AIDP

AIDP.store — Decentralized AI Compute Network

Whitepaper

Whitepaper v1.0 | 2025

Abstract

AIDP.store is a decentralized AI compute network built on DePIN (Decentralized Physical Infrastructure Network) architecture. It unifies globally distributed GPUs, storage, bandwidth, and orchestration nodes into a verifiable compute market with cryptographic proofs, tokenized incentives, and programmable financial coordination.

This whitepaper presents:

- Full system architecture
- Node roles & staking models
- Proof-of-Resource (PoR)
- Decentralized Inference & Orchestration Layer (Phase 2)
- Cryptographic verification
- Tokenomics and economic models
- Treasury (DAT) mechanics
- Governance and security assumptions

AIDP aims to become a global decentralized inference layer capable of supporting LLMs, multimodal AI, and general-purpose compute workloads.

1. Executive Summary

Artificial Intelligence Demand

Artificial intelligence is driving unprecedented demand for high-performance compute infrastructure. Traditional cloud systems—centralized, expensive, and geographically constrained—cannot meet global AI inference needs sustainably.

AIDP provides:

- A decentralized GPU marketplace
- Verifiable job execution through proof systems
- Token-staked orchestration for decentralized scheduling

- Cryptographic guarantees of correctness and delivery
- Sustainable economy through buybacks, burns, staking, and the DAT treasury

The protocol transforms global idle GPU resources into a unified compute fabric with economic incentives that align reliability, performance, and supply growth. By integrating programmable settlement and decentralized inference routing, AIDP becomes a scalable, permissionless network for AI workloads.

2. Introduction

AI-driven workloads such as LLM inference depend on large fleets of GPUs. These tasks require:

- High throughput
- Low-latency token streaming
- Batch scheduling
- Distributed compute execution

However, centralized cloud platforms exhibit:

- High compute rental prices
- Expensive egress and storage
- Single-point control
- Regional limitations
- Lack of transparency
- No cryptographic verifiability

Decentralized compute networks (DePIN) offer an alternative by incentivizing hardware owners to contribute compute to a global pool.

AIDP extends this model by embedding:

- Verifiable compute proofs
- Decentralized orchestration (Phase 2)
- P2P resource discovery
- Tokenized incentives
- Treasury-driven buybacks and burns
- Multi-chain settlement

3. Market Background

3.1 AI Compute Megatrend

AI models such as LLMs require:

- VRAM-intensive GPUs
- Parallel inference
- Micro-batching
- Multi-request concurrency

Demand is rising faster than centralized data centers can be built.

3.2 Global Idle Hardware

Millions of GPUs (consumer + prosumer + enterprise) sit underutilized. DePIN networks unlock these resources.

3.3 Problems in Current Cloud Market

Problem	Impact
High GPU rental prices	Limits AI deployment
Centralized control	Vendor lock-in
No cryptographic validation	Cannot trust compute correctness
Regional monopolies	High latency for global users
Slow provisioning	Difficult to scale workloads

AIDP provides a solution through:

- Decentralized marketplaces
- Verifiable outputs
- Tokenized incentives
- Distributed routing

4. System Overview

AIDP transforms distributed hardware into a unified compute graph. Jobs flow through:

```
Client → Orchestration Node → Compute Provider → Verification → Settlement → Rewards
```

Core system components:

- Compute Nodes
- Storage Nodes
- Bandwidth Nodes
- Orchestration Nodes (Phase 2)
- Settlement Layer (Solana + EVM)
- Verification Layer
- DAT Treasury

5. Architecture

AIDP uses a 5-layer modular design.

```
+-------------------------------------------------------+
|    APPLICATION LAYER                                  |
|    (SDKs, APIs, Model Endpoints, User Interfaces)     |
+-------------------------------------------------------+
|    SETTLEMENT LAYER                                   |
|    (Solana Treasury, EVM Staking, Rewards Distribution)  |
+-------------------------------------------------------+
|    VERIFICATION LAYER                                 |
```

```
|    (PoR, PoD, PoU, PoB, VCM, zk-Proofs, Merkle Audits)         |
+---------------------------------------------------------------+
|                  P2P ROUTING / NETWORK                        |
|    (Discovery, Pricing, Latency Optimization, Gossip)         |
+---------------------------------------------------------------+
|    PHYSICAL RESOURCE LAYER                                    |
|    (GPUs, Storage, Bandwidth, Sensors)                        |
+---------------------------------------------------------------+
```

AIDP's architecture is designed for scalability, verifiability, and decentralization. It consists of five modular layers, each responsible for distinct system functions.

6.1 Physical Resource Layer

This layer includes all hardware contributors:

- Compute Nodes (GPU Providers)
- Consumer GPUs (RTX series)
- Prosumer / workstation GPUs
- Datacenter GPUs

6.2 P2P Networking Layer

Provides decentralized communication between all participants.

Functions

- Node discovery
- Pricing broadcast
- Latency measurement
- Fault-tolerant request routing
- Bandwidth validation
- Secured gossip protocol

Storage Nodes

- Store datasets, model checkpoints, embeddings
- Must satisfy periodic availability audits (PoS)
- Enable content-addressable storage

Bandwidth Nodes

- Relay traffic between compute providers & orchestrators
- Prove bandwidth capacity through PoB

Sensor Nodes (Future Expansion)

- IoT edge sensors feeding real-time data streams
- Useful for edge AI workloads

6.3 Verification Layer

The verification layer guarantees trustless execution. It supports:

- Proof-of-Resource (PoR)
- Proof-of-Delivery (PoD)
- Proof-of-Uptime (PoU)
- Proof-of-Bandwidth (PoB)
- Verifiable Compute Module (VCM)

- Optional Zero-Knowledge Proofs
- Merkle-root aggregation

This ensures compute jobs are:

- Correctly executed
- Delivered to the user
- Produced by the claimed node
- Measurable and auditable

6.4 Settlement Layer (Solana + EVM)

AIDP combines Solana and EVM environments:

Solana Layer - High-performance treasury

- Real-time fee routing
- Revenue → Buybacks → Burns

EVM Layer

- Node registration
- Staking
- Slashing
- Emissions
- Governance

This hybrid model positions AIDP for both high-speed settlement and broad EVM compatibility.

6.5 Application Layer

Provides APIs and SDKs for:

- Submitting inference jobs
- Accessing compute resources
- Billing and metering
- Building apps using decentralized AI compute
- Dataset and model marketplace interactions

Users interact with AIDP through:

- RPC endpoints
- REST APIs
- Python libraries
- JavaScript SDK
- CLI tools

7. Node Classes

7.1 Compute Nodes

The backbone of the network. Contribute:

- GPU cycles

- VRAM
- Tensor compute
- Memory bandwidth
- Token streaming inference

Verification: PoR, VCM checks

Slashing: Offline, Incorrect compute, Fake submissions

7.2 Storage Nodes

Provide:

- Model checkpoints
- Datasets
- Cache layers
- Embeddings

Verification: PoS (Proof-of-Storage), Timing-based challenges

7.3 Bandwidth Nodes

Ensure network throughput by:

- Relaying inference streams
- Supporting multi-region load balancing

Verification: PoB (Proof-of-Bandwidth)

7.4 Sensor Nodes (Future)

Will enable:

- Edge inference
- Real-time data feeds
- IoT-integrated compute tasks

7.5 Orchestration Nodes (Phase 2)

A major upgrade to decentralize routing. Roles:

- Select optimal compute provider
- Run decentralized scheduling algorithms
- Measure latency + price + reliability
- Dispatch jobs
- Validate proofs
- Initiate settlement

Staking Requirement: $S_{node} \geq S_{min}$

Orchestration nodes earn:

- Routing fees
- Workload commissions

This removes all centralized chokepoints.

8. Phase 2 — Decentralized Inference & Orchestration Layer

(This is one of the most important sections.)

The centralized routing component is replaced by a decentralized marketplace of token-staked orchestration nodes.

Workflow

1. User submits job → enters global job pool
2. Orchestration nodes pick jobs using a fair selection rule
3. Each orchestrator computes the routing function
4. Best provider chosen
5. Job assigned
6. Execution → Verification → Settlement
7. Rewards distributed based on performance

8.1 Decentralized Routing Function (Mathematical)

Orchestration nodes compute a utility function:

```
p* = arg min_p( α L_p + β C_p + γ (1 - R_p) + δ (1/U_p) )
```

Where:

- L_p = latency
- C_p = cost
- R_p = reliability
- U_p = uptime score

Weights (α, β, γ, δ) are governance-controlled.

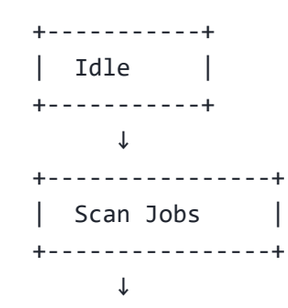
8.2 Verification in Phase 2

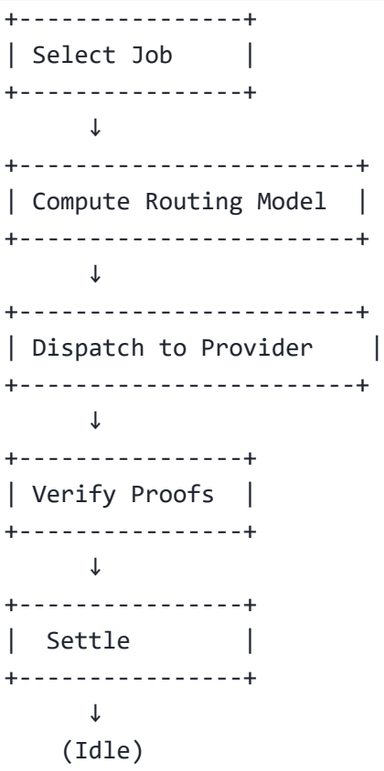
Orchestrators validate:

- PoR (output hash correctness)
- PoD (delivery signature)
- Re-computation checks (sampled tasks)
- Uptime signatures
- Completion proofs

Only after validation does settlement occur.

8.3 Orchestration Node State Machine





8.4 Benefits of Phase 2

- Full decentralization
- Market-based competition
- Lower cost-per-inference
- Better GPU utilization
- More providers onboarded
- Neutral scheduling governance

9. Proof Systems (Core of AIDP Trust Model)

AIDP relies on cryptographic verification to guarantee that compute, storage, and bandwidth contributions are real, correct, and delivered.

The 5 proof classes:

1. PoR — Proof of Resource
2. PoD — Proof of Delivery
3. PoU — Proof of Uptime
4. PoB — Proof of Bandwidth
5. VCM — Verifiable Compute Module

Each ensures trustless validation and prevents cheating.

9.1 Proof-of-Resource (PoR)

Ensures that a compute provider executed the requested job correctly.

```
PoR = H(Input, Model, Output, t)
```

Where: H = cryptographic hash, t = timestamp

PoR certifies that:

- The provider used the correct model
- The output is deterministic/computable
- The computation happened during the valid time window

9.2 Proof-of-Delivery (PoD)

Confirms output delivery to recipient.

```
PoD = H(O ‖ σ_provider)
```

Where: O = output, σ_provider = provider's signature

PoD ensures:

- The correct user received the correct output
- Provider cannot fake job completion

9.3 Proof-of-Uptime (PoU)

Used to measure provider availability.

```
PoU_t = σ_node(H(t))
```

Nodes must respond to random heartbeat challenges. Failure reduces rewards or triggers slashing.

9.4 Proof-of-Bandwidth (PoB)

Measures actual data throughput of bandwidth nodes.

```
B = D / T
```

Where: D = data transferred, T = time

9.5 Verifiable Compute Module (VCM)

Ensures compute correctness via deterministic recomputation.

```
VCM(Job) = Valid ⇔ ∃ O' : f(I) = O'
```

VCM may also use:

- sampling
- partial recompute
- redundant execution

to detect malicious nodes.

10. Protocol Mechanics

10.1 Job Lifecycle

A compute job passes through:

```
User → Job Pool → Orchestration Node → Compute Provider → Verification →
Settlement → Rewards
```

Detailed steps:

1. Submit Job — user signs job with constraints (price, latency, VRAM).
2. Queue — job enters global P2P pool.
3. Orchestrator Selection — token-staked nodes pick jobs fairly.
4. Routing Decision — compute routing model (utility function).
5. Dispatch — provider begins execution.
6. Proof Submission — PoR, PoU, PoD generated.
7. Verification — cryptographic + statistical checks.
8. Settlement — Solana treasury processes payments.
9. Reward Emission — EVM side distributes AIDP tokens.

10.3 Pricing Model

Providers publish:

- price per token
- price per second
- VRAM tier pricing
- data transfer pricing

AIDP supports:

- Static pricing
- Dynamic pricing
- Auction-based markets (future)

11. Tokenomics

11.1 Tokenomics Model

The AIDP token is the native utility and governance asset, designed with a strong focus on utility and long-term deflation.

Token Supply & Allocation

- Total Supply: 1,000,000,000 AIDP
- Circulating: 80%
- Locked/Vested: 20%

Category	Allocation	Vesting Schedule
Team & Contributors	5%	6-month cliff, then linear unlock (monthly)
Marketing & Growth	5%	Unlocked fully at TGE (0-month unlock)
Exchange Listing & Liquidity Fund	3%	Unlocked immediately at TGE

Category	Allocation	Vesting Schedule
Buyback & Burn	5%	Unlock after 3 months linear vesting
Emergency Treasury	2%	Unlock Immediately DAO-controlled

11.3 Token Utility

AIDP is used for:

- Compute payments
- Node staking (compute, storage, bandwidth, orchestration)
- Governance voting
- Priority routing
- LP incentives
- Burn-to-access premium datasets / models

11.4 Emission Reduction (Halving)

```
E_(n+1) = E_n / 2
```

Halving occurs every 24 months, or fixed number of epochs.

11.5 Reward Formula

The reward for node (i) at epoch (t):

```
R_(i,t) = β_r · w_res(i) · u_t(i) · D_n(t)
```

Where:

- β_r = base rate
- w_res = resource weight
- u_t = uptime score
- D_n(t) = network demand scalar

Resource Weights

- Compute: 2.2
- Bandwidth: 1.5
- Storage: 1.0
- Sensor: 0.8

11.6 Buyback & Burn Model

```
Buyback_t = 0.4 · Revenue_t
Burn_t = 0.2 · Fees_t
Slash_i = λ_s · Stake_i (where λ_s = 10–30%)
```

12. Economic Model

The AIDP economic system is designed to:

- Incentivize reliable GPU compute providers
- Reduce token supply over time
- Create sustainable revenue flows
- Stabilize long-term network economics
- Reward orchestrators and contributors based on performance

The model combines:

- Emissions (decreasing)
- Buybacks (increasing with revenue)
- Burns (constant percentage of fees)
- Staking lockups
- Slashing
- Treasury fund growth

12.1 Revenue Model

The protocol generates revenue from:

1. Compute Payments — Users pay for inference / compute jobs.
2. Data / Model Access Fees — Premium datasets and model checkpoints.
3. Storage Fees — Long-term model/dataset persistence.
4. Bandwidth Fees — Streaming, token-by-token inference, distributed jobs.
5. Enterprise API Plans — Bulk access contracts.

12.2 Revenue → Token Flow

```
Revenue → Treasury → Buybacks (40%) → Burns (20%) → Stakers / Nodes (rewards)
```

12.3 Treasury Equation

The DAT (Decentralized Autonomous Treasury):

```
T_(t+1) = T_t + Rev_t - (Rewards_t + Ops_t)
```

Where:

- Rev_t = incoming protocol revenue
- Ops_t = operational & ecosystem expenses

Treasury assets support:

- Market stability
- Buybacks
- Emergency fund
- Partnerships

12.4 Net Deflationary Condition

AIDP becomes deflationary when:

```
Buyback_t + Burn_t > Emissions_t
```

Given halving + growing revenue, this will naturally occur after network maturity.

13. DAT Treasury Model

AIDP integrates a dedicated treasury mechanism inspired by QPin's DePIN DAT architecture.

Core Responsibilities

- Accumulate value from compute demand
- Stabilize token economy
- Enable buybacks
- Burn tokens to reduce supply
- Fund ecosystem grants
- Backstop slashing / risks

13.1 Treasury Funding Sources

1. Compute revenue
2. Dataset & model fees
3. Transaction fee shares
4. Penalty & slashing fees
5. Enterprise contracts
6. Cross-chain fee flows

13.2 Treasury Expenditures

- Buybacks
- Burns
- Node rewards
- Liquidity incentives
- Development grants
- Emergency events
- Infrastructure improvements

13.3 Mathematical Model

```
T_(t+1) = T_t + R_f - B_t - Burn_t - S_t - G_t
```

Where:

- R_f = revenue inflow
- B_t = buybacks
- Burn_t = burns
- S_t = staking rewards
- G_t = grants & expenses

14. Governance Framework

AIDP governance follows a DAO model governed by token holders.

14.1 Governance Powers

Token holders can:

- Modify reward parameters
- Adjust resource weights
- Manage emissions schedule
- Update slashing conditions
- Approve treasury expenditure
- Modify routing algorithms (weights α, β, γ, δ)
- Onboard new node types
- Approve future protocol upgrades

14.2 Governance Lifecycle

```
Proposal → Review (48h) → Voting (24h) → Timelock (36h) → Execution
```

14.3 Staking-Based Influence

Voting power is based on staked tokens, not free-floating supply.

Optional feature:

```
VotingPower = Stake × LockDurationWeight
```

15. Security Model

Security is paramount for decentralized compute.

15.1 Threat Model

Threat	Vector	Mitigation
Sybil Attacks	Fake providers	Staking + identity checks
Fake Compute	False outputs	PoR + VCM
Censorship	Routing bias	Decentralized orchestration
Collusion	Routing manipulation	Multi-orchestrator audits
Downtime	Offline nodes	PoU + slashing
Data tampering	Storage nodes	Hash checks + PoS
Bandwidth fraud	Fake throughput	PoB

15.2 Slashing Conditions

Nodes are slashed for:

- Failed proofs
- Fraudulent compute
- Downtime beyond threshold

- Routing manipulation (orchestrators)
- Data tampering

Slashing amount:

```
Slash = λ_s • Stake (where λ_s = 10–30% depending on severity)
```

15.3 Assumptions

- Majority of nodes act honestly
- Cryptographic primitives remain secure
- Network partitions are rare but possible
- Multi-path fallback prevents censorship

16. Roadmap

Phase 1 — Testnet & Core Marketplace (2025)

Goals:

- Launch testnet with compute nodes
- Deploy initial routing logic
- Implement PoR, PoD, PoU
- Roll out staking for compute providers
- Provide LLM inference endpoints
- Release SDK (Python + JS)

Phase 2 — Decentralized Inference Layer (Late 2025–2026)

Upgrades:

- Token-staked orchestration nodes
- Fully decentralized routing
- Multi-orchestrator fault tolerance
- zk-based verification R&D
- Advanced pricing models

Phase 3 — Cross-Chain Compute + Institutionalization (2026–2027)

- Access compute from multiple blockchains
- Regulated access modes (KYC for large providers)
- Enterprise dataset and model marketplace
- Large-scale GPU onboarding (goal: 100,000 units)

Phase 4 — Global AI Infrastructure Layer (2027+)

Long-term goals:

- Edge inference network
- Confidential, zk-powered inference
- Multi-organizational model hosting
- AI service economy powered by AIDP

17. Compliance & Real-World Considerations

17.1 Compliance Principles

AIDP follows:

- KYC requirements for large providers (optional for small ones)
- Audit-friendly architecture
- GDPR-aligned data handling
- Enterprise contracts with legal review
- Market-integrity protections

The protocol minimizes personal data exposure by applying:

- Cryptographic proofs
- Pseudonymous staking identities
- Optional zk-based verification

17.2 Enterprise Compliance

For enterprise-grade clients:

- SLAs can be attached to inference requests
- Custom compliance routing (geo-limited compute execution)
- Private clusters (enterprise-exclusive compute pools)
- Custom billing and settlement agreements

17.3 Jurisdictional Constraints

Certain regions require:

- GPU provider identity verification
- Tax reporting from large-scale operators
- Usage and billing transparency
- Dataset usage compliance (copyright, model licensing)

AIDP's architecture allows modular toggle:

- Open, permissionless mode
- Regulated, enterprise-only mode
- Hybrid mode (per-workload enforcement)

18. Future Extensions

AIDP is designed with long-term extensibility.

18.1 Edge AI Inference Marketplace

Edge devices (IoT, small GPUs, industrial sensors) will eventually:

- Process real-time AI tasks
- Reduce latency
- Provide decentralized data collection

18.2 Inter-DePIN Interoperability

AIDP will interoperate with:

- Storage DePINs (Filecoin, Arweave)
- Compute DePINs (Akash, Render, io.net)
- Bandwidth DePINs (Helium)

Cross-network resource sharing enables broader ecosystem integration.

18.3 Sovereign Compute Clusters

For governments, enterprises, and regulated markets:

- Private subnetworks
- Policy-based routing
- Confidential compute environments
- Air-gapped inference clusters

These remain cryptographically verifiable while operating with privacy guarantees.

19. Conclusion

AIDP.store unifies decentralized compute, cryptographic integrity, programmable settlement, and a long-term sustainable economic system.

Through:

- Proof-of-Resource
- Verifiable Compute
- Decentralized Orchestration (Phase 2)
- Buyback & Burn tokenomics
- DAT treasury
- Multi-layer architecture
- Strong governance
- Roadmap to global-scale compute

AIDP is positioned to become a foundational layer for global decentralized AI infrastructure — enabling scalable, permissionless, cryptographically validated AI workloads.

20. Appendix A — System Diagrams

20.1 Resource Weight Table

Resource Type	Weight
Compute (GPU)	2.2
Bandwidth	1.5
Storage	1.0
Sensor	0.8

20.2 Buyback & Burn Parameters

Parameter	Value	Description
Buyback Ratio	40% of revenue	Allocated to market buybacks
Burn Ratio	20% of protocol fees	Permanently removes tokens
Slashing	10–30% of stake	Punishment for violations

20.3 Roadmap Table

Phase	Timeframe	Key Deliverables
Phase 1	2025	Testnet, basic routing, staking, PoR/PoD
Phase 2	2026	Decentralized orchestration, zk research, full mainnet
Phase 3	2026–27	Cross-chain compute, enterprise integrations, 100k GPUs
Phase 4	2027+	Global decentralized AI infrastructure

21. References

Web3 Protocol Research

- Filecoin Protocol Specification
- Helium Network
- Render Network Architecture
- Akash Decentralized Compute
- Solana Sealevel Runtime Documentation
- Ethereum Yellow Paper

Academic References

- Tanenbaum & Van Steen — Distributed Systems
- Verifiable Compute Research Papers
- Zero-Knowledge Cryptography Work
- Distributed Systems for AI Workloads
- GPU Scheduling & Multi-Tenant Compute Research