logs in real-time.
- **Detection:** The agent filters for specific "Jupiter Swap" instructions originating from watched "Star Trader" wallets.
- **Route Calculation:** Upon detecting a trade, the Agent queries the **Jupiter Aggregator API** off-chain to calculate the optimal split-route for the user's trade size. This ensures the user gets the best possible price execution without burdening the on-chain program with heavy computation.

3.3 Jupiter CPI Execution Engine

The core trading logic is executed securely on-chain:

1. The Agent submits the route instructions to the StellAlpha Program.
2. **Validation:** The Program verifies the trade matches the user's followed trader, checks slippage bounds, and ensures the user has sufficient balance.
3. **Execution:** The Program invokes the **Jupiter V6 Program** via **CPI (Cross-Program Invocation)**.
4. **Signing:** The Vault PDA signs the swap instruction internally. User funds are swapped directly within the vault accounts, never leaving the protocol's safety perimeter.

3.4 Relayer Service (Fee-Payer Architecture)

StellAlpha utilizes a native Solana **Fee Payer** model to provide a seamless experience:

- **Mechanism:** The Relayer Service constructs the transaction and signs as the fee_payer.
- **Gasless Experience:** Users do not need to hold SOL for transaction fees in their trading vaults.
- **Security:** The Relayer functions purely as a transaction facilitator; it has **no authority** to withdraw or move user funds outside of authorized swaps.

4. Tokenomics

[Placeholder: Specific token distribution figures and charts to be added pending finalization]

4.1 Token Supply

- **Total Supply:** [TBA, e.g., 1,000,000,000]
- **Distribution:** A fixed supply model is proposed, distributed among Community Incentives, Ecosystem Development, Team/Advisors (vested), and Liquidity Provisioning.

4.2 Token Utility

The StellAlpha token serves as the backbone of the protocol ecosystem:

1. **Governance:** Holders vote on protocol upgrades, including the addition of new DEX integrations, fee structures, and supported asset whitelists.

2. **Utility & Access:**
    - **Fee Discounts:** Stakers receive reduced fees on Relayer services.
    - **Premium Features:** Access to advanced analytics, faster execution priority (Watcher priority), and "Alpha" strategy modules.
3. **Staking:** Users can stake tokens to earn rewards and participate in the protocol's insurance safety net.

5. Feature Rollout Plan

- **Phase 1: Solana Migration (Current Focus)**
    - Development of Anchor Program stellalpha_vault.
    - Implementation of PDA Vault architecture and initialize_vault instructions.
- **Phase 2: Agent & Watcher Development**
    - Implementation of WebSocket-based monitoring for Star Traders.
    - Integration of Jupiter API for off-chain route calculation.
- **Phase 3: Execution Engine & Frontend**
    - Integration of Jupiter CPI for on-chain swaps.
    - Deployment of Relayer Service for gasless fee payment.
    - Launch of React Frontend with Solana Wallet Adapter.
- **Phase 4: Testnet Launch**
    - Full deployment to Solana Devnet.
    - Community testing and bug bounties.

6. Roadmap (Visionary & Operational)

Visionary Roadmap

Our long-term vision is to become the leading on-chain automation layer for Solana. Over the next 5 years, we aim to evolve from simple copy-trading to a fully modular DeFi automation suite, enabling users to build complex, "set-and-forget" wealth management strategies without ever giving up custody of their assets.

Operational Roadmap

- **Dec 2025:** Solana Devnet Release & Beta Testing.
- **Dec 2025:** Mainnet Beta Launch (Pending Audit).
- **Q1 2026:** Token Generation Event (TGE) & Exchange Listings.
- **Q2 2026:** Introduction of Premium Automation Modules (e.g., DCA, Grid Trading).
- **Q3 2026:** Transition to DAO Governance.

7. Competitive Analysis

Feature	StellAlpha	Centralized Platforms (Binance/Bybit)	API Bots (3Commas/Zignaly)
Custody	**Non-Custodial (User Controlled)**	Custodial (Exchange Controlled)	Funds on Exchange
Security	**PDA Vaults (No Private Keys)**	High Trust Required	**High Risk** (API Key Sharing)
Execution	**On-Chain (Jupiter CPI)**	Opaque Internal Matching	API Triggered
Transparency	**100% Verifiable on Blockchain**	Low / Hidden	Low
Latency	**Real-Time (WebSockets)**	High	Variable

StellAlpha's Unique Selling Point: We are the first to combine the speed of Solana with a truly non-custodial, vault-based architecture that requires **zero private key sharing**, solving the "not your keys, not your crypto" dilemma inherent in other copy-trading solutions.

8. Legal / Compliance Notes

This whitepaper is for informational and technical purposes only. It does not constitute an offer to sell securities or a solicitation for investment.

- **Non-Custodial Nature:** StellAlpha is a software protocol. At no point does the team take possession of user assets; users retain full custody via their private keys and Program Derived Addresses.
- **Regulatory Compliance:** The project aims to comply with relevant regulations in its operating jurisdictions. Token utility and governance structures will be reviewed by legal counsel prior to TGE.

9. Team & Advisors

- **Aakash Mandal** – *Founder, Full-Stack & Blockchain Lead*
    - Developer responsible for the core protocol design, migration from EVM to Solana, and Anchor program development.
- **Manobendra Mandal** – *Co-founder, Full-Stack Developer*
    - 10x Hackathon Winner. Specializes in backend systems, relational architecture, and rapid DApp prototyping.

10. References

1. **Anchor Framework Documentation** – *Coral (Project Serum)*
2. **Jupiter Aggregator V6 CPI Documentation** – *Jup.ag*
3. **Solana Developers Guide (PDAs & Account Model)** – *Solana Foundation*
4. **StellAlpha GitHub Repository** – *github.com/akm2006/stellalpha*