

# Blockchain Governance and Tokenomics

Vadym Khyzhniak, Holger Kinkel\*  
\*Chair of Network Architectures and Services

School of Computation, Information and Technology, Technical University of Munich, Germany

Email: vadym.khyzhniak@tum.de, kinkel@net.in.tum.de

**Abstract**—Resource sharing among distributed research testbeds demands contribution rewards, transparent membership, and fair use of hardware. We propose a dual-token Decentralized Autonomous Organization (DAO) on Algorand — “cash” Payment Tokens for real-time resource payments and “prepaid” Governance Tokens for contribution-based voting. We define initial issuance, token lifecycles, and two core workflows (new-member onboarding and monthly distributions tied to usage) and add safeguards to prevent pool depletion and abuse. Our framework focuses on tokenomics and offers a simple, automated solution for equitable testbed coordination.

**Index Terms**—DAO, Algorand, tokenomics, payment tokens, governance tokens, resource sharing, testbed coordination

## 1. Introduction

Resource sharing across multiple research testbeds poses specific challenges: how to reward contributions, ensure fair access to hardware, and make membership decisions in an optimal, transparent way. A *Decentralized Autonomous Organization* (DAO) built on Algorand can address these challenges by encoding payment and governance rules directly in on-chain logic.

### 1.1. Goal

This paper examines the tokenomics of a DAO that coordinates resource sharing among distributed research testbeds. By leveraging *Algorand Standard Assets* (ASA), the DAO issues two token types (proposed by us):

- **Payment Tokens**, used to “book” specialized hardware resources (GPU clusters, 6G testbeds, etc.),
- **Governance Tokens**, used to vote on membership and parameter changes.

Our objectives are to:

- Define the roles and lifecycle of each token type.
- Describe the initial minting and allocation to founding sites and the seeding of on-chain pools.
- Explain how those pools function under “cash” versus “prepaid” models.
- Analyze critical corner cases — such as empty pools and inactive token holders.
- Identify open questions and recommendations for sustainable operation.

### 1.2. Outline

The paper is structured as follows:

- **Section 2** presents a concise overview of the DAO testbed scenario and defines the key concepts and terminology (DAO, governance processes, Algorand smart contracts, ASA).
- **Section 3** analyzes our dual-token model (Payment & Governance), details initial issuance and ongoing token flows, including member admission workflows, and addresses critical corner-case protections.
- **Section 4** discusses remaining open questions and directions for future work, such as parameter tuning, cycle lengths, and replenishing mechanisms.
- **Section 5** concludes with the main takeaways and an outlook on prototyping.

Following this structure, the paper systematically develops the tokenomics framework and demonstrates its practical applicability to the research testbed ecosystem.

## 2. Background and Definitions

In order to design a robust tokenomics framework, we first explain the core concepts and terminology that form our DAO model. We focus on DAOs, their governance processes, the Algorand platform’s smart contract capabilities, and the role of ASA.

### 2.1. What is a DAO?

A Decentralized Autonomous Organization (DAO) encodes its rules in on-chain smart contracts, automating proposals, fund management, and membership without centralized boards [1].

### 2.2. Governance

Governance in a DAO typically follows three stages [2]:

- 1) **Proposal Creation (Off-Chain & On-Chain):** Members come up with and discuss proposals off-chain (e. g. forums or meetings), A finalized proposal is then submitted on-chain, registering its details in the governance contract.
- 2) **Voting:** Token holders lock (or stake) their GTs to cast votes for or against the proposal within

a predefined period. Vote weight can be proportional to holdings or adjusted by other rules.

- 3) **Execution:** After voting closes, if the quorum and approval thresholds are met, a smart contract automatically carries out the decision implementation, such as minting new tokens, transferring assets, or updating the system's parameters.

Such a hybrid off-/on-chain model balances the organization's low-cost coordination with transparency.

### 2.3. Algorand and Smart Contracts

Algorand is a public blockchain that uses a pure proof-of-stake (PPoS) consensus mechanism to achieve fast finality and low transaction fees. Its design prioritizes scalability and energy efficiency, making it suitable for high-frequency micro-transactions such as resource payments in our scenario.

Algorand supports two types of smart contracts [3]:

- **Stateless Contracts:** Lightweight scripts that approve single transactions based on static conditions. They cannot store data between calls.
- **Stateful Contracts:** Programs with an on-chain key-value store, which allows to retain state across multiple transactions, what is essential for tracking token-pool balances, proposal counts, and vote tallies over time.

Our DAO uses *stateful contracts* because they “re-member” pool reserves and governance state, enabling multi-step workflows like minting, reclaiming, and member admission without using any external databases.

### 2.4. Algorand Standard Assets (ASA)

ASAs provide native support for issuing and managing fungible tokens without a contract code [4]. ASAs are defined by parameters including total supply, decimal precision, and designated addresses with minting, freezing, or management privileges. ASA tokens benefit from Algorand's high throughput and low transaction fees, making them suitable for frequent transactions (resource payments) and governance actions (proposals). Our system will create two ASA instances: one for PT and one for GT — leveraging Algorand's built-in support for transparent token management.

## 3. Tokenomics

The proposed DAO's design depends on two tokens: *Payment Tokens* (PT), operating as “cash” for resources, and *Governance Tokens* (GT), issued monthly in a “pre-paid” way to reflect recent contributions. Such an approach enables payments while keeping voting power aligned with participation.

### 3.1. Payment Tokens (Cash Model)

PTs are the DAO's on-chain cash for resource bookings. At Genesis, the DAO mints a fixed PT supply of (illustratively) 1,000,000 tokens. Founding members (five testbeds) receive 50% of that supply (500,000 PT), split

equally at 100,000 PT each to cover initial operations and establish baseline liquidity. The remaining 50% (500,000 PT) is held in the PT Pool, from which newly admitted testbeds receive a one-time allocation (determined by governance) when they join.

Under this model, whenever Testbed A books, say, 10 GPU hours or 10 6G time slots (defined by the rules) on Testbed B, A transfers 1,000 PT directly from its wallet to B's wallet. This immediate, wallet-to-wallet payment ensures providers are compensated in real-time. If a testbed exhausts its PT balance, it may either earn tokens by providing resources or, if necessary, subject to a governance vote and receive a capped one-time top-up from the PT Pool (e. g. up to 1,000 PT per defined cycle).

By granting founding members half the supply and reserving the rest for governed new-member allocations and emergency top-ups, the cash model provides both initial stability and the flexibility to onboard additional testbeds in a controlled, democratic fashion. The controlled top-up mechanism pushes testbeds to manage their PT budgets carefully and use resources wisely, preventing frivolous or simply wasteful bookings.

### 3.2. Governance Tokens (Prepaid Model)

GTs encode decision-making power within the DAO and are never used for resource payments. At launch, the DAO mints a total supply of 100,000 GT. Founding members (five testbeds) receive 50% of that supply (50,000 GT), split equally 10,000 GT each to establish initial governance rights. The remaining 50% (50,000 GT) is held in the GT Pool, reserved for periodic, contribution-based issuance to all active testbeds.

Rather than a one-off grant, GT are distributed (e. g. monthly) according to each testbed's contribution: at the start of each cycle, the DAO measures the PT each testbed earned during the previous month (e. g. A earned 20,000 PT and B earned 5,000 PT). To maintain a 1 GT : 10 PT ratio reflecting the GT Pool's ten times smaller size relative to PT, the contract issues 1 GT for every 10 PT earned. Thus, the smart contract transfers GT from the pool: A receives 2,000 GT, B receives 500 GT. Any GT that remain unspent by the end of the cycle are returned to GT pool, preventing accumulation and ensuring that voting power always reflects recent, active participation.

Since newcomers have no prior earnings, the DAO grants each newly admitted testbed a minimal allocation, 1,000 GT for example, ensuring they can immediately participate in governance. This welcome grant comes from the GT Pool and helps integrate new members without skewing voting power excessively.

This prepaid issuance model aligns governance influence with actual contributions and discourages passive or long-inactive members from hoarding voting power. By resetting GT balances every cycle and returning unused tokens, the DAO maintains a dynamic, merit-based governance structure in which decision-making rights are based on each testbed's real-time engagement.

### 3.3. Operational Workflows

This subsection describes the DAO's two routines that drive token distribution and resource coordination. First,

we detail a hypothetical New-Member Admission Flow, which governs how candidates join and receive their initial token allocations. Then, we describe the ongoing monthly token flow, showing how PT earnings translate into GT issuance, resource bookings are made, and how unused governance tokens return to the pool each cycle.

**3.3.1. New-Member Admission Flow.** Membership starts off-chain: candidates announce their hardware capabilities on the DAO forum. A sponsoring member then submits the candidate's details, such as address and capacity, to the on-chain governance contract, opening a forty-eight-hour vote. Token holders stake their cycle-issued GT to vote. If a simple majority supports admission, the contract atomically transfers the one-time PT allocation (e. g. 50,000 PT) and a welcome GT grant (e. g. 1,000 GT) to the newcomer's wallet, as shown in Fig. 1. Finally, the registry marks the testbed as active, enabling it to both consume and provide resources in the next cycle [5].

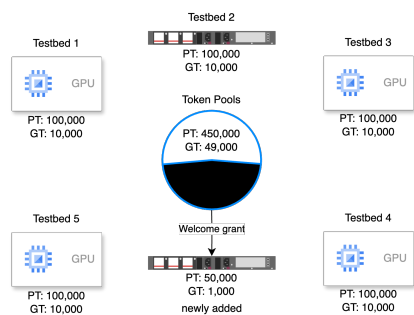


Figure 1: New-Member Admission, welcome grant.

**3.3.2. Ongoing Monthly Token Flow.** On the first day of each cycle, the contract tallies each testbed's PT earnings from the previous month and issues GTs at a 1 GT : 10 PT ratio from the GT Pool. Resource bookings occur via instant PT transfers between wallets as the month proceeds. Governance proposals may arise anytime: members stake GTs to vote, and once voting closes, the contract enacts the decision and returns staked GT. At the cycle's end, any unspent GTs return to the pool. This loop (PT earnings → GT issuance → PT resource bookings → GT return) ensures that token balances remain tightly coupled to actual contributions and usage while maintaining the integrity of both token pools.

### 3.4. Corner-Case Protections

Even the most robust tokenomics framework must prepare for exceptional situations. Below, we expand our original principles with complementary safeguard methods for PT/GT pools, treasury reclamation, emergency needs, and "rage quits" to ensure the DAO remains operational and equitable.

**3.4.1. PT Exhaustion and Emergency Top-Ups.** Because Payment Tokens are granted once upfront, heavy

consumers may run their wallets to zero before earning new tokens. Our DAO offers a capped emergency top-up mechanism to prevent deadlock where a testbed cannot book critical resources. Any testbed in need may submit a governance proposal, requiring only a simple majority of staked GT, to unlock a one-time PT grant (e. g. limited up to 1,000 PT per cycle) from the PT Pool. This advance must be repaid through subsequent resource earnings. By tying such top-up to governance approval, we ensure that requests are reasonable and encourage careful PT usage. Beyond liquidity, contract-level security risks must be considered. Attacks such as double-reporting of resource usage, Sybil identities, or malicious proposals could undermine fairness. These risks suggest the need for trusted measurement oracles and rate-limiting mechanisms in future prototypes.

**3.4.2. PT Pool Depletion.** If repeated top-ups exhaust the PT Pool, the DAO faces a shortage of reserve tokens for new member allocations and emergencies. Two recovery paths are available:

- **Controlled Minting:** The DAO can pass a vote to mint additional PTs at a predetermined percentage of the original supply (e. g. 5%) to replenish the pool, balancing liquidity needs with token scarcity.
- **Inactive Balance Reclamation:** If a testbed remains inactive (no PT earnings or bookings) for two consecutive cycles, its leftover PT balance can be returned to the PT Pool via a routine governance action, reclaiming tokens without affecting total supply.

These methods ensure that reserve tokens remain available for both new members and emergencies.

**3.4.3. GT Pool Depletion.** A similar risk arises if continuous GT issuance to existing and new members drains the Governance Token Pool. Once depleted, no further GTs can be allocated, meaning governance for newcomers is frozen, and active contributors are disabled. To guard against this, two ways are possible:

- **Mint Voting:** Voting approves the creation of new GTs, similarly to PT minting, to refill the pool for upcoming cycles.
- **Inactivity Confiscation:** Members who neither earn PT nor participate in votes for three consecutive cycles lose their remaining GTs, which return to the pool. Thus, active participants are supported without increasing supply.

**3.4.4. Imbalanced Resource Usage.** A handful of very large testbeds might monopolize resource consumption and earn disproportionate GT allotments, threatening to tilt governance in their favor [6]. To maintain equitable influence, the DAO can cap monthly GT allocations per member (e. g. no more than 20% of total GTs issuable per cycle). In such a way, the system is safeguarded against "plutocracy".

**3.4.5. Departing Members.** As mentioned in [7], the right to exit should be part of the core protocol so that any member can exit at any time and for any reason.

When a testbed voluntarily exits the DAO, it should return its PT and GT holdings to the pools rather than retain them. Upon an on-chain exit call, the governance contract returns any unused PTs/GTs to the respective pools rather than burning them, thereby preserving supply for future cycles. The system will have no orphaned tokens and maintains pool integrity.

#### 4. Open Questions and Future Work

While our dual-token framework lays the groundwork for coordinating resource sharing, some parameters merit further study. For example, the emergency top-up cap must balance availability and token scarcity: too low, and testbeds lack critical access; too high, and PT loses value. Simulations will be crucial to pinpoint an effective threshold.

The cadence of GT issuance also deserves scrutiny. We currently use monthly cycles tied to PT earnings, but shorter intervals (e. g. weekly) might align voting power more closely with activity, whereas longer intervals could cut overhead. Likewise, the newcomer GT grant must strike a balance — large enough to include new members yet small enough to avoid temporary imbalances. Allowing the community to vote on these parameters offers flexibility.

Our framework would benefit from empirical validation. Even simple simulations of token flows could show whether pools deplete too quickly or whether voting power stabilizes over time. Future work should include cost comparisons against existing resource-sharing federations (e. g. GENI, PlanetLab) to quantify advantages in fairness and overhead.

Finally, our pool-replenishment strategies, like minting via majority vote or reclaiming inactive balances, have to be tested in real environment. We must determine when to mint additional tokens rather than reclaiming them based on pool levels and member activity. Adding adaptive logic that monitors these metrics might result in a more efficient system.

#### 5. Conclusion

We have presented an architecture for a research testbed DAO: PTs enable instant, peer-to-peer transfers for resource usage, and GTs are distributed cyclewise in proportion to contributions, ensuring decision rights align with activity. Algorand-implemented solution via stateful smart contracts and ASA enables streamlined membership onboarding, token issuance, resource exchanges, and token "recycling" in a standalone system.

By incorporating emergency votes, controlled minting, and user exits, we are establishing the DAO principles and providing an efficient structure for a distributed testbed ecosystem. At the same time, several trade-offs remain to be explored, such as balancing emergency liquidity against token scarcity, calibrating cycle lengths for governance, and ensuring that heterogeneous resources are priced in a fair and flexible way.

Our contribution should therefore be seen less as a final protocol and more as a blueprint for experimentation. A logical next step will be to implement a minimal prototype and run simulations of token flows under different

workloads. Pilot deployments across a small federation of research sites would further validate scalability and robustness in real-world settings. Such evaluations will ultimately determine how well the proposed dual-token system generalizes and how it can be integrated with broader DAO frameworks.

#### References

- [1] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F.-Y. Wang, "Decentralized autonomous organizations: Concept, model, and applications," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 870–878, Oct. 2019.
- [2] K. Pakseresht and H. Kinkel, "Governance of a distributed autonomous organization," Chair of Network Architectures and Services, Technical University of Munich, Tech. Rep., 2024, seminar paper.
- [3] J. Weathersby, "Linking algorand stateful and stateless smart contracts," <https://developer.algorand.org/articles/linking-algorand-stateful-and-stateless-smart-contracts>, Nov. 2020, Algorand Developer Portal.
- [4] "Algorand Standard Assets (ASAs)," <https://developer.algorand.org/docs/get-details/asa/>, [Online; accessed 22-June-2025].
- [5] Upstream, "Joining a dao," <https://guide.upstreamapp.com/dao-members/joining-a-dao>, 2024, [Online; accessed: 19-June-2025].
- [6] M. Esposito, T. Tse, and D. Goh, "Decentralizing governance: exploring the dynamics and challenges of digital commons and daos," *Frontiers in Blockchain*, vol. 8, p. 1538227, 2025, published: 16 May 2025.
- [7] Moloch, "Ragequit," <https://moloch.daohaus.fun/features/ragequit>, 2025, [Online; accessed: 19-June-2025].