Nyxi

Execution governance for irreversible actions on Windows

Public overview | No implementation details | 2026-01-27

Execution governance for irreversible actions

Nyxi is a Windows-hosted, long-running execution-governance system designed to control irreversible actions through a single non-bypassable execution boundary.

AI proposes. Nyxi governs.

Single execution boundary

Irreversible actions execute only through one governed checkpoint; upstream reasoning is never sufficient for authority.

Fail-closed decisions

Actions are authorized (ALLOW), blocked (DENY), or halted for guidance (UNKNOWN). UNKNOWN never executes.

Durable evidence

Every decision and execution produces durable audit artifacts to support review, replay, and accountability.

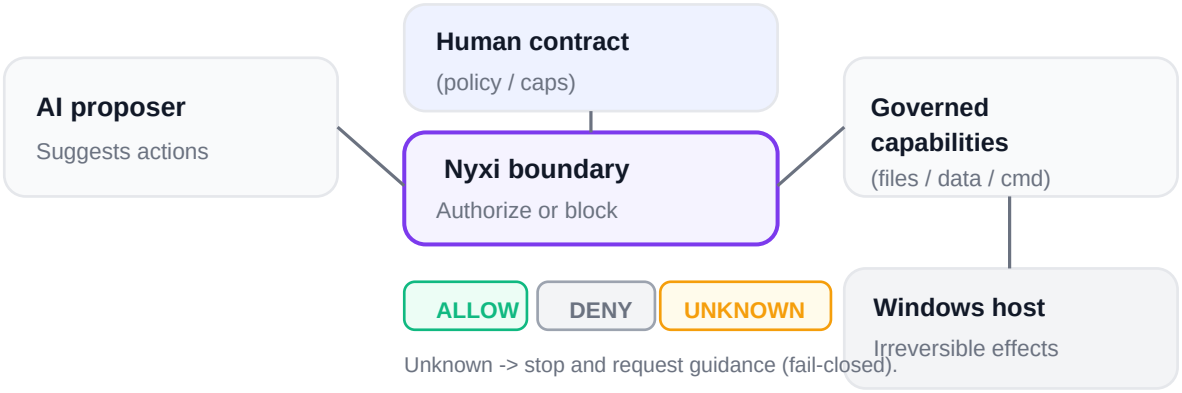
This brief is intentionally high-level. It describes observable guarantees and operating model while omitting implementation details, internal policy syntax, and capability internals.

What Nyxi is

Nyxi is an execution-governance layer that sits at the last responsible moment: the point where an irreversible action would occur. Instead of relying on upstream prompts, reviews, or model confidence, Nyxi enforces a human-defined contract at execution time.

Operating model

- Humans define the contract (scope, invariants, caps).
- AI models can propose actions, but do not have execution authority.
- Nyxi evaluates each proposed irreversible action at the boundary and produces a decision: ALLOW, DENY, or UNKNOWN.
- ALLOW authorizes and executes; DENY blocks; UNKNOWN stops and requests guidance.



Why this matters

Nyxi is built for environments where "we will notice and rollback later" is not acceptable. The system is structured so irreversible actions are vetoable at execution time under explicit invariants, and ambiguity fails safe.

Decision	Meaning
ALLOW	Authorized under the human contract; execution may proceed under caps.
DENY	Blocked by policy; execution is prevented.
UNKNOWN	Policy cannot prove correctness; execution halts and requests guidance (fail-closed).

What Nyxi can govern

Nyxi can be extended to govern Windows workflows when irreversible effects are implemented as governed capabilities routed through the boundary.

Capability categories (examples)

- File operations within explicit scopes (controlled writes, structured appends, patch-style edits).
- Structured data emission (writing verified JSON artifacts).
- Tightly constrained command execution (allowlisted commands with strict limits).

Intent-pack mode

Workflows can be expressed as declarative intent inputs. Nyxi validates intent and clamps it to policy ceilings before any irreversible action is considered.

Where Nyxi fits

- Controlled automation on Windows hosts (repo maintenance, artifact generation, scripted workflows).
- Agentic systems where models propose changes but execution must be contract-governed.
- Governed pipelines needing reproducible decisions and evidence artifacts.

Key differentiators

Differentiator	What it means
Execution-time authority	Irreversible actions are controlled at the boundary, not via post-hoc monitoring.
Separation of duties	Human contract defines scope and limits; AI is proposer-only.
Fail-closed safety	If policy cannot prove correctness, execution halts (UNKNOWN).
Evidence-first design	Produces durable artifacts for later review and accountability.

Limitations and correct expectations

Nyxi is intentionally opinionated: it prioritizes correctness at the irreversible boundary over convenience upstream. The following are deliberate constraints, not gaps.

- Nyxi only governs irreversible effects routed through its execution boundary.
- Policy quality matters; Nyxi enforces the human contract as written.
- Fail-closed behavior trades convenience for safety by design.
- Nyxi is not a model-improvement product; it is an execution authority and enforcement layer.

How to evaluate Nyxi (external-safe checklist)

- Confirm every irreversible action in scope routes through the governed boundary (no side channels).
- Review that the human contract captures required scopes, invariants, and caps for your environment.
- Verify fail-closed behavior: ambiguous inputs stop rather than execute.
- Inspect evidence artifacts: decisions and executions should be auditable and attributable.
- Run adversarial and regression suites to ensure no bypass paths emerge as capabilities expand.

Roadmap direction

Nyxi scales by adding governed capabilities one at a time, each with boundary enforcement, invariants and caps, non-bypassability tests, and evidence logging.

Contact / next steps

If you are evaluating Nyxi for a controlled automation or agentic workflow, prepare a short list of irreversible actions you want governed, plus the invariants/caps you require. Nyxi is designed to grow capability-by-capability without granting execution authority to the AI proposer.

This document is shareable. It omits sensitive details by design.