

NEXUS

A Deterministic Blockchain for Verifiable Agent Reasoning and Machine-Energy Settlement

Neiver Alvarez

Draft 0.1 — 24 August 2026

Research whitepaper. The current implementation is a development network, not a production mainnet or investment product.

Abstract

NEXUS is a proposed sovereign blockchain architecture for autonomous agents and machines. It separates deterministic monetary consensus from non-deterministic artificial intelligence. The base state machine validates signatures, nonces, balances, protocol supply, block roots and finality using deterministic rules. Cognitive nodes perform reasoning outside consensus and return signed commitments, evidence references and typed results. Only deterministic Reasoning Receipts are admitted to canonical state. This separation is intended to let agents reason about economic actions without granting a probabilistic model authority to redefine ledger validity.

The protocol further introduces agent accounts with bounded budgets and capabilities, machine identities with hardware-attestation hooks, and an energy-settlement layer in which signed meter attestations can release NEX payments and create accounting credits for verified energy delivery. A separate energy unit, VEC, avoids coupling a scarce monetary asset directly to a volatile physical commodity. The native monetary cap is 8,388,608 NEX (2^{23}), allocated entirely at Genesis into accounts and protocol vaults; protocol rewards are releases from those vaults rather than post-Genesis native issuance.

This paper specifies the architecture, trust boundaries, preliminary Proof-of-Reasoning mechanism, economic invariants, adversarial model and results from a four-process Devnet Alpha. It does not claim that arbitrary language-model reasoning can presently be proven correct. Instead, it defines a path from attestable computation and deterministic task verification toward stronger verifiable cognition, including dispute systems, specialized verifiers and future zero-knowledge or trusted-execution proofs.

1. Introduction

Bitcoin demonstrated that a peer-to-peer network can maintain a scarce digital asset without a central mint by making agreement on transaction history a cryptographic and economic problem [1]. Modern Byzantine-fault-tolerant protocols later showed that replicated state machines can provide deterministic finality under explicit fault assumptions [2]. A different class of systems is now emerging: software agents and physical machines that can discover services, hold keys, negotiate prices and initiate payments without a human signing every individual action.

The missing primitive is not merely a wallet for an AI model. Autonomous systems require constrained authority, verifiable identities, machine-readable policies, resource markets and a way to record what an agent claimed to have computed without making probabilistic model output itself a consensus rule. NEXUS therefore treats “reasoning” as a protocol input that must cross a deterministic verification boundary before it can affect canonical state.

The design goal is a ledger on which humans, agents and machines can exchange scarce digital value and metered physical resources while preserving a small set of constitutional invariants. Energy is a motivating use case because robots, vehicles, chargers, batteries and compute facilities can both consume and provide measurable resources. The same architecture can generalize to compute, data, API access and machine services.

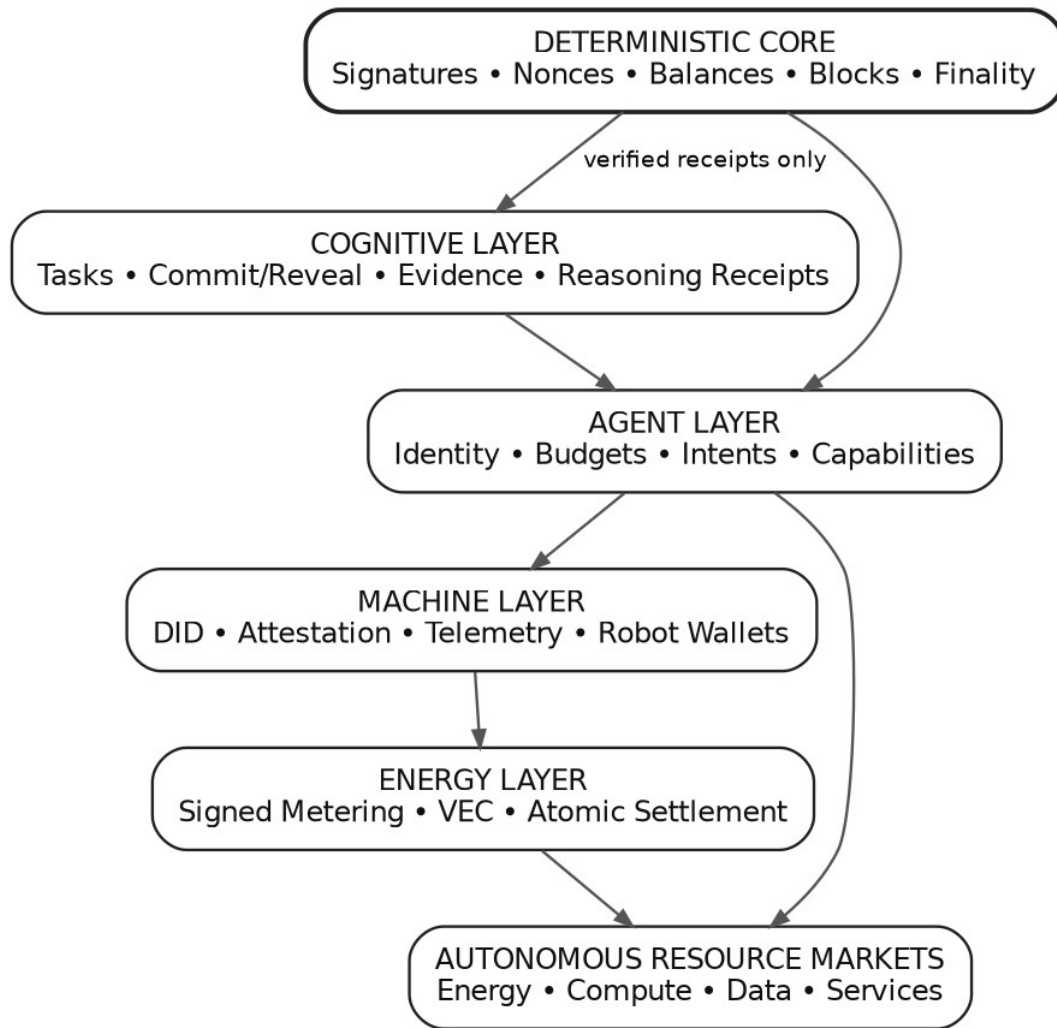


Figure 1. NEXUS separates deterministic ledger consensus from cognition and external machine/energy systems.

2. Design Goals and Non-Goals

Deterministic monetary safety. Given the same valid input and pre-state, honest validators must compute the same post-state. No model inference, stochastic sampling or private prompt state is allowed inside the validity function for balances, signatures, nonces, supply or finality.

Bounded autonomous authority. An agent account must be able to act without repeated human signatures while remaining constrained by budget, capability, counterparty and policy rules.

Verifiable cognitive work. Reasoning nodes should be economically rewarded only for outputs that satisfy an explicit verifier or quorum rule, and the chain should retain compact authenticated receipts rather than private chain-of-thought.

Machine-native settlement. Physical devices should be representable by cryptographic identities, with signed telemetry and replay protection sufficient for deterministic settlement when an accepted oracle policy is satisfied.

Scarcity without discretionary minting. The native asset has a fixed maximum state at Genesis. Reward schedules release previously allocated balances; they do not create native units after Genesis.

Interoperable execution. The production architecture should expose SDKs to general software agents and ROS 2-class robotics systems without requiring the robot to execute the full validator stack.

NEXUS does not claim to solve general artificial intelligence, prove arbitrary natural-language reasoning, eliminate oracle trust for physical events, or make an LLM a judge of blockchain validity. These are explicit non-goals of the base protocol.

3. System Model

The network distinguishes five roles. A participant may operate more than one role, but security analysis treats them separately:

Role	Responsibility
Validator	Executes the deterministic state transition function, verifies blocks and participates in finality.
Cognitive Node	Performs a specified off-chain task and signs commit/reveal messages that may contribute to a Reasoning Receipt.
Agent Account	A programmable economic principal with identity, keys, budget and capability constraints.
Machine Node	A physical or embedded device that can authenticate, sign telemetry and authorize bounded economic actions.
Oracle / Meter	A device or service whose signed statements are accepted by a defined market policy, such as a meter reporting delivered energy.

3.1 Fault assumptions

For the production consensus layer, the intended security target is a partially synchronous, stake-weighted BFT model with safety while Byzantine voting power remains below one third and liveness after the network satisfies the timing assumptions of the chosen BFT implementation. Devnet Alpha currently models a strict greater-than-two-thirds finality threshold; it is not a complete production BFT implementation.

Cognitive committees use a separate adversarial assumption. Their output cannot directly make an otherwise invalid transaction valid. A dishonest committee can at worst corrupt tasks whose verifier policy accepts the committee result; therefore high-value task classes require stronger verifiers, larger committees, cryptographic proofs, trusted hardware attestations, or explicit human/governance review.

4. Monetary Invariants

The native monetary unit is NEX. Its smallest accounting unit is the quanta.

$$\text{MAX_SUPPLY_NEX} = 2^{23} = 8,388,608 \text{ NEX}$$

$$1 \text{ NEX} = 10^9 \text{ quanta}$$

All 8,388,608 NEX are present in the Genesis state. Some balances belong to protocol vaults whose spend conditions are enforced by runtime logic. There is no unrestricted post-Genesis native mint operation. Thus, for every reachable valid state S :

$$\sum_{a \in \text{Accounts}(S)} \text{balance}(a) = 8,388,608 * 10^9 \text{ quanta}$$

Burning, if later introduced, may reduce circulating supply but cannot permit re-minting above the constitutional cap without a new protocol whose identity is explicitly distinct. A software upgrade that silently changes MAX_SUPPLY is invalid under the protocol constitution.

4.1 Genesis allocation

Genesis vault	Draft allocation
Security / validator vault	30%
Cognitive / reasoning vault	25%
Energy-network bootstrap vault	15%
Protocol treasury	10%
Developer / ecosystem vault	10%
Long-term reserve vault	10%

These percentages are research parameters rather than a finalized launch schedule. Mainnet release curves require economic simulations of validator security, agent demand, energy-market adoption, collusion and treasury governance.

5. Deterministic Ledger

NEXUS uses an account-based state machine in the reference implementation. Transactions bind the public key, derived sender address, nonce, destination, amount and optional typed payload. A validator rejects a transaction when the signature is invalid, the public key does not derive the claimed sender, the nonce is not the account's next nonce, the balance is insufficient, or a typed payload fails its deterministic verifier.

The reference cryptographic identity uses Ed25519 signatures, a widely specified EdDSA instance [3]. Production deployment may add hardware-backed or post-quantum migration paths through versioned key types; such additions must not make signature verification probabilistic.



Figure 2. A block commits to canonical state, including cognitive, machine and energy roots.

5.1 Canonical roots

A production block header is intended to commit to at least the transaction root and global state root. The global state may additionally expose domain roots for efficient proof generation: agent_root, reasoning_root, machine_root and energy_root. Domain roots are commitments; the consensus rule remains the canonical deterministic state transition.

6. Finality and Validator Consensus

The Devnet Alpha rotates proposers and collects signed votes weighted by configured stake. A block is marked finalized only when signed voting weight exceeds two thirds of total voting weight:

$$\text{Finalized}(B) \text{ iff } \sum(\text{weight}(v) \text{ for } v \text{ signing } B) > (2/3) * \text{total_weight}$$

The production track is designed to migrate this model onto an audited Rust BFT runtime rather than treating the Python Devnet as a mainnet consensus engine. HotStuff and related partially synchronous BFT protocols provide a well-studied reference point for deterministic blockchain finality [2]. The exact production consensus protocol, networking behavior, evidence rules and validator-set transitions must be fixed in a separate consensus specification before a public testnet can make security claims.

7. Agent Accounts

A normal wallet answers “who can sign?” An autonomous account must additionally answer “what may this signer cause the account to do?” NEXUS therefore models an Agent Account as a principal with keys plus explicit policy state.

```
AgentAccount = (id, owner, keys, model_ref, capabilities, budget, counterparties,
policy_root, reputation)
```

Capabilities may include `pay_energy`, `buy_compute`, `buy_data`, `hire_agent`, `publish_offer` or `invoke_contract`. Budgets may be bounded per transaction, per epoch or per resource class. Counterparty rules can require minimum reputation, machine certification or an allow-list. The deterministic runtime evaluates these fields; the agent’s model does not override them.

7.1 Intents

Instead of requiring a human to construct every low-level transaction, an agent can receive an intent such as “acquire at least 5 kWh within two kilometers, before 18:00, for no more than 0.30 NEX per kWh, from a station above reputation threshold R.” The agent may reason over quotes and routes off-chain. It then submits a concrete transaction or escrow instruction that the chain can validate deterministically against its policy.

8. Proof-of-Reasoning

Proof-of-Reasoning (PoR) is the name of the NEXUS cognitive receipt protocol. It is intentionally not a claim that a blockchain can inspect or prove the truth of arbitrary private reasoning. PoR proves that a defined committee produced signed commitments, revealed outputs under specified rules, and satisfied a task-specific acceptance function.

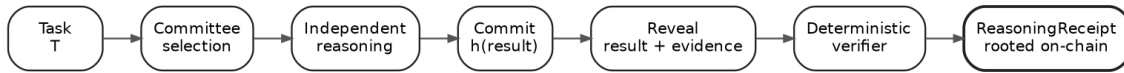


Figure 3. Proof-of-Reasoning commit/reveal pipeline. The final verifier is deterministic.

8.1 Task object

A reasoning task is typed. A canonical task description can be represented as:

```
T = (task_id, task_type, input_hash, verifier_id, committee_policy, deadline,
reward, evidence_policy)
```

The input may remain off-chain if its hash and availability policy are sufficient. `verifier_id` identifies the deterministic function or adjudication policy that interprets reveals. This is critical: free-form text equality is not a safe general consensus rule.

8.2 Commit and reveal

Each selected cognitive node i computes a typed result r_i , local evidence descriptor e_i and nonce n_i , then publishes a signed commitment:

```
c_i = H(domain || task_id || node_id || normalize(r_i) || H(e_i) || n_i)
```

After the commit phase closes, the node reveals r_i , e_i and n_i . The runtime checks identity binding, signature validity, commitment opening, task deadline and replay protection. Nodes that do not reveal can lose reward eligibility; provably inconsistent reveals can trigger deterministic penalties when the task policy specifies them.

8.3 Receipt

If the verifier accepts an output class R , the chain records a compact receipt such as:

```
ReasoningReceipt = H(task_id || verifier_id || result_hash || evidence_root ||
committee_root || vote_weight || confidence || epoch)
```

The receipt is evidence that the protocol process occurred, not evidence that a language model’s hidden thought process was correct. Private chain-of-thought is neither requested nor placed on-chain.

8.4 Verifier classes

Class A — Deterministic computation. The result is directly recomputable on-chain or by validators, e.g. a bounded optimization, signature verification, route scoring from committed data, or exact arithmetic.

Class B — Proof-bearing computation. The output includes a cryptographic proof, SNARK/STARK proof, authenticated execution trace or other efficiently verifiable certificate.

Class C — Attested execution. The output is signed by an approved TEE, hardware meter, sensor module or externally governed oracle set.

Class D — Committee judgment. A committee reaches a threshold on a subjective or model-derived result. This class must have capped economic authority and explicit dispute rules because quorum is not truth.

8.5 Confidence and reputation

For a weighted committee with valid revealed weight W and agreeing weight W_R , a basic observable confidence measure is:

$$\text{confidence}(R) = W_R / W$$

Confidence is descriptive, not a universal probability of correctness. Reputation can affect committee selection or reward only under rules designed to prevent irreversible cartel formation. New nodes require a path to participation; high-value tasks should prefer slashable stake, proof-bearing outputs or independent verifier diversity over pure historical reputation.

9. Cognitive Safety Boundary

The principal protocol rule is that cognition can recommend actions but cannot redefine ledger validity. A model may propose a transfer, classify risk, choose a charging station or evaluate a market. It cannot decide that an invalid signature is valid, skip a nonce, exceed an account budget, increase `MAX_SUPPLY`, retroactively finalize a conflicting block, or slash stake solely because an opaque model predicted malicious intent.

This boundary converts probabilistic intelligence into a source of proposals and attestations while retaining deterministic cryptographic enforcement. It is the main architectural defense against model hallucination becoming monetary consensus failure.

10. Machine Identity and Attestation

Machines require keys that can survive unattended operation and support rotation, ownership transfer and constrained authorization. NEXUS machine records therefore bind a machine identifier to one or more verification keys, capabilities, owner/controller policy, firmware or hardware commitments and optional certification claims. Decentralized Identifiers provide an interoperable conceptual model for such identifiers [4].

A production robot should not expose a long-lived private key to an unrestricted language-model process. Signing should be mediated by a policy engine and, where hardware permits, a secure element, TPM, HSM or trusted execution boundary. High-value commands can require multi-party authorization or short-lived session keys.

11. Energy Protocol and VEC

Energy introduces a physical oracle problem: the chain cannot directly observe a kilowatt-hour. It can only verify signed statements under an oracle policy. NEXUS therefore separates monetary NEX from a verified energy accounting unit, VEC.

Reference convention: 1 VEC = 1 verified kWh under an accepted meter policy

The reference implementation stores micro-kWh integers to avoid floating-point consensus. A signed energy attestation binds meter identity, buyer/machine identity, seller/station identity, quantity, price terms, time window and a unique nonce or attestation identifier. Replay protection prevents the same signed delivery statement from being settled twice.

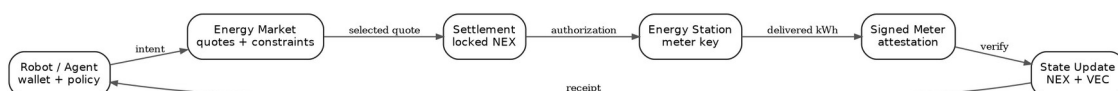


Figure 4. Example robot-to-energy settlement using a signed meter attestation.

11.1 Why NEX is not pegged to kWh

A scarce network asset and a physical energy commodity have different supply dynamics. Defining 1 NEX = 1 kWh would create monetary instability as either energy prices or NEX scarcity changed. VEC therefore represents attested energy quantity, while NEX remains the scarce settlement/security asset. Markets can quote VEC in NEX or stable external units without changing the native cap.

12. Autonomous Resource Markets

The same intent and receipt architecture extends beyond energy. A machine can buy GPU time, inference, mapping data, storage, bandwidth or another robot's service. Existing work on machine-to-machine and agent-mediated payment protocols demonstrates increasing demand for programmatic payments [8]. NEXUS differs by making bounded agent authority, machine identity, cognitive receipts and resource accounting part of one sovereign protocol design rather than only an application-level payment flow.

A generic market order can be represented as constraints rather than a single destination:

```
Intent = (resource_type, minimum_quantity, max_total_cost, deadline,
         geo_constraint, quality_floor, counterparty_policy)
```

Off-chain agents discover and rank offers. The chain validates the final authorization against deterministic account policy and the selected market contract.

13. Fees, Rewards and Security Budget

The native asset has three protocol-level economic roles: transaction fees, validator security and cognitive/resource-market settlement. Validator and reasoning rewards are released from Genesis vaults according to explicit schedules. When vault emissions decline, long-term security must be supportable by transaction and service fees rather than hidden inflation.

Reasoning rewards should be task-funded wherever possible. A party requesting cognitive work escrows a task reward. The protocol may subsidize early network tasks from the cognitive vault, but a sustainable network should price cognitive work according to demand, compute cost, verifier strength and risk.

14. Security Analysis

NEXUS creates additional attack surfaces beyond a conventional ledger. The protocol therefore treats reasoning, machine telemetry and autonomous execution as adversarial boundaries rather than trusted features.

Threat	Failure mode	Primary mitigation
Key impersonation	Signer uses own key while claiming another sender.	Address derivation is bound to the public key; sender/key mismatch is invalid.
Replay	Old transaction or meter statement is resubmitted.	Account nonces plus unique attestation identifiers and replay sets.
Byzantine validators	Validators sign conflicting or invalid blocks.	BFT threshold, evidence and slashing in production specification.
Reasoning collusion	Committee coordinates on an incorrect subjective result.	Task-class limits, random selection, stake, verifier diversity, proofs/attestations for high-value tasks, disputes.
Model nondeterminism	Same prompt produces different outputs.	Model inference remains off-chain; only normalized typed results and deterministic verifier rules enter state.

Threat	Failure mode	Primary mitigation
Oracle fraud	Meter lies about energy delivery.	Hardware keys, certification, multi-meter reconciliation, challenge windows and capped exposure.
Agent prompt injection	External data convinces agent to violate owner intent.	Runtime-enforced budgets/capabilities; model cannot bypass deterministic policy.
Sybil attack	One operator creates many cognitive identities.	Stake/resource admission, VRF-like committee sampling, identity diversity and anti-concentration rules.
Governance capture	Upgrade changes monetary/security rules.	Constitutional invariants, delayed upgrades, public code hash, multi-stage governance and exit window.

14.1 Slashing constraints

Slashing must be based on objective protocol evidence: double-signing, invalid commitment opening, signed contradiction under a deterministic task policy, or other reproducible faults. An opaque AI classification is not sufficient evidence to destroy stake. Subjective misconduct is handled through bounded reputation, market exclusion or governance processes rather than automatic irreversible punishment.

15. Privacy and Data Minimization

Autonomous systems can generate sensitive telemetry. NEXUS therefore prefers commitments and receipts over raw sensor histories or model traces. A task can commit to `input_hash` and `evidence_root` while encrypted or access-controlled data remains off-chain. The ledger should not require publication of private chain-of-thought, raw camera feeds, precise robot trajectories or proprietary model prompts merely to prove that a task occurred.

Future privacy work includes zero-knowledge proofs for policy compliance, selective-disclosure credentials for machine certification and threshold encryption for committee tasks. These are roadmap items, not properties of Devnet Alpha.

16. Governance and Upgrades

A protocol that contains agents must not permit agents to silently rewrite the rules governing their authority. Runtime upgrades therefore require deterministic code hashes, publicly inspectable proposals, activation delays and explicit migration logic. Constitutional changes such as the native supply cap should require an exceptional governance path and produce a visibly distinct protocol version rather than being disguised as routine maintenance.

Emergency security controls, if present at launch, must be time-bounded and transparent. Long-term decentralization requires removing unilateral administrator powers as the implementation and validator set mature.

17. Devnet Alpha

A reference implementation has been built to test protocol invariants before a production Rust runtime. The current Devnet Alpha launches four independent validator processes with a deterministic shared Genesis, TCP peer propagation, proposer rotation, signed votes, state-root verification, transaction gossip, block persistence/replay, basic synchronization, JSON HTTP RPC and a browser explorer.

The integration scenario funds a robot account, finalizes a transfer, executes a four-node reasoning task with one deliberate dissenter, settles signed energy delivery, and verifies that all validator processes converge on the same chain tip and state root.

Verified Alpha result	Value
Independent processes	4
Finalized height	3
All nodes same state	true
All nodes same tip	true
Reasoning agreeing nodes	3 of 4
Reasoning confidence	75%
Energy delivered	5.4 kWh
NEX paid	1.08 NEX
Robot ending balance	98.92 NEX
Station ending balance	1.08 NEX
Station VEC balance	5,400,000 micro-kWh

The verified Genesis hash for this run is ec24ae1352d576d3b9eb34ab774c954ee707268b407f75955d72b1521ef38c5e. The result demonstrates state convergence of the reference implementation; it is not a proof of production security.

18. Related Work

Bitcoin established peer-to-peer digital scarcity and a chain of computationally expensive timestamps [1]. NEXUS does not attempt to replace Bitcoin's proof-of-work security model with probabilistic AI; it adopts the opposite lesson that monetary validity must remain mechanically verifiable.

HotStuff and other BFT protocols formalize deterministic finality under Byzantine faults and partial synchrony [2]. The production NEXUS consensus layer is intended to be based on audited BFT engineering rather than the research Devnet's simplified voting model.

Bittensor is relevant prior art for economically coordinating machine-learning participants and scoring useful model output. Its Yuma Consensus is an incentive mechanism associated with subnet intelligence rather than a claim that arbitrary LLM reasoning defines base-chain validity [5]. NEXUS instead defines a generic receipt boundary between off-chain cognitive work and canonical deterministic state.

Decentralized Identifiers standardize identifiers controlled through cryptographic verification methods and are relevant to portable human, agent and machine identity [4]. Machine-economy networks such as peaq and emerging x402 payment flows demonstrate machine identities and autonomous programmatic payment use cases [6][8]. NEXUS treats those trends as evidence for a machine economy while focusing its protocol novelty on bounded autonomous authority, typed reasoning receipts, and an energy-aware sovereign state model.

19. Implementation Roadmap

Stage	Scope	Status
V0	Constitution and hard invariants	Complete in reference repository
V1	Signed transactions, blocks, roots	Complete in reference repository
V2	Stake-weighted finality model	Complete in Devnet model

Stage	Scope	Status
V3	Agent accounts and policies	Reference implementation
V4	PoR commit/reveal	Reference implementation
V5	Cognitive roots and firewall	Reference implementation
V6	Machine identity	Reference implementation
V7	Signed energy protocol and VEC	Reference implementation
V8	Autonomous intents	Reference implementation
V9	Adversarial simulation / slashing hooks	Reference implementation
V10	Genesis composition	Reference implementation
Alpha -> Testnet	Rust sovereign node, audited BFT, persistent networking, formal state spec	Next engineering phase
Public Testnet	Open validators, SDKs, ROS integration, external meters, explorer	After audits and fuzz/property testing
Mainnet	Decentralized validator set, finalized economics/governance, independent audits	Not scheduled until testnet evidence is sufficient

20. Open Research Questions

- How should task-specific verifiers compose without turning the base runtime into an unbounded execution environment?
- Which cognitive tasks merit objective slashing, and which should only affect reputation or market payments?
- How can committee selection remain permissionless while resisting Sybil attacks and long-lived reputation cartels?
- Which classes of agent inference can be efficiently converted into proof-bearing computation?
- How should physical meter trust be decentralized across hardware roots, independent meters and challenge markets?
- What fee market balances validators, reasoning nodes and resource providers when autonomous microtransactions dominate traffic?
- How can machine privacy be preserved while still proving policy compliance and delivered resources?
- What upgrade process can evolve cryptography and agent capabilities without weakening the monetary constitution?

21. Limitations

The current PoR implementation demonstrates signed commit/reveal quorum, not proof of semantic correctness. Current energy settlement trusts configured meter keys. Devnet networking is suitable for research integration tests, not hostile Internet deployment. The validator model has not undergone formal verification or independent security audit. Economic allocations and rewards are draft parameters. Hardware key custody, cross-jurisdiction energy regulation, sanctions/compliance obligations, robotics safety and consumer-protection questions are outside the current protocol implementation and must be addressed before commercial deployment.

These limitations are intentionally explicit. The project should be judged by whether later specifications and testnets eliminate or constrain them, not by branding.

22. Conclusion

NEXUS proposes that autonomous economic systems require two different kinds of agreement. The first is exact: who owns what, which signature is valid, which block is final, and whether the monetary invariants hold. The second is cognitive: which plan, classification or resource choice an agent recommends. Conflating them is unsafe. NEXUS therefore keeps monetary consensus deterministic while making cognitive work an authenticated, typed and economically accountable input.

If the architecture succeeds, a robot could hold a constrained account, discover energy or compute, reason about competing offers, authorize a bounded purchase, receive a signed resource attestation and settle value without a human approving every micro-action. The ledger would not “think” in the human sense. It would provide a deterministic constitution within which many competing agents can reason, prove what protocol process they followed, and exchange resources.

Appendix A. Canonical Objects (Draft)

Transaction

version, sender, public_key, nonce, destination, amount_quanta, fee_quanta, payload_type, payload_hash, signature

ReasoningTask

task_id, task_type, input_hash, verifier_id, committee_policy, deadline, reward_quanta, evidence_policy

ReasoningCommit

task_id, node_id, commitment, public_key, signature

ReasoningReveal

task_id, node_id, normalized_result, evidence_hash, nonce, signature

ReasoningReceipt

task_id, verifier_id, result_hash, evidence_root, committee_root, agreeing_weight, revealed_weight, confidence_bps, receipt_hash

MachineRecord

machine_id, owner, verification_keys, capabilities, firmware_commitment, attestation_policy, status

EnergyAttestation

attestation_id, meter_id, buyer_id, seller_id, micro_kwh, price_quanta, timestamp_window, nonce, signature

Appendix B. Constitutional Rules

1. MAX_SUPPLY = 8,388,608 NEX and precision is integer quanta.
2. No cognition module can mutate supply, signature validity, nonce validity or finality rules.
3. Consensus is deterministic for identical state and input.
4. Private model chain-of-thought is neither requested nor stored as consensus data.
5. Cognition is represented by commitments, signed evidence references, typed verdicts and receipts.
6. Machine and energy claims require cryptographic attestation plus replay protection.
7. Autonomous agents operate inside explicit budgets, capabilities and counterparty policies.
8. Slashing is rule-based and reviewable; an opaque model prediction alone cannot slash stake.
9. Protocol upgrades cannot silently change the monetary cap.
10. Mainnet requires independent security/economic audits and public testnets.

References

- [1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. <https://bitcoin.org/bitcoin.pdf>
- [2] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, I. Abraham, “HotStuff: BFT Consensus in the Lens of Blockchain,” arXiv:1803.05069, 2018. <https://arxiv.org/abs/1803.05069>
- [3] S. Josefsson, I. Liusvaara, “Edwards-Curve Digital Signature Algorithm (EdDSA),” RFC 8032, 2017. <https://www.rfc-editor.org/rfc/rfc8032>
- [4] W3C, “Decentralized Identifiers (DIDs) v1.0,” W3C Recommendation, 19 July 2022. <https://www.w3.org/TR/did/>

[5] Bittensor Documentation, “Yuma Consensus / Consensus,” accessed 2026. <https://docs.learnbittensor.org/>

[6] peaq Documentation, machine identity and machine-to-machine payment materials, accessed 2026. <https://docs.peaq.xyz/>

[7] NEXUS Cognitive Chain reference repository, Devnet Alpha result, 24 August 2026. Genesis: `ec24ae1352d576d3b9eb34ab774c954ee707268b407f75955d72b1521ef38c5e`.

[8] x402 Documentation, “Welcome to x402,” accessed 2026. <https://docs.x402.org/introduction>

Publication Note

This document is Draft 0.1. Technical criticism, attack scenarios, counterexamples and independent implementations are explicitly invited. Before any production mainnet, the consensus protocol, cryptographic constructions and economic mechanism must undergo independent review.