Merss: A High-Trust Yield Distribution Protocol v1.0.0

Joshua Sajous founder at www.mersstoken.org

Legal Disclaimer

This document is provided for informational and research purposes only and does not constitute an offer to sell or the solicitation of an offer to buy any securities, tokens, or financial instruments. Any future offering of Merss (MSZ) tokens, if applicable, will be governed by legally binding agreements. All forward-looking statements are subject to change.

Abstract

Merss (MSZ) is a next-generation decentralized finance protocol that redefines the economic utility of time through Proof of Trust Epochs (PoTE) and programmable yield systems. Building upon the foundational architecture of Maximus, Merss upgrades the model with refined smart contract logic, vesting mechanisms, DAO-controlled permissions, and a composable Delegated Yield Ledger (DYL). By encoding trust, duration, and delegation directly into the protocol's core, Merss enables the deterministic distribution of yield without staking, slashing, or speculation. With Mites as its atomic unit, MSZ supports micro-transactions, yield delegation, and time-weighted governance at scale.

1. Introduction

In decentralized finance (DeFi), short-term speculation and staking loops dominate incentive models. Merss challenges this paradigm by valuing time and consistency as the true basis of economic trust. Through PoTE, users are rewarded not for activity alone but for holding and delegating over defined epochs, creating a system where belief, patience, and contribution form the backbone of yield distribution.

Unlike traditional proof systems that emphasize resource expenditure (PoW) or token lock-ups (PoS), Merss records the passage of time as a primary asset. Trust Epochs serve as verifiable snapshots of how long tokens have been held or delegated, enabling a new generation of programmable, slashing-free, and socially-aware yield protocols.

2. Core Concepts

Proof of Trust Epochs (PoTE)

PoTE measures the consistency of token holding or delegation over time. Each epoch is a cryptographic checkpoint recording MSZ holding duration and any active yield pledges. These epochs are used to calculate trust weight, yield rights, and governance power without exposing users to slashing or unstaking penalties.

Delegated Yield Ledger (DYL)

The DYL is an append-only, permissionless ledger that records yield delegations and redirections. Users can stream their accrued or expected yield to any wallet or smart contract while retaining principal ownership of their MSZ. This enables dynamic relationships such as creator sponsorship, DAO support, or programmable royalties.

Mites: Micro-Units of Trust

1 MSZ is divisible into 1,000,000,000 Mites. These ultra-granular units power micro-yield flows, programmable trust weights, and reputational metrics. Mites allow economic participation even at sub-cent levels and support equitable involvement across low-income regions, education platforms, and DAO funding models.

3. Tokenomics

Total Supply: 100,000,000 MSZ

Divisibility: 1 MSZ = 1,000,000,000 Mites
Allocation:
- 30,000,000 MSZ (30%) pre-minted to DAO & partners
- 70,000,000 MSZ (70%) vested to treasury/owner over time

Emission Model
- MSZ follows a time-weighted, DAO-controlled emission schedule
- No arbitrary mining or inflationary rewards
- Early adopters benefit from higher Trust Epoch multipliers

Burn Rights
The owner may burn up to 1% of total supply (1,000,000 MSZ) to manage inflation or remove maliciously acquired funds

4. Network Architecture
Merss operates on a hybrid architecture combining Delegated Proof-of-Stake (DPoS) with PoTE. Key actors include:
Delegators: Hold or assign MSZ to signal trust
Executors: Generate Trust Epoch sequences and manage yield pledges
Verifiers: Confirm delegation streams and epoch validity

Each transaction involving delegated yield, epoch snapshots, or governance participation is processed with sub-second latency and validated against the DYL. Mites enable yield precision, and Verifier Nodes ensure determinism across EVM-compatible environments.

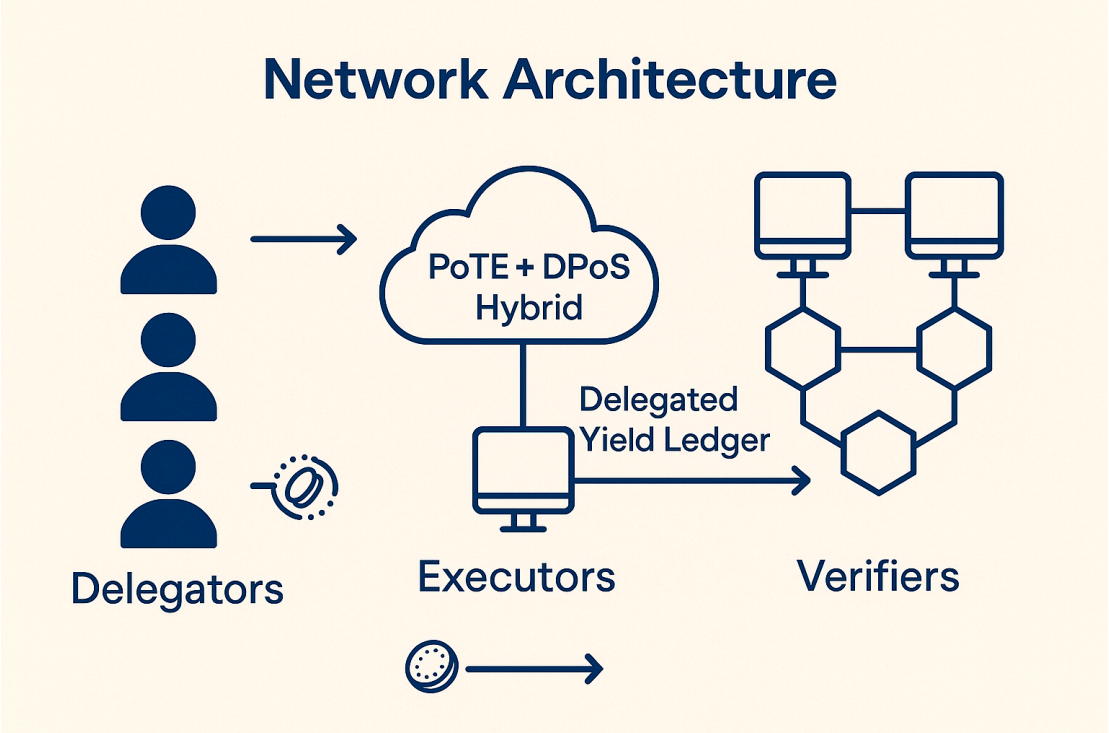

Image 1 Network Architecture

5. Applications
Regaarder Integration: Fans pledge Mites as trust, not payment, enabling a new model of creator support based on commitment. While the official Regaarder app is

not yet live, Merss will be fully integrated into the platform upon its launch, establishing a native trust-driven layer for creator-fan relationships.
Micro-DAOs: Small groups form around time-aligned goals, pooling delegated yield to fund mentorship, shared tools, or collective action.
Trust Pools: Reputation-based savings or credit groups formed by aggregating holding behavior and delegated belief.
Lending, Borrowing, and Collateralization with Delegated Yield: Users will be able to deposit MSZ or trust-locked assets as collateral to borrow against their time-weighted yield rights. These loans may be issued in stablecoins or other trusted assets. The Delegated Yield Ledger (DYL) allows borrowers to maintain principal ownership while directing future yield as collateral, making yield streams composable, revocable, and programmable. Borrowing terms such as interest rates or borrowing power can be algorithmically determined by trust history, duration of holding, and yield delegation behavior.

6. Governance
Trust Epoch-Based Voting Governance power is not determined by raw holdings, but by how long MSZ has been held or delegated. This discourages vote manipulation by short-term whales.
Proposal Lifecycle
Drafting: Open forum
Validation: Minimum trust endorsement threshold
On-Chain Voting: Weighted by Trust Epoch score
Slashing-Free Participation Voters face no penalties for downtime. Long-term holders are never penalized; instead, influence naturally fades if inactivity continues.

7. Smart Contract Mechanics (v1)
- Upgradeable architecture via UUPS proxy pattern
- DAO-exclusive functions gated via `onlyDAO()`
- Controlled transfer rights and `burn()` functionality
- External Chainlink integration for BNB/USD price feeds
- Yield claims follow a time-based vesting schedule: users become eligible to withdraw rewards only after a minimum holding period or Trust Epoch completion. The `canWithdraw()` function enforces this condition, ensuring that only committed participants receive yield. Claims can be executed periodically based on elapsed time, not block height.
- `canWithdraw()` gating logic based on time held or elapsed epoch

8. Future Work
zkPoTE Proofs: Zero-knowledge proof integration for confidential trust epochs and anonymous voting
Decentralized Identity (DID): Link trust to pseudonymous reputations for cross-platform validation, enabling the formation of epoch-weighted credit scores for lending markets and on-chain trust verification
Yield Marketplaces: Trade or lease delegated yield rights
Trust Bonding Curves: Time-lock MSZ to mint governance access tokens or fund public goods
Collateralized Yield Instruments: Design systems that allow delegated trust epochs to back lending, DAO fundraising, and synthetic asset issuance without requiring token liquidation

Programmable Repayment Streams: Develop smart contracts that automatically stream repayment in Mites over time, optimizing repayment schedules and improving capital efficiency for both lenders and borrowers

9. Conclusion

Merss (MSZ) transforms blockchain from a speculation game into a trust economy. It rewards commitment over hype, consistency over activity, and belief over volatility. Through programmable epochs, delegation mechanics, and DAO alignment, Merss offers a next-generation protocol for value built on time.

For developers, DAOs, creators, and everyday usersMerss is the economic layer where trust is not assumed. It's earned.

Moreover, the contract's modular architecture and composability enable seamless integration with external dApps, DAOs, and lending systems. Merss is engineered not only as a protocol, but as an interoperable economic substrate, where time-based trust can empower a variety of decentralized use cases.