NEXGPU CHAIN

TECHNICAL WHITEPAPER V1.0 — COMPACT EDITION

Decentralized GPU Infrastructure for the Autonomous Machine Economy

WORLD COMPUTE GRID / NEURALMESH / COMPUTE PASSPORT
HYPERCLUSTERS / AGENTOS / NGX

Executive Summary

NEXGPU Chain is an open coordination layer for artificial intelligence compute. Decentralized providers execute AI workloads; an EVM-compatible layer records machine identity, task commitments, proofs, staking and settlement; and an Agent-Native layer gives autonomous software controlled access to wallets, budgets and machine-to-machine services.

Contents

01. Thesis and Design Principles

NEXGPU coordinates AI compute instead of placing heavy matrix operations inside consensus. GPU execution remains off-chain; machine identity, task commitments, proofs, staking and settlement are coordinated by the protocol. The design prioritizes modularity, heterogeneous hardware, workload-specific verification, privacy minimization and economic accountability.

02. System Architecture

The stack has three layers: GPU Compute Network, AI Execution Chain and Agent-Native Protocol. The first executes workloads, the second coordinates state and settlement, and the third exposes controlled economic capabilities to autonomous agents. Large model weights, datasets, checkpoints and verbose logs remain off-chain; hashes, proof summaries, state transitions and economic commitments belong in consensus.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

03. World Compute Grid

World Compute Grid turns heterogeneous machines into discoverable services. Providers advertise GPU class, VRAM, benchmark roots, network characteristics, regional policy, TEE capability, availability and price constraints. Service classes range from Pioneer Nodes to Professional Nodes and Enterprise Clusters with stronger SLA and attestation requirements.

04. NeuralMesh Scheduling Protocol

NeuralMesh ranks candidates using performance, reliability, latency, price, reputation and policy compatibility rather than matching only on free capacity. Conceptual score: Score_i = w1*Perf_i + w2*Reliability_i + w3*Latency_i + w4*Price_i + w5*Reputation_i + w6*PolicyMatch_i.

Scheduling Formula

Score_i = w1*Perf_i + w2*Reliability_i + w3*Latency_i + w4*Price_i + w5*Reputation_i + w6*PolicyMatch_i

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

05. Compute Passport

Compute Passport links a provider address to hardware attestations, benchmark history, reputation, stake and disputes. Suggested state includes NodeID, HardwareFingerprint, GPUClass, VRAM, RegionCode, TEECapability, BenchmarkRoot, ReputationScore, StakeAmount and LastAttestation.

06. Proof of Compute and Intelligence

Proof of Compute asks whether a task was executed. Proof of Intelligence is a broader contribution concept for useful, reliable AI work. Verification can combine recomputation, probabilistic sampling, redundant execution, TEE attestation and challenge mechanisms. ECU = NormalizedPerformance * ActiveTime * TaskFactor * ReliabilityFactor * VerificationFactor.

Effective Compute Unit

ECU = NormalizedPerformance * ActiveTime * TaskFactor * ReliabilityFactor * VerificationFactor

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

07. HyperClusters

HyperClusters are temporary virtual superclusters assembled from multiple providers for suitable distributed workloads. Formation must model bandwidth, synchronization, checkpoint cost and data locality. A Cluster Manifest records members, roles, shards and failure substitution policy.

08. AI Execution Chain

The coordination layer may include NodeRegistry, PassportRegistry, TaskMarket, StakeManager, Settlement, Dispute, Governance, Treasury and AgentRegistry. EVM compatibility is a design target. 3,000+ TPS and low fees are targets requiring independent benchmarking before being described as achieved.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

09. AgentOS

AgentOS gives autonomous agents controlled identity, wallets, service discovery and compute budgets. Agent wallets should support spending limits, session keys, allowlists, time locks, multi-policy approval and emergency revocation.

10. Machine-to-Machine Economy

Software agents can discover services, compare bounded prices, purchase compute and settle outcomes without a human approving every transaction. A Payment Intent defines service, budget and policy; the provider executes, submits proof and receives settlement after verification.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

11. Privacy and Trusted Execution

Baseline jobs use encrypted transport, isolated containers and scoped credentials. Higher-security jobs may require TEE, remote attestation, encrypted storage and restricted egress. Zero-knowledge techniques may support selected cases, but the protocol should not claim universal privacy for every AI workload.

12. Runtime, Storage and Networking

The runtime manages image verification, secure pull, GPU binding, isolation, health checks, logs, checkpoints, proof generation and cleanup. Storage may combine object storage and content addressing. Networking provides private task networks, port policies, bandwidth controls and region-aware routing.

Container Lifecycle

Task Accepted → Image Verification → Secure Pull → Resource Reservation → Container Launch → Health Check → Compute → Checkpoint → Proof Generation → Result Upload → Cleanup

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

13. NGX Token Economy

NGX is proposed for compute payment, provider staking, validator security, governance, protocol fees and machine-to-machine payments. Proposed maximum supply: 8,800,000,000 NGX. Final issuance, legal characterization, deployment chain and contract address remain to be determined.

14. Token Allocation and Vesting

Proposed allocation: 38% Compute & Node Incentives; 20% Ecosystem & Developer Grants; 15% Foundation & Protocol Development; 10% Treasury; 10% Core Contributors; 7% Strategic Network Development. Contributor and strategic allocations should use cliffs and long-term linear vesting; node incentives should follow verifiable service contribution.

Allocation	Share
Compute & Node Incentives	38%
Ecosystem & Developer Grants	20%
Foundation & Protocol Development	15%
Treasury	10%
Core Contributors	10%
Strategic Network Development	7%

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

15. ComputeFi and Protocol Revenue

ComputeFi describes compute as a measurable, programmable economic primitive; it does not imply guaranteed yield. Protocol fees may support providers, treasury, security reserves and an optional burn component. Burning does not guarantee token appreciation.

16. Governance and Treasury

A proposed structure combines a Foundation, a 15-seat DAO Council and broader token governance. Major upgrades and large Treasury expenditure should use wider governance, timelocks and emergency controls.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

17. Developer Cloud

Proposed developer interfaces include Compute API, Agent SDK, CLI, GPU Console, Explorer, Model Deployment and task monitoring. NexusGPT, VisionForge, AgentX and LaunchHub are concept modules in this document, not confirmed independent companies or partnerships.

18. Security and Threat Model

Threats include fake hardware, fabricated results, provider dropout, container escape, data theft, Sybil nodes, DoS, malicious jobs, key compromise and governance attacks. Mitigations include staking, reputation, attestation, sandboxing, rate limits, challenges, audits, bug bounties and emergency pause procedures.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

19. Roadmap 2030

Phase 01 GPU Network Foundation; Phase 02 AI Execution Chain; Phase 03 NeuralMesh and HyperClusters; Phase 04 AgentOS; Phase 05 Machine Economy Layer. Dates, node counts and performance figures should only be published as facts after engineering validation.

20. Risks and Disclosures

During implementation, this project will face various practical challenges, including scheduler-module anomalies, network bandwidth constraints, verification computing overhead, smart-contract security vulnerabilities, shifting market demand, imbalanced incentive models, concentration of computing-power resources, token-price volatility, and changes in regulatory policies across jurisdictions.

This whitepaper sets forth the technical vision and product concepts of NEXGPU for reference and discussion among developers and computing-power participants. It does not constitute any professional investment, legal, or tax advice. Future features, partnership progress, and launch arrangements of the project are still in the planning phase. No assurances are made regarding commercial returns or final implementation outcomes.

Implementation Notes

Implementation should proceed through measurable testnet milestones. Performance targets, security properties and economic parameters should be backed by reproducible benchmarks, public code or independent review. Protocol parameters remain modular, while engineering priorities include deterministic interfaces, observable failure states, safe key management, signed artifacts, checkpoint recovery and explicit policy boundaries for autonomous agents.

Appendix A — Protocol Data Flow

Stage	Primary Object	Execution
1	Task Manifest / Budget Lock	On-chain + off-chain
2	NeuralMesh Selection	Off-chain
3	Passport Verification	Registry + attestation
4	GPU / HyperCluster Execution	Off-chain GPU
5	Proof & Challenge	Proof summary
6	Settlement & Reputation	On-chain