NIMSFOREST

Operating at the Limits

An Architecture for the Compute-Native Organization

An organization is a social construct for achieving goals. In the intelligence age, compute pursues those goals. Goals are bounded by Goodhart's law; compute is bounded by Amdahl's law. This paper describes a system engineered to operate continuously at both frontiers.

Whitepaper · July 2026
nimsforest.com

Abstract

Every organization that pursues its goals through computation is bounded by two laws. Goodhart's law bounds the goals: any metric, optimized hard enough, diverges from the purpose it was meant to represent. Amdahl's law bounds the compute: any computation with sequential dependencies has a speedup ceiling set by its serial fraction, no matter how much parallel hardware is applied. Neither law can be broken. Most systems ignore one or both, and pay for it in silent drift or wasted scale.

NimsForest is an event-driven organizational operating system built on the opposite premise: that both limits are permanent, that they are different in kind, and that a system should therefore be engineered to run as close to each frontier as physics and judgment permit. The mechanical limit receives a mechanical answer — an architecture that compresses its own serial residue toward the theoretical floor and continuously self-optimizes its resource use. The definitional limit receives the one non-mechanical component in the system — a human-machine seam that translates organizational purpose into deterministic strategic KPIs and audits them against reality.

This paper describes the architecture, shows how each classical hazard of distributed and self-optimizing systems is cornered into a small, inspectable place, and states plainly what the system does not claim to do.

1. The Problem: Two Laws, Two Different Kinds of Limit

1.1 Organizations in the intelligence age

An organization is a social construct for achieving goals. Historically, the pursuit of those goals ran on human labor and coordination. In the intelligence age, an increasing share of that pursuit runs on compute: software agents, learned policies, and automated workflows execute the work that people once did, at machine speed and machine scale. This shift does not remove the organization's fundamental constraints. It relocates them — from human throughput and human politics to two laws that govern any goal-pursuing computation.

1.2 Goodhart's law: the limit on goals

When a measure becomes a target, it ceases to be a good measure. Every deterministic KPI is a lossy compression of a richer purpose, and an optimizer will exploit precisely the loss: it will satisfy the metric in ways that diverge from the intent, because that divergence is free reward. The more powerful the optimizer, the more ruthlessly the gap is found. Critically, no metric can detect its own divergence — if it could, the detection would be folded into the metric and the gap would close.

Guarding against Goodhart is therefore structurally not a computable task. It requires judgment from outside the metric.

1.3 Amdahl's law: the limit on compute

Formulated by Gene Amdahl in 1967, the law states that a program's maximum speedup is 1/S, where S is its inherently sequential fraction. If five percent of the work is a dependency chain — each step requiring the previous step's result — then infinite parallel hardware yields at most a twenty-fold speedup. This is mathematics, not engineering: data dependencies are real, and no hardware repeals the requirement that B wait for A's result. The practical consequence in the modern era is stark. Parallel throughput (GPUs, distributed workers) scales almost without bound, while the sequential critical path is pinned to single-unit latency, which has been nearly flat for two decades. The serial residue is the permanent tax on every computation.

1.4 The asymmetry that shapes the design

These two limits are not the same kind of thing, and this asymmetry is the foundation of the architecture. Amdahl is a mechanical limit on how well a given target can be pursued: it is measurable, and approaching it is unambiguously good — a machine can and should drive toward it automatically. Goodhart is a definitional limit on whether the target is the right one: it is not measurable from inside the system, and a machine is actively dangerous here, because the harder it optimizes, the more aggressively it exploits any gap in the target's definition. A sound system therefore places the machine where more optimization is always better — pursuit — and places human judgment where more optimization is a hazard — definition. NimsForest is built around exactly that placement.

2. The Architecture

2.1 Two planes: transport and truth

The system separates events from state. Events are pure signal — ephemeral notifications that something changed. State is authoritative truth, held in an atomic, eventually consistent key-value store. Events never carry the truth; they are hints that prompt consumers to consult it. This single separation is what makes the entire system robust to message loss: a dropped event costs one tick of staleness, never a corrupted world, because the next read of authoritative state corrects everything.

2.2 The event plane: fire-and-forget by design

Task events travel over core NATS with at-most-once delivery and no persistence — deliberately. Persistence is confined to the boundary, where JetStream captures events arriving from the outside world; the unreliable exterior must be recorded reliably, but the interior simulation runs on ephemeral, best-effort, always-forward state. The model comes directly from real-time game-engine rendering: transient frame state is never persisted, a dropped event is a dropped frame superseded by the next tick, and replaying stale causal state would be actively wrong.

Equally deliberate is the prohibition on waiting. No worker blocks on another; no component may construct a distributed barrier or a join. This makes the classic fan-in reduction — the logarithmic tree of dependent combining steps that dominates the serial cost of parallel batch systems — structurally impossible rather than merely discouraged. The system trades away the ability to ask "is everything done?" in exchange for never suffering a synchronization stall. It is the same bargain a frame loop makes, applied to an organization.

2.3 The state plane: truth compressed to a compare-and-swap

Authoritative state lives in the NATS key-value store: durable, atomic per key through monotonic revisions and compare-and-swap updates, eventually consistent across the cluster. There are no multi-key transactions, by design — each unit of authoritative state is modeled so its consistency boundary fits in one key. Consumers are level-triggered: they react to current state rather than depending on having witnessed every transition, which is what makes at-most-once delivery safe. The consequence is notable: the only sequential primitive the system permits anywhere is a single per-key compare-and-swap — a revision bump. This is approximately the smallest unit of "B must see A's result" that a system with shared state can have. The serial residue that Amdahl guarantees must exist somewhere has been compressed to its practical floor and pinned where it can be inspected.

2.4 Workers: those who execute and those who decide

Routing is subject-based, in the NATS idiom: workers subscribe to specific subjects, and the subject hierarchy is the filter. An event reaches only its subscribers; for everyone else it costs nothing. Among subscribers, workers divide into two classes. Deterministic workers do not decide — they execute. An event on their subject arrives and they run: same input, same action, every time. They are the analyzable substrate, the fixed systems of the engine. Nondeterministic workers decide. They receive an event, may read — never write — the key-value store for context, and choose whether and how to act. These are the learning agents, and they are where adaptive behavior lives.

The read-not-write rule for decisions is a load-bearing safety property. Causal chains extend only through emissions — an event published, a subject-bearing state write. A read is a causal dead-end. By confining decision-making to reads, the act of deciding can never silently deepen the causal graph; only deliberate emission can, and deterministic emitters form a static, inspectable skeleton of subject-in to subject-out edges. Nondeterminism is quarantined to one worker class, and its only channel of influence on the graph's shape is the governed vocabulary described next.

2.5 The codex: a governed vocabulary of cause and effect

All subjects belong to a commonly shared codex, and new subjects are added only deliberately. This is the structural bound on emergent behavior. A learning worker can become arbitrarily sophisticated about whether and when it emits, but it can only emit into the vocabulary that exists — so the set of possible causal edges in the entire system is finite, curated, and reviewable. A potential feedback loop is a statically detectable cycle in the subject graph, not a runtime surprise. The moment a subject is added is the moment a new causal edge enters the universe, and the governance ceremony around that addition is, in effect, a causal-graph review. The codex fixes what connections can exist; the workers freely determine which fire, and when.

2.6 Self-optimization: the system minimizes its own residue

NimsForest is a real-time, nondeterministic system that continuously optimizes toward its organizational goals with fewer resources: a minimal codex, minimal time per worker from event received to event triggered, and minimal token expenditure. These three quantities are not arbitrary efficiency metrics. Codex size is the count of possible causal edges — the graph's structural complexity. Per-hop latency is the constant on every edge of any causal traversal. Token use is the cost of each node's work. Their product is the total propagation cost of the system's behavior. In other words, the optimization target is the system's own Amdahl residue: the thing the law says cannot be abolished is instead made continuously self-reducing toward its floor. The codex, in particular, is under endogenous pressure — a new subject must earn its place by improving goal-achievement more than it costs in the very budget being minimized — so the vocabulary's minimality is enforced by the objective function itself, not only by human discipline.

2.7 Strategic KPIs: the human-machine seam

Resource minimization alone is a hazard: an optimizer told only to spend less will find the cheapest structure that appears to satisfy its goals. The system therefore contains one role whose purpose is to translate the organization's purpose into deterministic strategic KPIs — a North Star metric decomposed, in the manner of an entrepreneurial operating system, into the KPI cascade the workers optimize against. This role is a composite of human and machine, and the division of labor within it mirrors the paper's central asymmetry. The machine half renders judgment into a crisp, deterministic signal at the same fidelity as the resource budget, so that goal-achievement and cost-reduction compete as equals in the objective and the optimizer cannot drift toward whichever is easier to measure. The human half performs the one act no metric can perform on itself: noticing when the KPIs, though perfectly satisfied, have stopped meaning the purpose — when the world has moved, when the optimizer has found an exploit, when an unencoded value is being traded away. The human owns meaning; the machine owns expression; the swarm owns pursuit.

3. Every Hazard, Cornered

The design's coherence is easiest to see as a single pattern: each hazard that distributed-systems theory or optimization theory says must exist somewhere is deliberately compressed into the smallest, most inspectable place it can occupy — never ignored, never allowed to spread.

Hazard	Where it would normally live	Where NimsForest corners it
Lost messages	Corrupted or diverging state; replay complexity	Truth lives in the KV store; events are hints. A drop costs one tick of staleness, self-corrected on the next level-triggered read.
Barriers and joins	Log(N) fan-in reductions; distributed "is it done?" logic; stalls	Structurally impossible: no waits, no multi-key joins. The reduction cannot be built.
Torn state	Multi-key transactions and their coordination cost	One invariant per key; the sole serial primitive is a per-key compare-and-swap.
Runaway decisions	Decision logic silently extending causal chains	Decisions may read state, never write it. Reads are causal dead-ends; only emission extends the graph.
Emergent cascades	Unbounded, unauthored event chains and feedback loops	The codex makes the set of possible causal edges finite and reviewable; loops are statically detectable cycles.
Amdahl residue	Hidden serial fractions inflated by coordination overhead	Made the optimization target itself: codex size, hop latency, and token cost are continuously self-minimized.
Goodhart drift	Metrics optimized perfectly while meaning the wrong thing	A human-machine seam owns the purpose-to-KPI translation

Hazard	Where it would normally live	Where NimsForest corners it
		and audits it against reality, outside the fast loop.

The pattern is uniform: nothing is claimed to be abolished. The serial dependency, the possibility of divergence between metric and purpose, the fragility of shared state — all are acknowledged as permanent, and each is pinned to a location small enough to see, govern, and minimize.

4. What This System Does Not Claim

Precision here is not modesty; it is the intellectual content of the proposal. Systems that claim to beat these limits are the systems that silently violate them.

4.1 Amdahl is approached, not beaten

Amdahl's law is a mathematical limit, and NimsForest does not break it. What the system does is asymptotic: it compresses the serial residue to a per-key compare-and-swap, keeps causal chains shallow by construction, refuses to manufacture reductions, and then continuously drives its own remaining residue — edges, hops, and per-node cost — toward the floor. This is riding the limit as closely as a system can, in the way a well-designed engine rides thermodynamics: not past the bound, but wasting as little as possible against it. That is the only honest victory available, and it is a real one.

4.2 Goodhart is guarded, not solved

No metric can be made immune to Goodhart, because the law is a statement about all metrics under sufficient optimization pressure. NimsForest does not close the gap between KPI and purpose; it stations judgment at the seam where the gap opens, so divergence is noticed and corrected faster than it compounds. The gap still opens. The claim is a corrective loop of the highest available fidelity, not immunity.

4.3 Emergent depth is bounded and observable, not eliminated

Because routing is autonomous — each worker independently deciding whether an event concerns it — causal cascades are emergent, data-dependent, and variable in depth. The codex bounds the shape of what can happen; it does not fix what will. A set of individually reasonable local rules can still compose into long reaction chains at runtime, and since no worker waits, this cost never appears as an error — it appears as end-to-end convergence latency. The system's answer is structural bounding plus visibility, not prevention: the deterministic subject skeleton is statically traceable, and cascade length is a runtime property to be observed and minimized, not assumed away.

5. Residual Risks, Stated Plainly

Three risks remain after everything above, and all three concentrate — by design — at the human-machine seam, because that is where the system has deliberately pushed every non-mechanical question.

The first is the rubber stamp. The machine half of the strategic-KPI role operates at machine cadence; the human half at human cadence. If the machine proposes adjustments faster than the human can genuinely evaluate, the human's role degrades from judging to approving — present on the org chart, absent in practice — and Goodhart protection silently becomes zero while appearing intact. The mitigation is the same minimization discipline applied everywhere else: the human's decision surface must be kept small and high-leverage, so that judgment remains real rather than nominal.

The second is KPI staleness. The fast loop optimizes against the current strategic KPIs with increasing precision; whether those KPIs still mean the organizational purpose is a slower question that lives outside the loop. Without a deliberate re-translation cadence — actual outcomes fed back to the human seam, grading the KPIs themselves rather than performance against them — the system converges confidently on yesterday's purpose.

The third is proxy incompleteness. A deterministic KPI set is a lossy compression of a purpose, and any value that fails to be encoded — resilience in rare states, trust, restraint in the tail — exerts zero force on the optimizer and will be traded away for measured gains without any alarm. The dangerous KPIs are rarely the wrong ones; they are the incomplete ones. Only the human seam, auditing against unencoded reality, catches what the compression dropped.

6. Conclusion

The intelligence age does not change what an organization is — a social construct for achieving goals. It changes what pursues them. Once compute carries the pursuit, the organization inherits computing's two permanent limits: Goodhart on the definition of success, Amdahl on the speed of pursuit. These limits are different in kind, and the central design decision of NimsForest is to answer each with the only kind of response that works. The mechanical limit gets a mechanical answer — an event-driven, wait-free, state-separated architecture whose own objective function drives its serial residue toward the mathematical floor, continuously and automatically. The definitional limit gets the one non-mechanical component in the system — a human-machine seam that owns what winning means and audits that meaning against reality.

The machine pursues, bounded by Amdahl. The human defines and audits, guarding against Goodhart. Everything mechanical in the system self-minimizes; the one thing that cannot be automated is whether the human at the seam keeps doing the irreducible judging that nothing in the machine can do for them. NimsForest is a proposal to run an organization at both frontiers at once — not past them, which is impossible, but against them, with nothing wasted in between.

nimsforest.com