Ø NØNOS

NØNOS

Sovereignty From Ø
A Decentralized Future

Litepaper v1.0
By Erik (Founder)

12/2025

nonos.systems

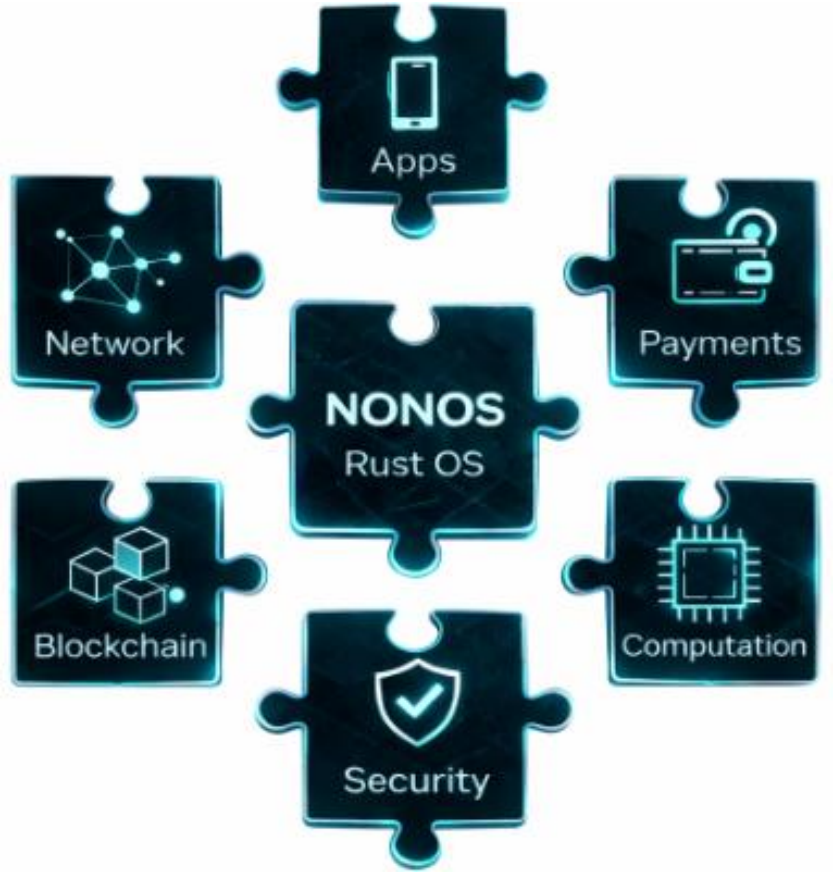
Table Of Contents

Important notice: This litepaper is for informational purposes only and does not constitute financial, legal, or investment advice. NOX is intended as a utility token for network participation and services. Nothing in this document is an offer to sell or a solicitation of an offer to buy any asset. Always do your own research and comply with applicable laws in your jurisdiction.

1. Executive Summary

NØNOS is a zero-trust, Rust-built operating system designed to become the secure endpoint layer for Web3. Today, most wallets, signing workflows, DeFi operations, and sensitive Web3 work still run on general-purpose operating systems (Windows/macOS/Linux) that are persistent, complex, and frequently targeted - which makes the endpoint the weakest link.

NØNOS aims to deliver a clean, private computing session that can erase itself when you're done, while also integrating Web3-native primitives (wallet, privacy networking, and verifiable execution concepts).
Alongside the operating system, the NØNOS ecosystem introduces a proof-native, decentralized infrastructure layer within built micro-fees for services such as capsule execution, verification, and distribution. The NOX token underpins this economy by enabling staking, rewarding infrastructure contribution, and coordinating liquidity and treasury functions.



Core claim: Nothing like this exists today. A purpose-built operating system that is explicitly designed to interconnect Web3 systems securely at the endpoint.

1.1. Key Benefits

- Zero-trust core with minimized attack surface and least-privilege access.
- Runs primarily in memory; shutdown aims to leave nothing behind by default.
- Anonymous, onion-routed mesh networking designed to reduce network-level tracking.
- Integrated Ethereum wallet and privacy-preserving login primitives (zkAuth).
- Micro-fee + Proof-of-Infrastructure incentives for decentralized Operator Nodes.
- NOX tokenomics with fixed supply, fee-based burn, and deflation controls.

1.2. High Level Snapshot

- Bootable, diskless-by-default environment (run from USB; no installation required).
- Memory-first / stateless sessions: reduced persistence reduces long-lived compromise paths.
- Rust microkernel design to shrink the trusted computing base and reduce memory safety bugs.
- Signed, permissioned “capsule” applications (explicit permissions, verifiable integrity).
- Privacy networking designed in: onion-routed mesh capability for metadata resistance.
- Token-aware entitlement layer (NOX) to enable access control, restrictions, staking, and governance.

2. Vision & Core Principles

NØNOS is building toward sovereign computing: secure and private digital workspaces where individuals and organizations can interact with Web3 and the modern internet without treating the operating system as a liability.

2.1. Vision outcomes

- Universal access to secure, private computing (simple boot-from-USB distribution).
- Crypto and banking safely anywhere (even on untrusted or shared devices).
- Trusted endpoints through secure devices (optional secure USB hardware with secure element and verified boot).

2.2. Core principles

- **Zero trust by default:** No process, app, or service is implicitly trusted. Access must be explicitly granted and continuously constrained.
- **Least privilege via capabilities:** Sensitive actions are gated by scoped permissions (capabilities), not ambient OS privileges.
- **Ephemeral by default (ZeroState):** Sessions are designed to minimize residue: less persistent state means fewer long-lived compromise paths.
- **Verifiable security posture:** Favor cryptographic verification (signed modules, integrity checks) and auditable controls over implicit trust.
- **Commercial realism:** Adopt licensing patterns enterprises already understand per-seat, per-module, and concurrent licensing via an entitlement engine.

3. The Problem

Most mainstream operating systems prioritize convenience, legacy compatibility, and monetization over strict security boundaries. Even when users adopt antivirus tools, VPNs, and password managers, these layers still sit on top of a base OS that accumulates logs, caches, and artifacts that can be recovered, exfiltrated, or used for profiling.

At the same time, adversaries are evolving: malware can be generated or adapted rapidly, phishing and deepfakes are increasingly convincing, and attackers can capture encrypted data today with the intention of decrypting it later as cryptographic capabilities improve. The result is a widening gap between what users assume is private and what their devices actually leak.

Web3 security is often discussed at the protocol level, but many real-world failures originate at the endpoint. Traditional operating systems are large, complex, and persistent - a combination that creates broad attack surfaces and durable footholds for malware.

3.1. Failures In Traditional OS Environments

- Persistent traces of system logs, temp files, browser artifacts, and crash dumps remain on disk.
- Large attack surface in legacy subsystems and privileged services expand exploitable code paths.
- Memory-safety risks bugs in low-level languages can lead to remote code execution and privilege escalation.
- Network-level tracking and IP exposure and metadata leakage enable profiling and censorship.
- Credential, key theft and malware targets clipboard data, browser sessions, and locally stored secrets.

3.2. Failures In Existing Web3 Endpoints

- Key material and sessions live on endpoints (wallets, browser extensions, password managers, 2FA).
- Persistent storage leaves forensic and telemetry residue (logs, caches, crash dumps, history).
- Malware targets the user layer (clipboard and screenshot stealers, session hijacking, keyloggers).
- Network metadata can expose who you are and where you connect (Wi-Fi and DNS leakage, traffic analysis).

4. The NØNOS Solution

NØNOS is designed from first principles to reduce data footprints, shrink the trusted computing base, and make security guarantees more explicit. Instead of assuming that "trusted" components remain trustworthy, NØNOS aims to enforce least-privilege access and to rely on cryptographic verification as deeply as possible.

4.1. Key product characteristics

- Memory-only default which runs primarily in RAM; avoids writing user activity to disk by default. (ephemeral sessions by default).
- Rust-built microkernel using a memory-safe language for core components to reduce entire classes of exploits.
- Secure boot and integrity verification to reduce supply-chain and boot-time compromise.
- Signed capsule applications with declared permissions; unsigned or tampered code is rejected.
- Zero-trust permissions for applications and services to receive only the capabilities they need.
- Anonymous networking via built-in onion-routed mesh networking to reduce IP/metadata exposure.
- Integrated crypto + zkAuth and built-in Ethereum wallet and privacy-preserving authentication primitives.
- Post-quantum ready designed to adopt post-quantum cryptography at the OS level.

4.2. Trust and verification layers (conceptual)

NØNOS structures trust as a layered chain. The goal is to establish integrity early (at boot) and extend it through system services and user processes using cryptographic verification and capability enforcement.

Layer	Component	Verification / Control
0	UEFI Firmware	Hardware root of trust (platform security features)
1	Bootloader	Signature verification (e.g., Ed25519)
2	Kernel	Integrity checks (e.g., BLAKE3) + proof mechanisms
3	System Services	Capability tokens / least-privilege permissions
4	User Processes	Behavioral attestation / policy enforcement

5. Ecosystem Overview

NØNOS combines a secure OS with an ecosystem that can include nodes, privacy routing, modular applications ("capsules"), and an economic layer (NOX) that aligns incentives.

5.1. Ecosystem roles

- **Users:** individuals who need secure, private sessions (crypto users, journalists, activists, travelers, analysts).
- **Organizations:** teams and enterprises who deploy hardened endpoints with manageability and compliance needs.
- **Developers:** build capsule apps, integrations, and tools on top of NØNOS.
- **Node operators:** provide infrastructure services where applicable; stake NOX and earn rewards.
- **Partners:** distribution, infrastructure, and ecosystem partners.

5.2. Mesh networking and onion routing

The NØNOS mesh networking layer aims to provide resilient peer discovery and communication. Traffic can be routed through encrypted hops to reduce metadata leakage, support censorship resistance, and decouple services from traditional DNS and clearnet dependencies.

5.3. Capsules (.mod) and Operator Nodes

The ecosystem introduces capsules (portable application packages) and Operator Nodes that execute, seed, and verify capsule workloads. Nodes maintain a local micro-fee ledger and can operate in an offline-first mode, settling periodically on-chain via batching.

Common Operator Node roles (conceptual)

Role	Core function	Typical trigger
Seeder	Serves capsule binaries from cache	Capsule request from a mesh peer
Executor	Runs deterministically & produces execution proof	Capsule scheduled locally
Verifier	Validates proofs and updates trust scores	Proof relay received from peers
Broadcaster	Propagates proof receipts through the mesh	New proof produced locally

5.4. Proof-of-Infrastructure and micro-fees

NØNOS adopts a micro-fee model for network services (for example: capsule installs, verified execution, seeding, and verification). The intent is to create sustainable incentives for distributed operators without relying solely on inflationary emissions.

5.5. zkAuth and privacy-preserving identity

zkAuth is a privacy-preserving authentication layer intended to let users prove access or authorization without revealing passwords or personal identity. This supports passwordless flows and reduces the exposure of sensitive credentials.

5.6. Developer integrations (Anyone SDK)

NØNOS is designed to be composable: developers can integrate privacy-preserving networking and execution features into applications and services via SDK tooling. The goal is to enable ‘privacy as infrastructure’ rather than as an optional add-on.

6. NOX Token Overview

NOX is the native utility token designed to power participation, incentives, and micro-fees within the NØNOS ecosystem. It is intended to coordinate decentralized Operator Nodes, support sustainable infrastructure growth, and provide an economic layer for network services.

Token specification (as published)

Field	Value
Token name	NOX
Network	Ethereum Mainnet
Standard	ERC-20
Total supply	800,000,000 NOX (fixed cap)
Contract address	0x0a26c80Be4E060e688d7C23aDdB92cBb5D2C9eCA

6.1. Core utilities

- **Micro-fees:** Pay for capsule execution, seeding, and verification services within the network.
- **Staking:** Stake NOX to participate as an Operator Node and align incentives with honest behavior.
- **Network incentives:** Reward contributions of compute, bandwidth, and storage via Proof-of-Infrastructure.
- **Treasury and governance (planned):** Support DAO-driven parameter updates and long-term ecosystem funding.

6.2. Smart-contract design notes

The published token contract includes modern conveniences (such as permit-style approvals) and is designed with upgrade controls (timelocked administration) intended to balance security with the ability to evolve protocol parameters under governance.

7. Tokenomics

7.1. Total supply philosophy

NOX is designed with a fixed, hard-capped maximum supply of 800,000,000 tokens. The intent is to avoid perpetual inflation and instead fund incentives through a combination of genesis allocations, micro-fees, and treasury-directed programs.

7.2. Genesis allocation (published)

The genesis distribution assigns supply across community distribution, liquidity provisioning, staking rewards, development, marketing, and ecosystem support wallets.

Allocation	Percent	Tokens	Purpose (summary)
Airdrop to Holders	75.0%	600,000,000	Broad community distribution
Liquidity Collector	5.0%	40,000,000	Protocol-owned liquidity & market operations
Staking Vault	4.0%	32,000,000	Bootstrap staking rewards
Dev Wallet	3.0%	24,000,000	Development & engineering
DAO Wallet	3.0%	24,000,000	Treasury for governance-directed initiatives
CEX Listings Wallet	4.0%	32,000,000	Exchange listing liquidity & operational needs
Contributor & Node Operator	3.0%	24,000,000	Ecosystem contributors & operator support
Marketing Wallet	2.5%	20,000,000	Growth & marketing programs
NFTs Wallet	1.5%	12,000,000	NFT ecosystem & incentives

7.3. Transaction taxation framework

NOX applies a base transaction tax on buys and sells (with no tax on ordinary wallet-to-wallet transfers). Collected fees are split across liquidity operations, development, the DAO treasury, and burns.

Transaction type	Base tax rate
Buy	2%
Sell	2%
Wallet transfer	0%

Fee destination	Share of collected fees	Use
Liquidity Collector	40%	Strengthen protocol-owned liquidity
Developer Wallet	30%	Fund ongoing development; may redirect to staking
DAO Treasury	20%	Governance-directed ecosystem initiatives
Burn	10%	Deflationary reduction of circulating supply

7.4. Adaptive velocity-based deflation

In addition to the base burn component from transaction fees, NOX introduces an adaptive burn mechanism that can increase deflation during periods of high trading velocity. This mechanism is designed to be governance-tunable and capped to limit adverse user impact.

- **Additional burn:** Up to an additional 2% can be applied as a burn component during high-velocity periods (as defined by the protocol).
- **Caps and safeguards:** Buy, sell, and transfer tax rates are individually capped (e.g., at 10%), and combined deductions are capped (e.g., at 20%).

7.5. Liquidity strategy

Liquidity is treated as protocol infrastructure. A designated liquidity collector accumulates NOX and can deploy it into protocol-owned liquidity (POL) positions to deepen markets and reduce volatility. The long-term strategy targets durable liquidity rather than short-lived incentives.

7.6. Staking economics

Staking is intended to align long-term participants and Operator Nodes with the health of the network. The published model includes a staking vault seeded at genesis and a time-lock multiplier system that rewards longer commitments.

Lock duration	Multiplier	Bonus shares
Flexible (0 days)	1.0×	0%
3 months	1.1×	10%
6 months	1.2×	20%
1 year	1.4×	40%
2 years	1.8×	80%
3 years	2.2×	120%
5 years	2.7×	170%
10 years (max)	3.0×	200%

Reward funding is intended to be sustainable: the staking vault is initially seeded (32M NOX at genesis) and can be supplemented via treasury decisions, fee redirection, and external revenue sources, rather than requiring perpetual inflation.

8. Go-To-Market

NØNOS is designed to be accessible to everyday users while also providing a path for professional and enterprise deployments. The go-to-market strategy combines a free community OS distribution with optional hardware and premium service tiers.

8.1. Primary distribution and revenue channels

- **Community Edition (free):** Core OS available at no cost to maximize adoption and community feedback.
- **Official secure USB hardware:** Optional pre-loaded device for plug-and-play use and additional hardware security.
- **Pro Edition subscription:** Advanced features, automated updates, management, and priority support for power users and businesses.
- **Enterprise licensing and services:** Fleet management, compliance-ready features, and custom integrations for regulated industries.

9. Governance

The tokenomics framework anticipates governance-driven evolution. Parameters such as fee splits and adaptive deflation controls can be updated through governance processes, with explicit caps designed to limit abusive or unexpected changes.

9.1. Governance scope (intended)

- Treasury management (DAO wallet) and ecosystem funding programs.
- Liquidity operations policies (protocol-owned liquidity deployment).
- Adjustments to adaptive deflation parameters within predefined caps.
- Long-term upgrades to the token or staking systems under timelocked controls.

10. Roadmap

This litepaper is a snapshot of an evolving project. The roadmap below summarizes major workstreams for the OS, ecosystem infrastructure, and token mechanics.

10.1. Near-term priorities

- Expand early-access distribution, documentation, and onboarding flows.
- Continue hardening the boot chain and core security subsystems; pursue independent audits where applicable.
- Ship initial Operator Node tooling and micro-fee ledger settlement paths.
- Stabilize staking vault UX and security practices; publish clear governance processes.

10.2. Mid-term objectives

- Broaden hardware support (desktop/laptop focus first, then additional device classes).
- Mature the capsule runtime and .mod distribution economics.
- Expand privacy-preserving identity (zkAuth) and proof-bound execution capabilities.
- Introduce Pro Edition features and enterprise deployment tooling.

10.3. Long-term objectives

- Scale mesh networking and onion routing resilience across diverse network conditions.
- Advance Proof-of-Infrastructure economics and sustainable operator incentive programs.
- Progressively decentralize governance and treasury operations through the DAO.
- Explore cross-chain settlement and liquidity parity programs as the ecosystem expands.

11. Team and Partners

NØNOS is built by a team spanning operating systems, cybersecurity, and Web3 commercialization, supported by advisors and industry partners.

11.1 Founding team (publicly listed)

- Ek (Erik) - Founder
- Ironhastag (Andrea) - CEO & Co-founder

11.2 Advisors (publicly listed)

- Maxi - Growth Strategist
- CryptoBobRoss - Business Development Advisor
- Pano - Creative Officer
- Daniel & Maxime - Strategic Advisors
- Eric Jordan - Advisor (privacy-centric tech)

11.3 Selected partners (publicly listed)

- PAAL
- Hotspotty
- Messier
- DGRS
- MAXKO Hosting
- DeBros
- IQ Wiki
- Cellframe

12. Risks and Disclosures

NØNOS and NOX are part of an evolving ecosystem. Users and participants should understand that both software systems and token networks carry meaningful risks.

Key risk categories:

- **Technology maturity risk:** Components may change materially during development; features described may evolve.
- **Security risk:** Despite strong design goals, all software may contain vulnerabilities; audits reduce but do not eliminate risk.
- **Smart contract risk:** Token and staking contracts may have bugs or economic edge cases.
- **Market risk:** Token prices can be volatile; liquidity conditions may change.
- **Regulatory risk:** Regulatory treatment of tokens varies by jurisdiction and can change over time.
- **Governance risk:** Governance decisions can affect parameters, incentives, and protocol behavior (within stated caps).

Nothing in this document should be interpreted as a guarantee of performance, security, or future value.

Always verify contract addresses and official communications through trusted channels.

Thank you!

From the NONOS Team