

NYX

Whitepaper

Version 1
January 2026

Notice

This document is a technical whitepaper. It specifies enforceable rules, explicit assumptions, and hard boundaries for the NYX protocol system.

- This document is not investment advice.
- The protocol described herein is experimental software.
- No party provides any warranty or guarantee of correctness, fitness, availability, or security.

Where requirements are stated, they are intended to be testable.

Table of Contents

1. Introduction
 2. Terminology and Conventions
 3. System Requirements
 4. Threat Model and Assumptions
 5. Architecture
 6. Identity System (NYX ZK-ID)
 7. Web2 Gateway Boundary
 8. Cryptography Requirements
 9. Key Management Requirements
 10. Economic Requirements
 11. Governance Requirements
 12. Compliance Criteria
 13. Risks and Limitations
 14. Glossary
-

1. Introduction

NYX is a protocol system designed to operate in a permanently adversarial environment. This whitepaper defines the minimum constraints that any NYX v1 implementation MUST satisfy.

This document is normative: it uses mandatory language to express requirements and prohibitions.

1.1 Scope

In scope:

- system axioms and non-negotiable prohibitions
- adversary model and explicit non-goals
- a five-layer architecture with a one-way dependency law
- a protocol-native identity system (ZK-ID) and lifecycle invariants
- a Web2 gateway boundary and forbidden behaviors
- cryptographic selection constraints
- key-management responsibility boundaries
- economic constraints for fee enforcement and emergency semantics
- governance scope limits and anti-capture constraints

Out of scope:

- client UI/UX design
- growth, promotional, or narrative materials
- token sales, price targets, yield programs, or financial promises
- deployment operations and rollout procedures
- jurisdiction-specific compliance as a core protocol dependency

Any out-of-scope item **MUST NOT** be used to reinterpret or weaken in-scope constraints.

2. Terminology and Conventions

2.1 Normative Language

The key words **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT**, and **MAY** are to be interpreted as described in RFC 2119.

2.2 Core Terms

- **NYX Identity**: protocol-native subject that accumulates permissions and constraints.
 - **Wallet**: replaceable key-management interface; not identity.
 - **Context**: domain-separated proof environment.
 - **Shared-state mutation**: any action changing protocol state visible to more than one participant.
 - **Privileged path**: any advantage not available under the same public rules (fee, inclusion, execution, identity).
 - **Emergency mode**: restrictive state that can only remove capabilities (pause/slow/raise-cost).
-

3. System Requirements

This section defines system-level rules that are non-negotiable in v1.

3.1 Protocol-First Enforcement

Core security, legitimacy, and continuity **MUST** be enforced by protocol structure rather than discretionary human intervention.

3.2 Identity Requirements

- The only valid subject inside NYX is a NYX Identity.
- A wallet, address, account, or keypair **MUST NOT** be treated as identity.
- A controller **MAY** operate multiple identities.
- NYX Identity **MUST NOT** require binding to real-world persons as a core dependency.

3.3 Wallet Requirements

- A wallet is a replaceable key-management interface.
- Losing or rotating a wallet **MUST NOT** imply losing a NYX Identity.

3.4 Fee Requirements

- Any shared-state mutation recognized by the protocol **MUST** incur a non-zero, protocol-enforced fee.
- Fee bypasses, privileged exemptions, or “free lanes” are forbidden.

3.5 Web2 Requirements

- Web2 inputs **MUST** be treated as adversarial and non-authoritative.
- All Web2 access in the NYX context **MUST** be mediated by the NYX Gateway.
- Web2 **MUST NOT** back-propagate authority into core protocol logic.

3.6 Decentralization Requirement

Optimizations that weaken decentralization or introduce irreversible power concentration are invalid.

3.7 Reputation Requirement

Credit/reputation **MUST** be generated exclusively from protocol-defined behavior and **MUST** be non-transferable. Tradable or transferable reputation markets are forbidden.

3.8 Hard Prohibitions

NYX **MUST NOT**:

- treat wallets, addresses, accounts, or keys as identity
 - rely on trusted administrators, discretionary recovery, or “support override”
 - provide zero-fee shared-state actions
 - use Web2 as a source of truth
 - allow silent rule changes (implicit upgrades or undocumented behavior changes)
 - introduce permanent admin keys or unremovable privileged roles
 - treat token ownership as default legitimacy for governance
-

4. Threat Model and Assumptions

4.1 Adversary Classes

NYX assumes persistent adversaries including:

1. nation-state actors
2. platform-level adversaries
3. onchain analytics and surveillance firms
4. external hackers
5. internal malicious actors

4.2 Environmental Assumptions

- Centralized infrastructure can be compromised.
- Adversaries are not assumed benevolent, rational, or passive.
- Modern cryptography is assumed intact (no practical breaks).
- Complete global suppression of all decentralized participants is not assumed always achievable.

4.3 Explicit Non-Goals

The protocol does not guarantee:

- prevention of all user losses from phishing, malware, or key compromise
- anonymity against voluntary disclosure or user operational mistakes
- jurisdiction-specific compliance or surveillance accommodation
- uninterrupted availability under global coordinated shutdowns
- graceful degradation under cryptographic failure

Missing these properties is not considered protocol failure.

5. Architecture

5.1 Five-Layer Model

NYX is structured into five layers with strict responsibility boundaries:

- **L0: Identity & Cryptography**
- **L1: Chain / Consensus**
- **L2: Currency & Settlement**
- **L3: Markets & Exchange**
- **L4: Entry Software (Deferred)**

5.2 One-Way Dependency Law

- Dependencies **MUST** flow strictly upward: a layer **MAY** depend only on lower layers.
- Cycles and bidirectional dependencies are forbidden.

5.3 Cross-Layer Invariants

- Identity and privacy invariants **MUST** be enforced at or below L0 and **MUST NOT** depend on Web2.
 - Governance **MUST NOT** be able to alter identity state.
 - Fee enforcement **MUST** be applied to all shared-state mutations.
-

6. Identity System (NYX ZK-ID)

6.1 Definitions

- **ZK-ID**: zero-knowledge proving interface bound to a NYX Identity.
- **Credential**: signed assertion usable in zero-knowledge proofs.
- **Claim**: provable statement.

6.2 Security Invariants

The identity system **MUST** satisfy all of the following:

- A wallet/address/account/keypair **MUST NOT** be used as an identity identifier, root, or proxy.
- Reputation **MUST NOT** be transferred, sold, leased, merged, replayed, or reused across identities.
- Proofs generated in different contexts **MUST** be cryptographically unlinkable by default.
- No administrator/operator/issuer/recovery agent **MAY** possess unilateral authority to alter identity state.
- Any identity action that creates, mutates, or preserves protocol state **MUST** incur a non-zero, protocol-enforced cost.

6.3 Lifecycle Requirements

The identity lifecycle **MUST** support:

- **Generation**: local generation of required secrets; no stable identifier exposure.
- **Usage**: proofs bound to a context; no global identifiers.
- **Rotation**: required privacy primitive; retired material is invalidated for forward unlinkability.
- **Destruction**: irreversible; verification fails permanently thereafter.
- **Recovery**: protocol-defined and non-discretionary; results in new material with explicit lineage semantics.

6.4 Linkability Prohibitions

The identity layer **MUST NOT**:

- reuse stable identifiers, secrets, public keys, commitments, or metadata across identities
- reuse nullifiers/tags/commitments across contexts
- create long-lived identifiers enabling longitudinal tracking
- encode, depend on, or expose network/platform identifiers

6.5 Allowed Claim Classes

Claims MAY include:

- personhood / non-bot assertions without requiring real-world binding
- threshold reputation statements without disclosing exact values
- eligibility statements (including compliance-style claims) as optional modules

Any compliance-style claim MUST be optional and MUST NOT introduce regulatory backdoors.

7. Web2 Gateway Boundary

7.1 Gateway Necessity

All Web2 interactions in the NYX context MUST be conducted exclusively via the NYX Gateway.

7.2 Gateway Invariants

The Gateway MUST:

- wrap and isolate external inputs before any NYX core logic consumes them
- treat Web2-derived identifiers and metadata as hostile inputs
- remain optional and replaceable

7.3 Forbidden Gateway Behaviors

The Gateway MUST NOT:

- create or preserve identity linkage across sessions, sites, or contexts
- bypass or weaken external anti-abuse controls
- facilitate regulatory evasion
- simulate human behavior or adapt against platform detection

7.4 Boundary Rule

Gateway mediation is a privacy and safety boundary; it MUST NOT become a general-purpose proxy.

8. Cryptography Requirements

8.1 Primitive Selection Policy

- NYX MUST NOT design or implement new cryptographic primitives.
- NYX MUST use widely deployed and publicly reviewed primitives with mature reference implementations.

8.2 Baseline Constraints

- Transport security MUST use modern forward-secure protocols.
- Encryption MUST use AEAD constructions.

- Derivations MUST be domain-separated.
- Identity-level signatures MUST use a modern, widely reviewed scheme.
- ZK proofs MUST be context-bound and output non-identifying artifacts.

8.3 Implementation Constraints

- Crypto dependencies MUST be version-pinned.
 - Build artifacts including cryptographic code MUST have an SBOM.
 - Private-key operations MUST be constant-time.
-

9. Key Management Requirements

9.1 Non-Negotiable Invariants

- NYX MUST NOT rely on administrators/operators/support to preserve or restore identity control.
- Chain transaction keys MUST NOT be treated as NYX identity roots.
- NYX services MUST NOT possess the capability to decrypt subject end-to-end protected content.

9.2 Role Boundaries

- The subject controls identity secrets and initiates lifecycle actions.
- The NYX client performs identity/privacy cryptography locally.
- Operators deploy infrastructure but MUST NOT access user secrets.
- Issuers MAY issue/revoke credentials but MUST NOT control identity state.
- Verifiers validate proofs but MUST NOT demand stable identifiers.

9.3 High-Value System Keys

- Release/config signing keys MUST be protected by quorum custody (M-of-N) with independent custodians.
 - Deployed artifacts and configuration MUST be verifiable via signature.
-

10. Economic Requirements

10.1 Legitimacy Criteria

NYX economics is valid only if:

1. scarcity is priced by objective rules,
2. costs are internalized and non-bypassable,
3. security and critical operations are fundable without discretionary fundraising,
4. economic power does not automatically translate into unconstrained governance power.

10.2 Hard Requirements

- State transitions MUST require non-zero fees.
- Privileged fee bypass is forbidden.
- Hidden subsidies are forbidden.
- Fee routing MUST be auditable by consensus.

- Token ownership MUST NOT imply default legitimacy.

10.3 Emergency Semantics

Emergency mechanisms MUST be restrictive only (pause/slow/raise-cost). They MUST NOT grant new privileges.

11. Governance Requirements

11.1 Governable Scope

Governance MUST be limited to:

1. protocol parameter calibration
2. protocol upgrades
3. protocol treasury/reserves

Anything outside this scope is non-governable.

11.2 Non-Governable Objects

Governance MUST NOT:

- alter identity state or identity lifecycle
- select or censor individual transactions/accounts
- create exemptions to enforcement rules

11.3 Constitutional Supremacy

All governance outcomes MUST remain subordinate to Sections 3–10 and the prohibitions in Section 3.8. Any conflicting governance action is invalid.

11.4 Parameter Classes

Every mutable item MUST be classified as:

- **Frozen:** not changeable within v1
- **Constrained:** changeable only monotonically toward stricter constraints / higher auditability
- **Policy:** tunable numeric calibration

A proposal MUST declare the parameter class for each affected item and MUST NOT bundle unrelated changes.

11.5 Proposal Lifecycle Requirements

- Proposal submission MUST incur non-zero economic cost (fee or bond).
- Proposals MUST be time-delayed before execution (timelock) and cancelable prior to execution.
- Proposals MUST expire if not executed by a declared expiry time.

11.6 Upgrade Requirements

- Upgrades **MUST** be explicit and auditable.
- Upgrades **MUST** declare power delta and rollback feasibility.
- Any upgrade introducing new privileged roles or bypass paths is forbidden within v1.

11.7 Treasury/Reserve Requirements

- Treasury outflows **MUST** be rule-governed, time-delayed, capped, and auditable.
- Treasury **MUST NOT** route protocol value to token holders by default.

11.8 Emergency Governance

Emergency mechanisms **MUST** be negative-power only and **MUST NOT** authorize spending or bypass timelocks.

12. Compliance Criteria

An implementation is **NYX v1 compliant** if and only if it satisfies all **MUST** and **MUST NOT** requirements in this whitepaper.

Minimum conformance tests:

12.1 Identity & Privacy

- Wallets/addresses are never treated as identity.
- Proofs are context-bound and unlinkable by default.
- No stable identifier persists across contexts.

12.2 Web2 Gateway

- No component accesses Web2 services outside the Gateway.
- The Gateway does not bypass anti-abuse controls.
- The Gateway does not preserve cross-session identifiers.

12.3 Economics

- No shared-state mutation is executable with zero fee.
- Fees are objective and auditable.
- No privileged fee exemption exists.

12.4 Governance

- Proposals declare parameter class and cannot execute instantly.
 - Emergency cannot grant privileges or authorize spending.
 - Governance cannot affect identity state.
-

13. Risks and Limitations

This section enumerates constraints and failure modes that are explicitly not prevented by protocol design.

- Users MAY lose funds or control due to malware, phishing, coerced disclosure, or operational mistakes.
- Availability MAY be reduced under network partitions, coordinated infrastructure attacks, or censorship.
- Privacy MAY be reduced by user-side behavior, endpoint compromise, or voluntary linkage.
- Implementations MAY contain defects; correctness is not guaranteed.

These risks do not constitute protocol non-compliance.

14. Glossary

- **NYX Identity**: protocol-native subject controlling proofs and accumulating constraints.
- **Wallet**: replaceable key-management interface; not identity.
- **Context**: domain-separated proof environment.
- **Shared-state mutation**: any action changing protocol state visible to more than one participant.
- **Privileged path**: any advantage not available under the same public rules (fee, inclusion, execution, identity).
- **Emergency mode**: restrictive state that can only remove capabilities (pause/slow/raise-cost).