Disclaimer This document is provided for informational purposes only. It does not constitute an offer to sell, a solicitation of an offer to buy, or a recommendation for any security, token, or financial instrument. Participation in digital asset projects carries inherent risks, and individuals should conduct their own due diligence and consult with professional advisors before engaging. The information herein reflects the project's vision and goals as of the publication date and may be subject to change. No guarantees are made regarding the achievement of goals, the accuracy of forward-looking statements, or the absence of risks.

Executive Summary stOLAS is a revolutionary liquid staking solution for the OLAS token in the Autonolas ecosystem, designed to democratize access to staking rewards while maintaining full DeFi liquidity. Unlike traditional staking that requires technical expertise and infrastructure management, stOLAS provides a seamless, cross-chain staking

experience that automatically deploys and manages services on behalf of users.

Built on the ERC4626 vault standard with a sophisticated cross-chain architecture, stOLAS bridges the gap between L1 (Ethereum) for user interactions and L2 (Gnosis Chain) for active staking operations. This design ensures maximum security for user deposits while enabling efficient, automated service deployment and reward collection.

The system's core innovation lies in its intelligent staking management: users simply deposit OLAS and receive stOLAS tokens, while the protocol automatically handles service deployment, monitoring, and reward distribution across multiple chains. This creates a truly liquid staking experience where users maintain full DeFi composability while earning passive staking yields.

Introduction to stOLAS The Autonolas ecosystem represents a paradigm shift in autonomous service coordination, where OLAS tokens power a decentralized network of intelligent agents and services. Traditional participation in this ecosystem requires users to deploy, manage, and maintain complex infrastructure - a barrier that has limited widespread adoption despite the ecosystem's potential.

stOLAS eliminates these barriers by abstracting away the technical complexity of service deployment and staking management. Users gain exposure to OLAS staking rewards through a simple deposit mechanism, while the protocol handles all operational aspects automatically.

The system operates on a dual-layer architecture: - L1 (Ethereum): User deposits, withdrawals, and vault management - L2 (Gnosis Chain): Active staking, service deployment, and reward collection

This architecture ensures that users can interact with the system on the most secure and liquid chain (Ethereum) while benefiting from the cost-effective and efficient staking operations on Gnosis Chain.

Token Utility and Value Proposition stOLAS serves as a pure utility token representing ownership of staked OLAS assets. Each stOLAS token is backed 1:1 by OLAS deposits, with the backing ratio potentially increasing over time as staking rewards accumulate.

Key utility aspects:

1. **Liquidity Preservation**: stOLAS tokens can be freely traded, transferred, and used across the entire DeFi ecosystem without losing staking exposure.

2. **Automated Yield Generation**: Users earn staking rewards automatically without managing services, monitoring performance, or handling technical operations.

3. **Cross-Chain Efficiency**: The system automatically bridges assets between chains to optimize for both security (L1) and efficiency (L2).

4. **DeFi Composability**: stOLAS integrates seamlessly with existing DeFi protocols, enabling users to leverage their staked assets for additional yield opportunities.

5. **Governance Separation**: stOLAS focuses purely on utility, while governance rights are handled through a future vstOLAS token, ensuring optimal token design for each use case.

Architecture Deep Dive

L1 Layer (Ethereum) - User Interface and Security The L1 layer serves as the primary user interface and security foundation:

stOLAS Vault (ERC4626) - Manages user deposits and withdrawals - Issues stOLAS tokens based on current exchange rates - Maintains full on-chain reserves - Provides standardized ERC4626 interface for DeFi integration

Depository - Handles cross-chain bridging operations - Manages staking model lifecycle - Coordinates with L2 for service deployment - Tracks staking model statuses and balances

Treasury - Processes withdrawal requests - Issues ERC6909 tokens for withdrawal tracking - Manages withdrawal cool-down periods - Coordinates L1-L2 withdrawal flows

Distributor - Distributes rewards between veOLAS and stOLAS holders - Manages reward allocation ratios - Updates stOLAS total assets - Handles reward distribution timing

Lock (veOLAS) - Manages voting escrow for governance participation - Enables long-term commitment rewards - Provides governance power proportional to lock duration

L2 Layer (Gnosis Chain) - Active Staking Operations The L2 layer handles the core staking operations and service management:

StakingManager - Orchestrates service deployment and management - Manages staking model creation and activation - Coordinates with service registry and verification systems - Handles cross-chain communication and state synchronization

StakingTokenLocked - Manages individual staking instances - Handles reward calculation and distribution - Tracks service performance and liveness - Manages staking model lifecycle

ActivityModule - Verifies service activity and liveness - Manages reward claiming processes - Coordinates with multisig wallets for service control - Handles activity-based reward adjustments

Collector - Gathers rewards from staking operations - Manages cross-chain reward bridging - Coordinates reward distribution timing - Handles emergency unstaking operations

Cross-Chain Bridge Infrastructure The bridge system enables seamless asset and data transfer between L1 and L2:

LayerZero Integration - Secure cross-chain messaging protocol - Handles token transfers and data synchronization - Provides finality guarantees across chains - Enables complex cross-chain operations

Bridge Processors - L1DepositProcessor: Handles deposits from L1 to L2 - L2StakingProcessor: Manages staking operations on L2 - TokenRelayers: Handle token transfers between chains - MessageRelayers: Synchronize data and state across chains

Operational Workflow

1. **Deposit Process** - User deposits OLAS into stOLAS vault on L1 - stOLAS tokens are minted 1:1 (initially) - OLAS is bridged to L2 via LayerZero - StakingManager deploys services with OLAS backing - User receives stOLAS representing their staking position

2. **Staking Operations** - Services operate on L2 with OLAS backing - Rewards accumulate based on service performance - ActivityModule verifies service liveness - Collector gathers rewards and prepares bridging

3. **Reward Distribution** - Rewards are bridged from L2 to L1 - Distributor splits rewards between veOLAS and stOLAS - stOLAS total assets are updated - Users can claim rewards or let them compound

4. **Withdrawal Process** - User requests withdrawal through Treasury - ERC6909 tokens are minted for withdrawal tracking - If L1 has sufficient OLAS: immediate withdrawal - If not: L2 unstaking is triggered and bridged back - User finalizes withdrawal after cool-down period

5. **Model Management** - Staking models can be retired when services are deprecated - Retired models allow gradual unstaking - System maintains full backing during retirement

process - Closed models release all remaining OLAS to users

Technical Specifications

ERC4626 Compliance - Standardized deposit/withdraw functions - Preview functions for accurate calculations - Total assets and shares tracking - Maximum deposit and withdrawal limits

Cross-Chain Security - LayerZero's proven security model - Multi-signature controls for critical operations - Time-locked upgrades and parameter changes - Comprehensive audit coverage

Scalability Features - Batch operations for multiple services - Efficient reward distribution algorithms - Optimized cross-chain communication - Modular contract architecture for upgrades

DeFi Integration and Use Cases

stOLAS is designed from the ground up for maximum DeFi composability:

Liquidity Pools - Uniswap V3 integration for deep liquidity - Concentrated liquidity strategies - Automated market making opportunities - Yield farming with stOLAS pairs

Lending Protocols - Collateral for borrowing operations - Interest-earning on borrowed assets - Leveraged staking strategies - Risk management through diversification

Yield Aggregators - Automated yield optimization - Multi-protocol yield strategies - Risk-adjusted return maximization - Professional portfolio management

Cross-Chain DeFi - Multi-chain yield opportunities - Cross-chain arbitrage strategies - Diversified risk exposure - Global DeFi market access

Security and Risk Management

Smart Contract Security - Comprehensive internal audits completed - External audit planned via Hunt platform - Bug bounty program for community participation - Open-source transparency for community review

Operational Security - Multi-signature controls for critical operations - Time-locked parameter changes - Emergency pause mechanisms - Gradual upgrade processes

Risk Mitigation - Full asset backing at all times - Diversified service deployment - Activity-based reward adjustments - Comprehensive insurance considerations

Cross-Chain Security - LayerZero's battle-tested security model - Multi-chain state verification - Fail-safe bridging mechanisms - Emergency unstaking capabilities

Governance and Decentralization

vstOLAS Token (Future) - Dedicated governance token - Voting power proportional to lock duration - Proposal and voting mechanisms - Parameter adjustment capabilities

Governance Areas - Fee structure and allocation - Reward distribution ratios - Service deployment parameters - Protocol upgrade decisions

Community Participation - Open proposal submission - Transparent voting processes - Community-driven development - Decentralized decision making

Roadmap and Development Phases

Phase 1: Core Infrastructure (Completed) - stOLAS vault deployment and testing - Cross-chain bridge implementation - Basic staking operations - Internal security audits

Phase 2: Governance Launch (In Progress - Q4 2025) - vstOLAS token deployment - On-chain governance processes - Community proposal system - Parameter adjustment mechanisms

Phase 3: Cross■Chain Expansion (Q1 2026) - Additional L2 deployments - Multi-chain staking strategies - Cross-chain yield optimization

Phase 4: External service integration (Q2 2026) - Integrate LST performance with other services - Launchpad for LST related KPI-performing services

Phase 5: DeFi Integration (Q4 2026) - Yield■aggregator partnerships - Liquidity pools

Economic Model and Tokenomics

Initial Distribution - No pre-mine or private allocation - Fair launch through open staking - Community-driven adoption - Transparent token distribution

Reward Structure - Staking rewards from service operations - veOLAS lock rewards for governance participation - DeFi yield opportunities through stOLAS - Cross-chain arbitrage opportunities

Fee Structure - Initially no protocol fees - Future governance-controlled fee parameters - Fee allocation to protocol development - Community treasury funding

Sustainability - Long-term reward sustainability - Service performance optimization - Cross-chain efficiency improvements - Community-driven development funding

Risk Factors and Considerations

Smart Contract Risks - Despite comprehensive audits, smart contract bugs remain possible - Complex cross-chain operations increase attack surface - Upgrade mechanisms introduce governance risks - Emergency procedures may have limitations

Operational Risks - Service deployment failures could impact rewards - Cross-chain bridge vulnerabilities - Multi-signature key management - Time-lock bypass mechanisms

Market Risks - OLAS price volatility affects stOLAS value - DeFi protocol risks in integrations - Liquidity provider impermanent loss - Cross-chain arbitrage opportunities

Regulatory Risks - Evolving staking token regulations - Cross-chain compliance requirements - DeFi integration legal considerations - Jurisdictional regulatory differences

Technical Risks - LayerZero protocol risks - Gnosis Chain security considerations - Service registry vulnerabilities - Activity verification accuracy

Conclusion

stOLAS represents a fundamental advancement in liquid staking technology, combining the security of Ethereum with the efficiency of Gnosis Chain to create a truly liquid staking experience. By abstracting away the technical complexity of service deployment and management, stOLAS democratizes access to OLAS staking rewards while maintaining full DeFi composability.

The system's cross-chain architecture ensures optimal performance across different use cases: L1 for security and liquidity, L2 for efficiency and cost-effectiveness. This design enables users to participate in the Autonolas ecosystem without technical barriers while maintaining full control over their assets.

As the DeFi ecosystem continues to evolve, stOLAS is positioned to become a cornerstone of cross-chain liquid staking, providing users with unprecedented access to staking yields while maintaining the flexibility and composability that modern DeFi requires.

The future development roadmap focuses on expanding DeFi integrations, launching governance mechanisms, and extending to additional chains, ensuring that stOLAS remains at the forefront of liquid staking innovation.

Glossary

OLAS: Native token of the Autonolas ecosystem, used for staking, coordination, and service incentives.

stOLAS: Liquid staking token representing ownership of staked OLAS held in the stOLAS vault.

vstOLAS: Future governance token granting voting rights over protocol decisions.

veOLAS: Voting escrow token for governance participation and long-term commitment rewards.

L1: Ethereum mainnet, primary chain for user interactions and security.

L2: Gnosis Chain, secondary chain for efficient staking operations.

Staking Model: Individual staking instance managing specific service deployments.

Activity Module: System component verifying service performance and liveness.

Bridge Processor: Contract handling cross-chain asset and data transfer.

LayerZero: Cross-chain messaging protocol enabling secure inter-chain communication.

ERC4626: Standardized interface for tokenized vaults in the Ethereum ecosystem.

ERC6909: Multi-token standard for managing withdrawal requests and tracking.

Contact Information

Official Website: https://lstolas.xyz **Repository**: https://github.com/kupermind/olas-lst **Documentation**: https://github.com/kupermind/olas-lst/doc/ **Security**: security@lstolas.xyz

For inquiries, partnership opportunities, and security reports, please contact the development team via the official website contact form or security email address.

This whitepaper represents the current vision and technical specifications of the stOLAS project as of [Publication Date]. The project team reserves the right to update this document as development progresses and new features are implemented.

[Technical Clarification] Technical Clarifications & Risk Disclosures (Non■binding)

> These clarifications **do not modify** the business/marketing language above. They document engineering dependencies observed in the current implementation (commit `a23db47`) so legal, product, and engineering stay aligned.

[Technical Clarification] L2 → L1 Withdrawal Routing (Configuration Dependency) - **What matters:** `UNSTAKE` returns **must** be routed to **Treasury (L1)** so withdrawal tickets can be paid. - **Status:** This is a **configuration requirement**, not a code defect. Intended mapping on L2 `Collector`: - `REWARD → Distributor (L1)` - `UNSTAKE → Treasury (L1)` - `UNSTAKE_RETIRED → UnstakeRelayer (L1)` - **Operational ask:** add a preflight that reads back receivers; alert on mismatch; monitor `OperationReceiversSet`, `TokensRelayed` (L2) and `WithdrawRequest*` (L1).

[Technical Clarification] ERC4626 Entrypoints Are Non■Standard by Design - `deposit()` — callable only by **Depository**; `redeem()` — only by **Treasury**. - `mint()` / `withdraw()` are not intended for external integrators. - **Why:** preserves end■to■end control while keeping ERC4626 math (PPS). Please reflect this in API docs/SDKs.

[Technical Clarification] Trust Model & Invariants (Depository → stOLAS) - Vault uses **internal reserve accounting**: `totalReserves = staked + vault + reserve`; `pps = totalReserves / totalSupply`. - `syncStakeBalances(...)` lets Depository **update accounting** and move assets for `topUp` only. - **Operational invariants:** keep internal totals aligned with real token movements; reconcile periodically off■chain.

[Technical Clarification] Asset Assumption - The asset is **OLAS**, assumed to be a **plain ERC■20** (no hooks/rebase/fee). Any change requires compatibility review.

[Technical Clarification] Upgrades & Control - Production ownership is expected to be **multisig → timelock**. Upgrades should follow fork rehearsals plus storage layout checks.

[Technical Clarification] Observability - Maintain dashboards for `totalReserves` breakdown, PPS, outstanding tickets, `UNSTAKE` inflows, and bridge latency.

[Technical Clarification] Roadmap (Operational Gates Summary)

Phase 0 — Security & Ops Readiness (new) Guardrails for routing; ERC4626 caveat docs; monitoring dashboards; multisig+timelock; fork tests; bug■bounty process.

Phase 1 — Core Protocol Stabilization Operate with soft caps & SLOs; weekly PPS/liquidity reports; incident post■mortems.

Phase 2 — Governance Launch vstOLAS (if applicable); timelock as executor; upgrade runbooks.

Phase 3 — Cross■Chain Expansion Additional networks; provider■diverse bridges; cross■chain yield strategies gated by monitoring.

**Phase 4 - External service integration ** Integrate with other services, support launchpad for LST related KPI-performing services.

Phase 5: DeFi Integration Yield■aggregator partnerships and liquidity pools considerations.