DSDN: Semi-decentralized and distributed data and computing network with blockchain

Date: October 16, 2025

Author: Noven Rizkia

novenrizkia.8.5.6@gmail.com

On behalf of: Biteva Researcher

Abstract: We propose DSDN, a semi-decentralized data and compute network that combines cross-zone data replication, isolated program execution, and a validator layer for legal compliance. DSDN provides content-based data placement, triple replication targets, Rust/Python execution on WASM or microVMs, and scheduling that takes data and resource location into account. Validators verify public-serving applications without accessing encrypted private data. The primary goals are high availability, measurable costs, and broad public participation, with room for government to act as a monitor for illegal content, not as a data controller.

DSDN is designed as a verifiable-by-design system, where no single entity, including coordinators, validators, or foundations, holds authoritative control over data, execution, or network state. All system metadata is recorded in the Data Availability layer, which is public and can be deterministically replayed, allowing any node, independent auditor, or third party to verify the correctness of the network state without requiring implicit trust in any

operator. With this approach, DSDN prioritizes auditability, transparency, and limiting structural power as the foundation for long-term trust.

1. Introduction

The modern internet runs on a highly concentrated infrastructure. Most of the world's data, computing, and digital services are operated by a small number of global data centers. This model provides efficiencies of scale, but also creates structural concentrations of power, censorship risks, geopolitical dependencies, and systemic single points of failure.

While the blockchain revolution has successfully decentralized the value layer (money layer) and consensus, it has not yet decentralized the data and computing layers. The majority of Web3 applications remain reliant on traditional cloud infrastructure for storage, APIs, and program execution. Therefore, current Web3 architecture still relies on Web2 foundations.

On the other hand, a fully decentralized approach without compliance boundaries faces institutional adoption barriers and legal risks. Many countries demand moderation mechanisms and compliance with national laws, especially for public services, without granting direct control over users' private data.

This gap creates a new design space.

The Data Semi-Decentral Network (DSDN) is proposed as a data and computing infrastructure that:

- Strictly separates the data plane, compute plane, and governance plane.
- Adopts the principle of verifiable-by-design, rather than trust-based governance.
- Combines permissionless public participation with a limited, identity-verified compliance layer.
- Provides multi-zone replication, isolated execution, and deterministic auditing through a Data Availability layer.

The goal of DSDN is not simply to be an alternative storage or computing network, but rather to become a fundamental infrastructure capable of supporting large-scale systems—from education and national archives to private communications and AI computing—without creating new concentrations of power.

DSDN does not claim to be a completely trustless or censorship-resistant system. Instead, it explicitly defines boundaries of authority, technical restrictions, and audit mechanisms so that every action can be verified, recorded, and challenged.

With this approach, DSDN seeks to bridge two extremes: complete centralization and limitless decentralization, through an architecture that is scalable, auditable, and ready for widespread adoption.

2. System Summary

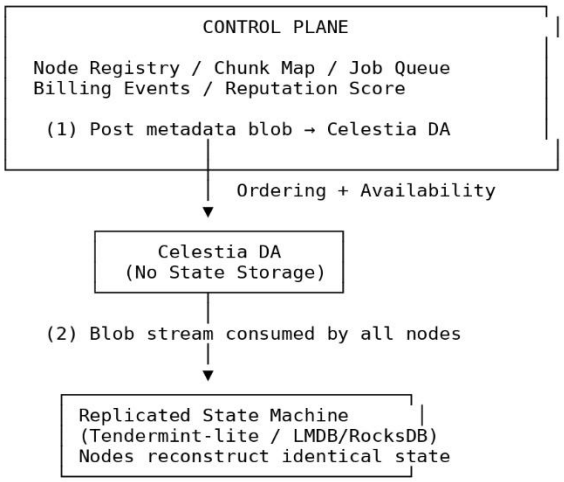
DSDN consists of three complementary planes:

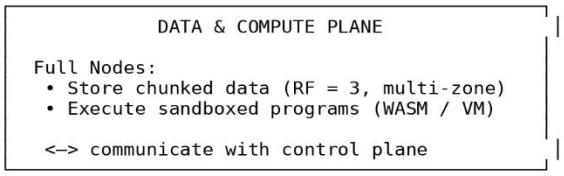
1. **Control plane** — Control plane metadata (node registry, chunk map, job queue, billing events, reputation score) is posted as blobs to the Data Availability layer (Celestia) for ordering and availability. Each node then consumes blobs from Celestia and builds local state through a replicated state machine (Tendermint-lite or a log-sink KV store like RocksDB/LMDB). Celestia does not store state, only guarantees the ordering and availability of blobs. The same state is deterministically reconstructed by each node from an identical sequence of blobs.

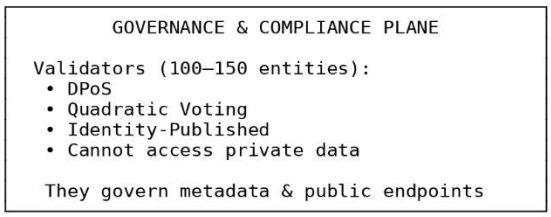
2. **Data and compute plane** operated by full nodes. Nodes store chunked data in three replicated chunks in different zones and execute programs in a sandbox.

3. **Governance and compliance plane,** Run by validators. The governance center can be run by up to 100-150 validators, which can be companies, organizations, or institutions. These 100-150 validators must stake a minimum of 50,000 tokens, with 20% of the stake being individuals with a minimum stake of 100,000 tokens. The DPoS mechanism is used, followed by quadratic voting from regular token holders. Validators must publish their identities.

This is the illustration:


CONTROL PLANE
Node Registry / Chunk Map / Job Queue
Billing Events / Reputation Score
(1) Post metadata blob → Celestia DA
Ordering + Availability
Celestia DA
(No State Storage)
(2) Blob stream consumed by all nodes
Replicated State Machine
(Tendermint-lite / LMDB/RocksDB)
Nodes reconstruct identical state


DATA & COMPUTE PLANE
Full Nodes:
• Store chunked data (RF = 3, multi-zone)
• Execute sandboxed programs (WASM / VM)
<–> communicate with control plane


GOVERNANCE & COMPLIANCE PLANE
Validators (100–150 entities):
• DPoS
• Quadratic Voting
• Identity-Published
• Cannot access private data
They govern metadata & public endpoints

2A. Trust Model and Security Assumptions

DSDN explicitly defines trust boundaries between system components. The primary goal of this design is to minimize the number of entities that must be trusted and ensure that each trust assumption is technically verifiable.

Key trust assumptions:

1. The Celestia Data Availability layer is assumed to be honest in guaranteeing blob ordering and availability, but is not trusted to store or execute state.

2. Storage and compute provider nodes are assumed to be untrusted by default, and their behavior is verified through replication, quorum, and slashing mechanisms.

3. Validators are assumed to be untrusted to maintain data privacy, as they technically do not have access to encrypted data or decryption keys.

4. The Coordinator is not trusted as a source of truth; it acts solely as a stateless scheduler whose decisions can be reconstructed and audited through the Data Availability log.

With this model, failure or malicious behavior of one or more components cannot unilaterally change the correctness of the network state; it can only cause temporary service degradation that can be detected and audited.

3. Network Model and Roles

DSDN supports two classes of nodes to facilitate public participation and lower operational costs, without sacrificing service quality for large-scale workloads.

Regular Full Nodes

Storage limited data capacity, run lightweight programs, and serve small-to-medium user requests. These nodes can be run on home-grade devices or small servers, making community participation more affordable. Regular full nodes still receive rewards based on their actual contributions (storage, compute, bandwidth), but their smaller capacity means the total reward received tends to be lower..

Stake requirement: Regular full nodes are required to stake a minimum of 500 $NUSA to register on the network. This stake serves as Sybil resistance and guarantees honest behavior. Stakes will be slashed if the node is found to have committed violations such as providing corrupt data or being unresponsive for an extended period.

Data Center Class Full Nodes

Provide significantly greater capacity for storage, high bandwidth, and intensive computing, including GPU loads. This class of nodes is the scheduler's preferred choice when jobs require high reliability, throughput, or SLAs. Rewards still follow the general DSDN mechanism (70% storage/compute fee), but due to the higher work volume, the total reward received is higher.

Stake requirement: Data center class full nodes are required to stake a minimum of 5,000 $NUSA. A larger stake reflects higher capacity and responsibility, and provides Sybil resistance proportional to the resources claimed.

Anti-Self-Dealing

To prevent reward manipulation, DSDN implements an anti-self-dealing rule: nodes do not accept rewards from workloads submitted by the same wallet address or affiliated wallet addresses (determined through simple on-chain graph analysis). If self-dealing is detected, rewards are diverted to the treasury, and the node receives a warning. Repeated violations result in stake slashing.

Validator Node

Validators enforce the governance and compliance plane. Validators only inspect public application metadata and cannot access end-to-end encrypted private data. Validators can double as data center-class nodes, but must still meet staking and compliance requirements.

In addition to validators, public token holders (non-validators) also play a role in governance through the Quadratic Voting (QV) mechanism. In this scheme, each account can allocate votes based on the square root of the number of tokens staked, so that a token holder's vote is calculated as vote = $\sqrt{}$stake_user. This approach balances power between large and small token holders: whales cannot dominate decisions simply because they hold a large number of tokens, but small users still have significant weight in the governance process. QV is used for proposals related to fee adjustments, the Node Cost Index, network technical parameters, and non-compliance policies. For severe compliance actions—such as blocking public endpoints—validators remain the dominant actors, but community votes are still counted through QV as a secondary check to ensure decisions remain balanced, transparent, and not easily manipulated by any single party.

Validator authority limitations:

1. Validators cannot delete or modify actual data stored on the node. Encrypted data is technically inaccessible to validators.

2. Validators can only delete or block pointers/endpoints—metadata references that point to specific data or applications. Chunks of data remain on the node until they expire or are deleted by the owner.

3. Validators do not replace the role of full nodes in providing storage or compute.

4. Validators do not directly manage applications. Validators require agreement from other validators and a portion of the community (but still a dominant validator group) or a governance process to take actions such as blocking applications, halting a system, and so on.

5. All compliance actions must go through on-chain governance with a 7–30-day delay, multisig approval from at least 60% of validators, and an appeals mechanism.

6. If the content is truly illegal (terrorism, CP), let law enforcement take the usual off-chain legal route.

coordinator

Stores global metadata, partition maps, node states, and job queues. The coordinator is run by multiple replicas that consume blobs from Celestia DA and deterministically construct local state. The coordinator does not maintain its own authoritative state—state is reconstructed from Celestia logs. The coordinator remains neutral and is not involved in storing user data.

The coordinator in DSDN has no authoritative authority over state, rewards, or final execution. All coordinator decisions including workload scheduling and receipt signing must be deterministically reconstructable by other nodes based on the order of events posted to the Data Availability layer.

To mitigate the risk of power concentration, DSDN supports a multi-coordinator model, where multiple coordinator instances run in parallel and independently. A receipt is considered valid only if it meets on-chain verification criteria and is consistent with the state reconstructed from the Data Availability log. In advanced designs, receipt signatures can be enhanced to a quorum scheme (e.g., 2 out of 3 coordinators) or reverified by other nodes as a mechanism.

Minimal Operational Requirement

The Nusantara Blockchain can technically operate with just one validator node for consensus and block finality. This allows for testing, initial bootstrapping, or emergency operations.

DSDN, as a storage and compute system, has different requirements:

1. A minimum of 3 full nodes across 3 different zones** is required to guarantee durability (RF=3) and meet availability SLAs.

2. With fewer than 3 nodes or 3 zones, the system enters **degraded mode**: data does not meet the RF=3 target, the SLA is not guaranteed, and the system displays a warning to the user.

3. The figures of 100-150 validators and 3+ nodes are formal operational targets for production environments, not hard requirements for a viable system.

Summary: A blockchain can survive with 1 validator. DSDN storage/compute requires a minimum of 1 node to survive but requires nodes across 3 zones for full durability.

Bootstrap Phase

In the early phase of the network, when the Replication Factor (RF) is 3, node and validator participation is bootstrapping-oriented, not profit-oriented. Nodes and validators in this phase are operated by trusted parties (the foundation, early

partners, or internal operators) with the primary goal of providing system stability and initial user acquisition, rather than maximizing revenue.

In this phase:

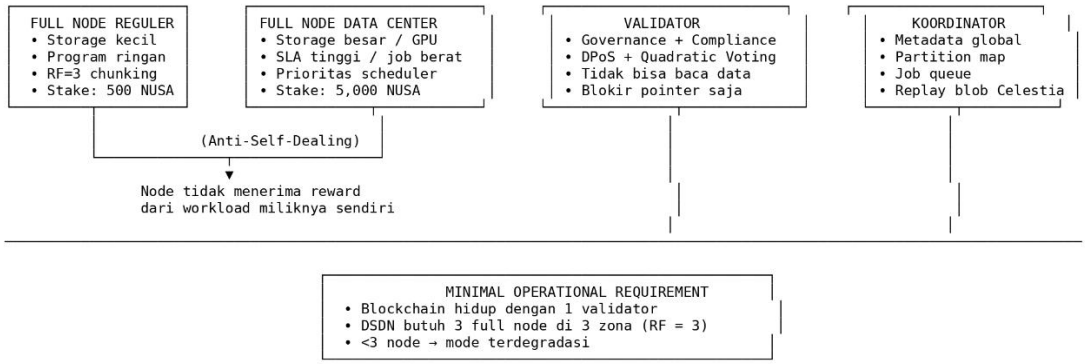
Service fees are set well below normal market rates.

Economic rewards for nodes are minimal or subsidized.

The primary goals are technical validation of the system, user acquisition, and generating real demand.

As the number of nodes and zones increases (RF > 3), the network gradually transitions to a normal economic mode, where public nodes begin operating with full economic incentives.

Ilustration:



4. Data Model and Replication

Application objects are divided into 16–64 MiB chunks and identified by hashes. Chunk placement uses consistent hashing with virtual nodes that recognize two capacity classes: regular nodes and data center nodes. The replication target remains RF = 3, but the system prioritizes placement on data center nodes when files are large or require a high SLA. Chunks can still be placed on regular nodes as long as they meet capacity and zone limits.

If the number of nodes is insufficient to achieve three replicas in three different zones, the network enters degraded mode with a lower RF and self-heals when capacity becomes available, regardless of node class. **Degraded mode should be avoided for production workloads as it does not meet the durability guarantee.**

Consistency. Write and read operations use a quorum of replicas, for example, write on two out of three and read on two out of three. For massive datasets, 6+3 erasure coding can be used to reduce costs on high-capacity nodes (typically data centers).

Availability. Replica distribution maintains three distinct failure domains. The combination of regular nodes and data center nodes in a single city still counts as a single failure domain; the domains must be completely different physically to approximate the failure independence assumption.

Privacy and keys. All objects are encrypted client-side or at upload time. Keys are managed by the application owner, with recovery using Shamir Secret Sharing 2-of-3 on a user-selected party. Nodes, both regular and data center, never have access to the decryption keys. **The validator also does not have access to the decryption key and cannot read the contents of the encrypted data.**

Consistency. Write and read operations use a quorum across three replicas, for example, write to two of three, read to two of three. **Erasure coding**, such as the 6+3 scheme, can be used for large datasets to reduce storage costs without sacrificing durability.

Availability. If the probability of failure of a single zone is *p* and it is assumed to be independent, the probability of losing all data is at worst *p*³, so data availability approaches 1 − *p*³. Zones must be physically isolated to approximate the independence assumption.

Privacy and keys. Application data is encrypted client-side or upon upload. Keys are managed by the application owner. Key recovery utilizes Shamir Secret Sharing with a 2-of-3 scheme with a trusted party selected by the owner.

Data owners can initiate object deletion through the User-Controlled Delete mechanism. When a user requests deletion, the system creates a chunk-key blacklist entry that is published as metadata in the control plane. The node holding the chunk marks the chunk as “expunged” and schedules a local garbage collection that deletes its physical data within a period of 7–30 days, depending on the node class. Because encrypted data cannot be decrypted by the node operator, this process is entirely metadata-based and does not involve inspecting the data content. With this approach, DSDN supports the user's right to delete data while maintaining a consistent content-addressed architecture.

5. Program Execution

DSDN runs four classes of workloads: static websites, stateful web services, batch jobs, and long-running processes like blockchain nodes or AI training. The selection of nodes to run workloads takes into account both the node class and the computational requirements.

Isolation. Rust programs are compiled to WASM/WASI whenever possible, while Python and native Rust run on microVMs based on Firecracker or similar technologies. On regular nodes, cgroup constraints are more stringent to maintain stability. Data center nodes can run large workloads, including GPU computing with time slicing or MIG.

Hybrid. Users can allocate some of the computation to their own devices, while the remainder runs on regular nodes or data centers as needed. The client agent adjusts the parameter synchronization mode (server parameters or federated averaging) based on the node class involved. Regular nodes are suitable for light-duty tasks and small inferences, while data center nodes handle large inferences and medium-scale training..

Isolation. Rust programs are compiled to WASM/WASI whenever possible, while Python and native Rust run on a microVM based on Firecracker or similar technology. Resources are controlled by cgroups. GPU load uses time-slicing or MIG scheduling when available.

Hybrid. Users can split computation between their own devices and the network. Client agents run a portion of the computation locally and synchronize parameters via a server parameter scheme or federated averaging.

6. Scheduling and Placement

Scheduling considers capacity, node class, and data proximity. The node score is defined as:

$$S = w_1 \cdot \text{CPU\_free} + w_2 \cdot \text{RAM\_free} + w_3 \cdot \text{GPU\_free} + w_4 \cdot (1/\text{latency}) - w_5 \cdot \text{IO\_pressure} + w_6 \cdot \text{class\_weight}$$

Where:

The class_weight assigns additional weight to data center-class nodes for heavy workloads (e.g., 1.0 for data center nodes, 0.3 for regular nodes).

For light or personal workloads, the weight is neutral to allow regular nodes to remain in use.

The scheduler selects the node with the highest score and retains a warm standby for fast failover. Chunk replication follows consistent hashing, which takes into account the remaining capacity and the proportion of regular nodes to data center nodes.

Dynamic migration can occur when regular nodes approach capacity limits. In such cases, additional replicas are moved to the data center nodes through lazy replication.

Anti-self-dealing in scheduling: The scheduler will not place workloads on nodes owned by the same wallet address as the submitter. This is verified through the on-chain ownership registry.

Receipt Verifiability and Audit

Each receipt generated in the DSDN is published in hash form to the Data Availability layer, allowing other nodes and independent auditors to verify the existence, sequence, and consistency of receipts against the claimed workload execution. In the event of any discrepancy—such as delays, missing receipts, or metadata manipulation—the aggrieved node can present evidence based on the Data Availability log to challenge the validity of the receipt through an on-chain dispute mechanism.

With this approach, the coordinator functions not as a trusted notary, but rather as a facilitator whose actions can be verified and, if necessary, challenged.

7. Security and Privacy

Node identities use Ed25519 keys. Nodes connect via WireGuard or QUIC with mTLS. Program artifacts are signed and equipped with SBOM. The sandbox restricts access through seccomp, AppArmor, non-root users, and read-only file systems whenever possible. OIDC is used for user authentication with short-lived tokens. For sensitive workloads, enclaves such as SGX or SEV SNP can be enabled.

Sybil-resistance: Each node is required to stake tokens according to its class (500 $NUSA for regular nodes, 5,000 $NUSA for datacenter nodes). Stakes are slashed if the node misbehaves. This mechanism prevents Sybil attacks, where a single entity registers multiple fake nodes to manipulate rewards or voting.

Node slashing is applied in stages based on the severity of the violation. (1) Liveness failure—a node being unresponsive or failing to serve a chunk for more than 12 hours—results in a mild slashing of 0.5% of the stake. (2) Data corruption—a chunk lost or corrupted on two consecutive quorum verifications—results in a 5% slashing and a 14-day cooldown period before the node can re-register. (3) Repeated malicious behavior—such as repeated data corruption or reward manipulation—results in a forced unbond and a 30-day node ban. This mechanism creates a strong incentive to maintain data integrity and service quality without excessively penalizing public participation.

In the context of Indonesia's strict regulations against storing illegal content, DSDN implements a security model where public nodes cannot be requisitioned.

accountability for the encrypted data they store. Because data cannot be decrypted without the application owner's key, node operators are technically and legally unable to know its contents. The system only exposes public pointers to validators; if a pointer is blocked, application access is disabled without deleting the actual data, allowing legal recourse to law enforcement rather than the node operator. This approach ensures public nodes are legally secure, in line with the ITE Law, the Minister of Communication and Information Technology Regulation, and the infrastructure provider's safe harbor principles.

8. Moderation, Validators, and Compliance

Validators apply declarative rules to public applications. An automated pipeline checks for blacklist hashes, malware signatures, and traffic anomalies. Validator actions are limited to deleting or blocking pointers/endpoints, not actual data. All events are recorded in an auditable WORM log.

For private applications, validators do not see the data content, only minimal metadata and a hash trail. Validators cannot technically access encrypted data because they do not possess the decryption key. Proofs without revealing the data can be added later using practical zero-knowledge techniques.

Validators in DSDN are not positioned as "trusted" parties to maintain data veracity or user integrity. Instead, validators perform only limited compliance functions for public services, with deliberately limited authority and oversight through on-chain mechanisms. Every validator action is procedurally reversible, subject to execution delay, and publicly auditable. With this design, DSDN eschews the actor-based trust model and replaces it with trust based on technical constraints and transparency.

Authority Clarification:

1. Validators can: block public endpoints, delete metadata pointers, flag applications for review.

2. Validators cannot: read encrypted data, delete chunks from nodes, or access user keys.

To prevent abuse of validator authority, the system defines Rules of Engagement (RoE) for moderation actions. Validators can only propose blocking of pointers if the

application or public endpoint meets certain categories: (1) serious criminal content such as terrorism and child exploitation, (2) active malware or structured fraud, (3) clear and verified violations of telecommunications law. Content that is political, social criticism, personal expression, or information that is not illegal cannot be used as a basis for validator action. For gray categories, decisions must go through a multi-stage review that includes validator voting (at least 60%), Quadratic Voting from the community, and a 7–30 day delay to ensure transparency. With this approach, DSDN maintains a balance between legal compliance, freedom of expression, and protection against abuse of power..

DSDN uses a hybrid governance model that combines off-chain discussion and on-chain decision-making. Proposals are initiated through a public forum or discussion repository, allowing for technical analysis, legal argumentation, and community contributions. After going through the draft stage, proposals are published on-chain as governance items and voted on using two layers: validator voting and community Quadratic Voting. This model mimics the practices of large networks like Cosmos, Optimism, and Arbitrum, where complex deliberations are more efficiently conducted off-chain, while final decisions remain final and transparent through on-chain governance.

8. A Progressive Governance Model

To ensure the stable operation of DSDN from its initial launch phase to full decentralization, the DSDN governance system is designed using a progressive governance approach. In this model, system liveness is independent of the presence or number of governance validators, while authority distribution occurs gradually as the network grows and matures.

The main principle of this design is that the data plane, compute plane, and Nusantara blockchain must remain able to operate automatically, deterministically, and independently, even when governance is not yet active or fully operational.

Phase 0 — Foundational / Pre-Governance Mode

In the initial phase of the network launch, when the number of validators is still below the minimum threshold or not yet available, DSDN operates in Pre-Governance Mode.

Key characteristics:

There is no public governance, voting mechanisms, or on-chain proposals.

All system parameters are static and set through the initial protocol configuration.

Storage, compute, scheduling, billing, and replication operations are fully automated based on predefined technical rules.

System continuity does not depend on a governance quorum or validator approval.

In this phase, limited operational authority is held by the Foundation Key (or Founder Authority) with an explicitly limited scope, including:

Implementing technical parameter updates.

Limited emergency actions, such as temporarily suspending a problematic public endpoint.

Deploying security patches or fixing critical bugs.

The Foundation Key does not have access to encrypted user data, cannot delete data chunks, and does not have the authority to move or withdraw user funds. All Foundation actions are transparently recorded as on-chain events and are auditable.

Phase 1 — Bootstrap / Semi-Governance Mode

When the network reaches at least three active validators (RF validators ≥ 3), DSDN enters Bootstrap Governance Mode. In this phase, the governance module is technically active, but does not yet have full authority to execute protocol decisions.

Key characteristics:

Validators can submit proposals and vote, but voting results are recommendatory (non-binding).

Governance proposals are not automatically executed without Foundation approval.

Governance serves as a means of testing the voting mechanism, coordinating between validators, and simulating protocol policies.

Governance-based slashing has not been fully implemented; only automated mechanisms such as liveness failure are active.

The power structure in this phase is semi-centralized, with the following division of roles:

The Foundation has limited override and veto rights.Validator berperan sebagai aktor advisory serta penguji stabilitas mekanisme governance.

> Komunitas dapat berpartisipasi secara terbatas, misalnya melalui simulasi voting tanpa dampak langsung terhadap eksekusi protokol.

All override or veto actions taken by the Foundation must be transparently recorded on the blockchain as part of the audit trail.

Phase 2 — Transition Governance Mode

When the number of validators increases and reaches a certain level of stability (e.g., 50–99 active validators), the network enters Transition Governance Mode.

Key characteristics:

Governance proposals begin to be executed automatically after a predetermined delay period.

The community's Quadratic Voting (QV) mechanism begins to be factored into the decision-making process.

Governance-based slashing is implemented in a limited manner, along with an appeal mechanism.

The Foundation's role is gradually reduced and is no longer absolute.

In this phase, authority is distributed in tiers between the Foundation, validators, and the community, with the goal of preparing for the transition to full decentralization without compromising system stability.

Phase 3 — Full Governance Mode

When the network reaches full operational scale with 100–150 active validators, DSDN enters Full Governance Mode, as defined in the main design whitepaper.

Key characteristics:

There is no longer any veto or override power from the Foundation.

All governance decisions are executed entirely through on-chain mechanisms.

Validators are the primary decision-makers, supported by Quadratic Voting from the community.

All governance actions follow the Rules of Engagement (RoE), an execution delay period of 7–30 days, a multi-signature validator mechanism, and an appeals system.

In this phase, the Foundation acts as a regular participant (e.g., a validator or contributor), without any special authority at the protocol level.

Governance Transition and Security

DSDN governance modes are represented as protocol state variables (governance_mode) that affect governance module behavior and policy execution. Governance mode changes can only occur through:

Meeting predefined technical thresholds, such as the number of active validators and network stability, or

On-chain governance decisions in later phases.

With this phased governance approach, DSDN avoids the risk of premature decentralization, maintains system stability in its early phases, and provides a clear, transparent, and measurable path to open governance and full decentralization.

9. Operations and Observability

DSDN collects standard metrics for CPU, RAM, GPU, I/O, capacity, and replication health. Logs and traces are partitioned per tenant. Alerts are issued when capacity is low, replicas fall short of target, jobs stall, or nodes experience frequent node downtime. The network maintains a 20 to 30 percent capacity reserve to absorb sudden spikes and failures.

State reconstruction: Because state is built from Celestia blobs, each node can perform state recovery by replaying the blob from its genesis. This makes debugging and auditing easier.

---

10. Scale, Cost, Economics, and Availability

With triple replication, storage costs are three times the size of the raw data without erasure coding. For large datasets, a 6+3 scheme reduces overhead to 1.5x with comparable durability. Read latency decreases when data is placed in the three closest zones with DNS anycast. The cost model can be shared between

community node operators and a central facility that provides networking and traffic balancing.

Economics:

DSDN was created alongside a blockchain for transactions and validation of the overall DSDN system. Those running DSDN nodes will also run the blockchain. Nodes are paid with DSDN's internal token ($NUSA), which is credited to the node owner's wallet to encourage people to run nodes. This blockchain, called Nusantara, uses proof-of-stake consensus on DSDN validator nodes. The maximum DSDN token supply ($NUSA) is 300 million.

SEGALA AKTIVITAS DI DSDN DIBAYAR MENGGUNAKAN $NUSA COIN,

BLOCKCHAIN SPECIFICATIONS:

1. ACCOUNT-BASED, NOT UTXO LIKE BITCOIN

2. PRIMARY HASH IS SHA3 512

3. Consensus -> PoS

4. Db -> LMDB

5. SEMI-DECENTRAL BECAUSE EVERYONE CAN RUN NODES AND SYSTEMS, BUT COMPANY-OWNED VALIDATORS CAN CONTROL TO KEEP DSDN ETHICAL (controlling involves removing only explicit content through pointer/endpoint deletion, the rest is managed by the community and governance).

The Nusantara Blockchain is designed with medium throughput to support DSDN metadata operations without becoming a bottleneck. The initial target is 500–1,500 transactions per second, with a block time of 2–4 seconds and deterministic finality of less than 5 seconds through Proof-of-Stake consensus. This capacity is more than sufficient because heavy operations like data uploads, chunk replication, and computational execution don't occur on the blockchain, but rather in the data plane. The blockchain only records user transactions, staking, governance, billing events, and metadata pointers, so the transaction load is relatively stable and manageable without requiring the extreme scale of a global L1 network.

Control plane architecture: Blockchain state and DSDN are not stored in Celestia. Celestia only provides a Data Availability layer for blob ordering. Each node builds its local state by sequentially consuming blobs and executing state transitions

deterministically. This approach is similar to Tendermint-lite or a log-structured state machine.

If you want to perform activities on DSDN, such as uploading files, encrypting files, or running runtimes, all of these activities are connected to the user's wallet (address & privacy key), and the user performing the activity will pay a gas fee and sign a digital signature.

Full text:

"The $NUSA token has a maximum supply of 300 million tokens with no new minting mechanism, so the supply is deflationary over time. The initial distribution is conservative to maintain the project's credibility: 60% is allocated to the "Node & Validator Reward Pool" which is released gradually over twenty years, 20% to the Foundation and research with a five-year vesting period, 10% to community grants, 5% to early testnet node rewards, and 5% for liquidity needs. The largest portion is indeed directed to nodes and validators because they run the entire DSDN infrastructure.

Nodes earn revenue directly from user activity. Every operation such as uploading or downloading data, storing files, running WASM or Python runtimes, performing AI inference, or running long-running applications generates fees in $NUSA. These fees are automatically divided: 70% goes to nodes providing resources (storage or compute), 20% to validators as staking rewards, and 10% goes to the treasury for auditing, development, and the burn mechanism. Note: Nodes do not receive rewards from workloads they generate. submitted by its own wallet address (anti-self-dealing). With this model, nodes don't rely on inflationary rewards but instead earn income from the actual services used by users.

The rates listed in this section represent the standard network rates applicable during the normal economic phase.

During the network bootstrapping phase, particularly when the Replication Factor (RF) is still at its minimum value (RF = 3), service fees can be significantly reduced from these standard rates to encourage early user adoption and reduce barriers to entry.

Cost increases toward standard rates are implemented gradually as network capacity, the number of nodes, and the replication rate increase (RF > 3), ensuring a stable economic transition without incurring cost shocks for users.

Stake requirement for node:

- Regular full node: minimum 500 $NUSA
- Data center full node: minimum 5,000 $NUSA
- Validator: minimum 50,000 $NUSA (as listed in Part 2)

Validators in a Proof of Stake system receive two revenue streams: 20% of all user activity fees and a small annual reward of up to 1% for staking incentives. Validators are required to maintain a substantial stake to keep the network stable and less vulnerable to attacks.

To keep the token price healthy and fees reasonable for users, the system uses three stabilization mechanisms. First, a portion of the treasury is periodically burned to gradually reduce the supply. Second, gas fees are adaptive and can adjust to the token price: when the price rises, the base fee can be lowered, and vice versa. Third, DSDN uses a "Node Cost Index," an on-chain module that monitors factors such as electricity prices, storage costs, bandwidth, and network load. This index proposes periodic fee adjustments, which are then voted on by validators. This way, fees remain in line with economic realities without burdening users or disadvantageous nodes.

A simple simulation shows that a user storing 100 GB of data, uploading 10 GB, downloading 30 GB, and running 100 CPU minutes and 10 GPU minutes per month spends approximately 18.9 $NUSA. With a token price of around Rp10,000, the monthly fee is approximately Rp189,000—much lower than commercial cloud services. The node serving that user receives approximately 13.23 $NUSA. If a single node has fifty active customers, its total monthly revenue can reach approximately 661 $NUSA, or approximately 6.6 million rupiah, enough to cover operating costs and provide a stable profit.

With this mechanism, DSDN has tokens that are actually used for services, not just speculation. Fixed fees are user-friendly, nodes receive consistent revenue, validators have long-term incentives, and the token supply is secure. This entire structure creates a sustainable ecosystem without relying on hype.

10.1 Deflationary Mechanism and Stability of $NUSA Token Value

The $NUSA token is designed as an infrastructure utility token with a fixed maximum supply and a controlled deflation mechanism. The primary goal of this design is to maintain the token's long-term purchasing power and protect the DSDN ecosystem from fiat currency inflationary pressures.

$NUSA's deflation is not fixed, but rather based on the network's economic activity, with an annual deflation target of 3–6%. This deflation mechanism consists of several layers:

1. Periodic Treasury Burn

A portion of the treasury funds derived from network fees are periodically burned. The burn rate is adjusted according to the level of network activity and economic conditions, ensuring controlled deflation.

2. Usage-Based Burn

Every storage and compute activity results in a small additional burn. The higher the network usage, the greater the deflationary pressure.

3. Slashing as a Deflationary Sink

A portion of tokens slashed due to node or validator breaches are burned, while the remainder is allocated to the treasury. This approach increases security while increasing deflationary pressure.

4. Supply Locking through Staking

Tokens staked by nodes, validators, and delegators are locked for a specified period, reducing the effective circulating supply and strengthening the token's value stability.

With this approach, the value of $NUSA is backed by real network usage, not mere speculation, and is designed to remain stable and inflation-resistant in the long term.

11. Reference Implementation

The primary languages are Rust and Python. Key components:

* Coordinator (Rust) — consumes blobs from Celestia and builds local state via a replicated state machine.

* Object storage (Rust) with content addressing, consistent hashing, and self-healing.

* WASM runtime and microVM for secure execution.

* HTTP/HTTP3 ingress with geography-aware routing.

* Client agent for hybrid compute and parameter synchronization.

* State machine (Rust) — log-sink KV store for state reconstruction from Celestia blobs.

The minimal interface between components is exposed through simple APIs for application state, data placement, and policy validation.

-

12. Conclusion

DSDN offers a practical path for a distributed, auditable, and privacy-preserving data and computing network. This system reduces reliance on a single data center, shares the electricity load, and opens up broad public participation, while still allowing for law enforcement in public services. This architecture is simple enough for gradual implementation, yet robust enough to support education, archives, communications, and AI computing in Indonesia.

System Limitations and Non-Objectives

DSDN does not claim to be completely trustless or immune to all forms of censorship and failure. It does not promise absolute anonymity, does not eliminate the risk of collusion entirely, and does not replace established legal processes. Instead, DSDN consciously chooses a verifiable, auditable, and constrained approach, where every form of power is limited, recorded, and can be challenged.

By explicitly defining what it does not promise, DSDN aims to build long-term trust through honesty of design, rather than absolute claims that are difficult to verify.

Appendix B. Compact Algorithm

Replica Placement: Calculate the object's hash, select the next three nodes on the ring that are in different zones and have sufficient capacity. If there are fewer than three, mark them as degraded and reschedule replication when available.

Self-Healing: The background agent scans for degraded objects, creating new replicas until it returns to the target RF.

Schedulability Score: Use the formula in Section 6 with calibrated weights. Apply preemption to maintain SLOs in the Critical class.

Anti-Self-Dealing Check: Before assigning rewards, verify that the node owner ≠ the submitter's workload through the on-chain registry.

-

Appendix C. Reference Node Specification (Revision Two Node Classes)

1. Regular Full Node (Public Participation)

Designed to allow public participation without high hardware costs.

- CPU: 4–8 vCPUs
- RAM: 8–32 GB
- Storage:
- NVMe 512 GB – 2 TB (hot tier)
- Optional 2–4 TB HDD (warm tier)
- Network: 300 Mbps – 1 Gbps

- GPU: Optional (if present, used for light computing)
- Power: UPS 5–10 minutes (optional)
- Location: Home, small office, or low-cost server
- **Stake: Minimum 500 $NUSA**

Regular nodes still play a role in chunk replication (RF=3), but the scheduler will direct heavy work to the data center nodes.

2. Full Data Center Class Node (High Capacity)

Designed for large storage, critical workloads, and heavy compute.

- CPU: 32–64 vCPU
- RAM: 128–256 GB
- Storage:
- 4–8 TB NVMe for the hot tier
- 8–16 TB HDD for the warm/cold tier
- Network: 10–25 Gbps with two uplinks
- GPU: 24–48 GB VRAM (optional but recommended)
- Power: 30-minute UPS + on-site generator
- Location: Minimum three different zones per city for high availability
- Stake: Minimum 5,000 $NUSA

This type of node will be selected more frequently for large workloads, earning greater rewards due to its capacity and stability.

---

1. **Control-plane architecture (Parts 2, 10, 11):** Clarified that Celestia is for DA/ordering only, with state built locally via a replicated state machine (Tendermint-lite / log-sink KV store).

2. **Validator limitations (Parts 3, 4, 8):** Emphasized that validators can only delete pointers/endpoints, not actual data. Validators cannot access encrypted data.

3. **Sybil-resistance & anti-self-dealing (Parts 3, 6, 7, 10, Appendix B, C):** Added stake requirements for nodes (500 $NUSA regular, 5,000 $NUSA data center). Nodes do not receive rewards from self-submitted workloads.

4. **Minimum operational requirement (Part 3):** It is clarified that blockchain can run with 1 validator, but DSDN storage/compute requires a minimum of 3 nodes across 3 zones for durability and SLA.