# Nightcord: A Dual-Layer Blockchain for Latency-Constrained, Privacy-Preserving Decentralized Markets

V1.1 (Lite)

By Kanade
nightcord06@gmail.com

2/12/26

**Abstract:**

As cryptocurrencies and blockchain technologies reach mainstream adoption, core elements of the ecosystem have become increasingly centralized. Mining and staking power concentrate among large operators, while liquidity and execution are dominated by custodial exchanges. Meanwhile, decentralized exchanges preserve permissionless access but struggle with latency, limited throughput, and extractive market dynamics. This trajectory diverges from the foundational vision of decentralized, user-sovereign infrastructure. This paper introduces Nightcord, a hybrid blockchain architecture designed to reconcile decentralization, privacy, and high-performance decentralized trading. By separating execution from settlement and embedding market-aware constraints at the protocol level, Nightcord aims to reduce oracle dependency, manage state growth, and mitigate extractive behavior while maintaining permissionless participation. The proposed framework seeks to advance decentralized market infrastructure without sacrificing security, transparency, or user autonomy.

## 1 Introduction

The current blockchain ecosystem remains structurally fragmented between accessibility, performance, and privacy. Bitcoin's Proof-of-Work model secures decentralization through energy-intensive computation, effectively limiting participation to specialized operators, while Ethereum's Proof-of-Stake framework reduces energy costs but introduces capital barriers and governance complexity. Centralized exchanges deliver high-speed execution at the cost of mandatory identity verification and custodial risk, whereas decentralized exchanges preserve permissionless access yet struggle with on-chain latency, oracle dependency, and extractive ordering dynamics that limit performance. Nightcord proposes a layered, market-aware blockchain architecture designed to reconcile these trade-offs by separating execution from settlement, embedding market constraints at the protocol level, enforcing economic limits on

state growth, and enabling high-throughput decentralized exchange functionality without sacrificing privacy or user sovereignty.

# 2 Nightcord's Adversarial, AI-Mediated Market Maker Ecosystem

Nightcord's Adversarial AI-Mediated Market Maker Ecosystem is a self-stabilizing DeFi architecture where liquidity providers (Market Makers), traders, and an AI governance layer interact in a controlled adversarial environment. Market Makers create and manage liquidity pools, can responsibly borrow capital to expand liquidity, and are evaluated through a dynamic reputation system. Traders execute against these pools with built-in protections that limit exploitation and systemic risk. At the core, an AI-Automated Market Maker (AI-AMM) continuously monitors liquidity health, redistributes capital, merges failing pools, slashes malicious actors, and allocates incentives.

## 2.1 Actors

1. **Market Makers (MMs)**
    - They create liquidity pools for traders.
    - Can borrow cryptocurrencies (BTC, ETH, SOL, etc.) to expand liquidity.
    - Face risk if their pool collapses.
    - The reputation system tracks reliability and risk-taking behavior.
2. **Traders**
    - Interact with liquidity pools to buy/sell assets.
    - Benefit from extreme market maker failures.
    - Large traders are matched with large pools; small traders with small pools to prevent exploitation.
3. **AI Mediator / Automated Market Maker (AI-AMM)**
    - Oversees liquidity flows and pool health.
    - Redistributes liquidity dynamically to prevent systemic collapse.
    - Controls slashing/burning of corrupted MMs.
    - Ensures smooth execution and minimal slippage for traders.

## 2.2 Liquidity Pool Creation

- MMs pay a fee to create a liquidity pool; larger pools require higher fees.
- They can borrow crypto in N25 to increase pool size.

- ❖ Borrowing is capped over time with a progressive fee, preventing sudden massive liquidity crises.

## 2.3 Pool Management

- ❖ MMs manipulate bid-ask spreads, range, and liquidity allocation to optimize profits.
- ❖ Traders interact with pools to execute trades; if pools are manipulated too aggressively, traders' losses become additional liquidity for the pool ("liquidity farming"), but capped to prevent catastrophic harm.

## 2.4 Pool Merging

- ❖ If an MM is failing, AI merges their pool into a larger, healthier pool.
- ❖ This ensures trader liquidity remains stable and execution quality is preserved.
- ❖ Merged liquidity distributes trader exposure fairly across the surviving MMs.

## 2.5 Risk Management & Rescue

- ❖ Other MMs may choose to help a failing MM to earn N25 rewards.
- ❖ AI monitors the system; if a corrupted MM becomes too contagious, AI can slash or burn liquidity and redistribute remaining assets to surviving MMs.

## 2.6 Reputation System

- ❖ MMs accumulate a dynamic reputation score based on:
  - ➢ Risk-taking behavior
  - ➢ Past rescues or failures
  - ➢ Reliability in pool management
- ❖ Lower reputation decreases the likelihood of rescue in future crises.

## 2.7 AI Governance & Security

- AI mediates all critical operations:
  - Liquidity redistribution

- Slashing/burning corrupted MMs
- N25 incentive allocation
- Pool merging

# 3 Nightcord's Blockchain Design

## 3.1 Nightcord DEX Layer (Oracle)

This system separates execution, governance, and intelligence to stay fast, secure, and scalable. Validators manage public transactions and pool balances while monitoring both on-chain liquidity and off-chain price signals. Prices are formed using on-chain execution with off-chain markets informing fair value. An off-chain AI monitors price divergence, liquidity imbalance, and state growth. It can suggest bounded adjustments, but it cannot override smart contracts. Validators enforce strict deterministic rules, so the AI remains advisory, not controlling. The protocol separates global state (protocol rules, finalized commitments, withdrawals) from local state (pool balances, trades, temporary signals). This reduces coordination overhead and improves scalability. To prevent blockchain bloat, storage rent, per-byte deposits, higher write costs, deletion refunds, and optional pruning make long-term state growth expensive and encourage cleanup. Reflexive attack dampening adds temporary virtual liquidity and extra slippage for large trades when pools become overly sensitive, reducing manipulation and artificial volatility.

### Responsibilities:

- Store public transaction data
- Track liquidity pool balances
- Maintain local state per pool
- Aggregate off-chain and on-chain price inputs
- Allow adaptive risk monitoring (off-chain AI)

### Price Formation

- Price is influenced by:
  - On-chain liquidity
  - Off-chain spot market signals
- AI operates off-chain to monitor optimal bid-ask ranges

Price agreement does not require full global voting at execution level.

### Global vs Local State

- Global State:
    - Protocol parameters
    - Finalized commitments
    - Withdrawal approvals
    - Virtual Liquidity
- Local State:
    - Pool balances
    - User trades
    - Transaction history

This separation reduces coordination overhead.

## State Growth Control

Mechanisms include:

- Storage rent
- Deposit per byte
- Refund on deletion
- Higher write costs than read costs
- Optional pruning policies

Goal:

- Make state expensive to grow
- Incentivize cleanup
- Prevent long-term bloat

## Reflexive Attack Dampening

When local liquidity becomes overly sensitive:

- Protocol applies virtual liquidity dampening
- Large trades face additional slippage
- System absorbs short-term distortions

Goal:

- Reduce manipulability of local pool signals
- Prevent artificial volatility triggering

## 3.2 Nightcord Security Layer

### Responsibilities:

- Store encrypted private transaction data
- Maintain validator voting mechanism
- Provide finality seals for withdrawals
- Commit cryptographic proofs to DEX layer

### Data Confidentiality

Uses:

- Asymmetric encryption (key exchange)
- Symmetric encryption (payload efficiency)
- Cryptographic hashes for commitments
- ZK-proofs for validity without revealing content

### Semi-Global Voting

- 2/3 validator quorum required for:
    - Final state confirmation
    - High-value withdrawal approval
- Validators are prohibited from trading
- Slashing penalties for misconduct

Goal:

- Prevent small-group censorship
- Maintain integrity of final settlement

### Loose Coupling (Pointer-Dependency Model)

Instead of strict synchronous mirroring:

- DEX layer continues producing blocks independently
- Security layer produces finality seals
- High-value withdrawals require finality confirmation
- Execution can continue during security lag

This prevents cross-layer deadlocks.

### Cross-Layer Commitment

Each layer stores:

- Cryptographic commitment (hash pointer)
  to the other layer's finalized block

This ensures:

- Detection of invalid isolation
- Tamper evidence
- Coordinated finality without full duplication

# 4. Nightcord Block Design

## 4.1 Execution Block (DEX Layer)

Each Execution Block contains:

- Previous Block Hash
- Timestamp
- Block Height
- Merkle Root of Transactions
- State Root (post-execution commitment)
- Validator Signature
- Randomness / Nonce (if applicable)

Inside the block:

- Trade transactions
- Liquidity updates
- Parameter updates
- Borrowing/repayment operations
- Fee distributions

## 4.2 Security Layer Block

Each Security Block contains:

- Previous Security Block Hash
- Commitment to Execution Block Hash

- Finality Seal (quorum signatures)
- ZK validity proofs
- Encrypted transaction payloads

The Security Layer does not execute trades.
It verifies and seals finalized state transitions.

# 5. Tokenomics

## 5.1 Nightcord Native Asset (N25)

N25 is the native utility asset of the Nightcord network. It functions as:

- The staking asset for validator participation
- The fee asset for transaction execution
- The collateral asset for liquidity operations
- The economic alignment mechanism for network security

N25 is not positioned as a security instrument. Its primary purpose is to coordinate incentives and secure protocol integrity.

### Core Functions of N25

1. Network Security (Staking)

Validators must stake N25 to participate in block production and finality on the Security Layer. Staked tokens are subject to:

- Slashing for malicious behavior
- Lock-up periods
- Quorum-based validation rewards

This aligns economic incentives with honest participation.

2. Execution Fees

All Execution Layer transactions require N25 to pay:

- Trade execution fees
- Liquidity adjustment fees
- State storage fees
- Withdrawal finalization costs

Fees may be partially burned or redistributed to validators, depending on governance parameters.

3. State Growth Control

Persistent on-chain storage requires a deposit in N25 proportional to the bytes stored.

- Deposits are refundable upon deletion.
- Write operations cost more than reads.
- Long-term unused state may incur rent.

This embeds economic discipline into state expansion.

4. Liquidity and Collateral

N25 may serve as:

- Base pair currency for trading
- Collateral for liquidity provisioning
- Borrowing collateral within the protocol

This creates internal utility demand beyond speculation.

## 5.2 Supply Model

Nightcord uses a structured supply model combining:

- Staking rewards (controlled issuance)
- Fee burning (deflationary pressure)
- Governance-adjustable parameters

### Controlled Issuance

New N25 tokens are issued as staking rewards at a predefined rate that:

- Declines over time
- Is bounded by governance rules
- Cannot be arbitrarily toggled based on short-term demand

### Fee Burning Mechanism

A portion of transaction fees is permanently burned. This:

- Offsets staking inflation

- Creates long-term supply discipline
- Aligns network usage with token scarcity

This model mirrors proven economic structures (e.g., EIP-1559-style burn logic).

## 5.3 Genesis and Initial Distribution

The network begins with:

- A fixed initial supply
- Allocation for validator bootstrap
- Ecosystem development reserve
- Long-term treasury

Distribution must be transparent and time-locked to prevent early concentration risks.

## 5.4 Economic Stability Philosophy

Nightcord does not attempt to directly control market price through artificial supply toggling.

Instead, stability emerges from:

- Real utility demand
- Fee-based burn mechanics
- Staking-based security
- State-cost discipline
- Sustainable issuance

Market value is determined externally through open exchange.

References and Citations:

1. Bashir, Imran. *Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more*. 4th ed., Packt Publishing, 2023.
2. Nakamoto, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 31 Oct. 2008, bitcoin.org/bitcoin.pdf. Accessed 11 Dec. 2025.
3. "Decentralized finance." *Wikipedia*, Wikimedia Foundation, 4 Dec. 2025, en.wikipedia.org/wiki/Decentralized_finance. Accessed 11 Dec. 2025.
4. Yakovenko, Anatoly. *"Solana: A new architecture for a high performance blockchain v0.8.13"*. 2017. *Solana*, solana.com/solana-whitepaper.pdf.
5. Buterin, Vitalik. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." *Ethereum Whitepaper*, 2014.