Decentralized Autonomous Organization (DAO) Fund Management: How to Allocate and Govern Funds

Conviction Voting

Conviction voting is a governance mechanism that allows token holders to amplify their voting power by locking their tokens for longer periods. It balances stake (how many tokens a person owns) with time commitment (how long they are willing to lock them), creating a system that rewards strong, long-term conviction rather than short-term speculation.

🧠 How Conviction Voting Works

In traditional 1-token-1-vote systems:

- The weight of your vote = the number of tokens you own.
- There's no time element — you can vote and instantly transfer your tokens elsewhere.

In **conviction voting**, however:

- You can **choose a "conviction level"**, which multiplies your vote strength.
- The higher your conviction, the **longer your tokens are locked** after the vote.

So, if you vote with **5x conviction**, your vote counts **5 times stronger**. —

✅ Advantages

1. Rewards Long-Term Commitment

Conviction voting ensures that those willing to **lock tokens** (showing real belief in the proposal) get more voting power. It aligns influence with **skin in the game**.

2. Discourages Short-Term Speculation

Traders who hold tokens for short-term profit won't want to lock their tokens, reducing governance manipulation by whales who don't care about network health.

3. Enhances Governance Stability

Since tokens stay locked after voting, it prevents rapid governance swings and promotes decisions that consider **long-term effects**.

4. Encourages Participation Diversity

Users can choose their own balance between **liquidity and influence** — you don't need to stake huge amounts to matter, just conviction.

—

❌ Disadvantages

1. Complexity for Users

Conviction voting is **hard to understand** for newcomers — especially choosing conviction levels, lock times, and understanding enactment periods.

2. Liquidity Trade-Off

High conviction locks funds for long durations. This reduces **liquidity** and can discourage smaller holders from engaging in governance.

3. Wealth Bias Still Exists
Even though conviction helps smaller holders, **whales** can still dominate governance by voting with both **high tokens** and **high conviction**.

4. Voter Fatigue
Since governance is continuous, long-term token locking might reduce enthusiasm for frequent participation.

—

🧭 How Conviction Voting Works in SymbioSky

🌱 The Simplest Version

Voters give a score between **1 and 10** on a proposal. After all votes are submitted, the **mean (average)** score is calculated.

If the **mean score** is **6** for a handle or proposal, then than handle receives **600 tokens** as an incentive. These incentives are distributed **every three months**.

Question: Should voters be rewarded with incentives just for participating in voting?

—

⚙️ Adding More Complexity

In this version, voters still give a score between **1 and 10**. The **mean score** is calculated, and handle receive **non-tradable reputation tokens** instead of regular tokens.

For example, if the mean is **6**, a handle earns **600 reputation tokens**. These tokens **cannot be traded or sold** but can be **used for conviction voting** for incentive distributions.

This design rewards honest and aligned voting with **governance power**, not just money.

One common criticism is that people are not always rational and do not make fully informed or optimal decisions, especially in specialized fields. Additionally, many individuals lack the necessary expertise or specialization to make sound judgments in these areas. To address this, one potential solution is to introduce *reputation scores* for academics and researchers. These scores could reflect their expertise, track their contributions, and assess their credibility. By relying on individuals with higher reputation scores to make decisions on behalf of others, we can enhance the quality and reliability of decisions in specialized domains.

—

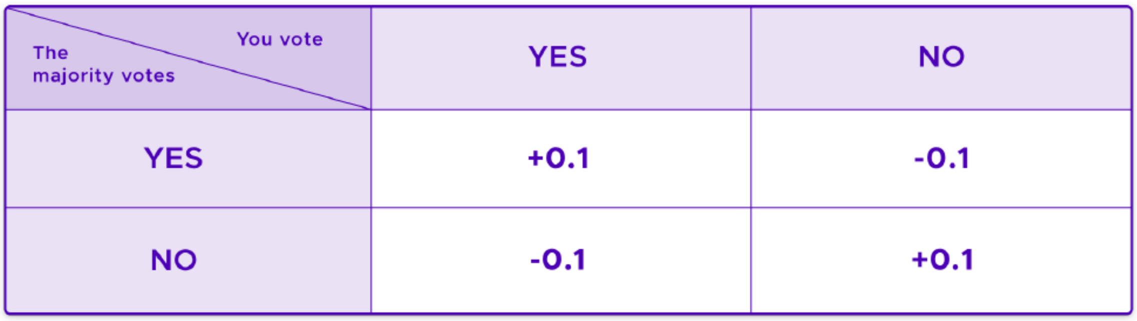
🔒 Adding further Complexity With Commit–Reveal Mechanism

To make voting fair and tamper-proof, **conviction voting in SymbioSky** uses a **commit–reveal scheme**:

1. **Commit Phase:** Voters **submit a hashed version** of their vote (keeping it secret). No one can see how others voted during this phase.
2. **Reveal Phase:** After the commit period ends, voters **reveal** their votes. The system verifies their vote matches the earlier hash.
3. **Reward Phase:** The **mean score** is computed.
   - Voters **close to the mean** earn more tokens.
   - Voters **far from the mean** lose some tokens.

4. **Randomization**

   Randomization **reduces bias** in decision making. No individual or faction can keep influence permanently because selection is unpredictable. Randomness makes it difficult for interest groups to capture the decision-making process. Instead of requiring everyone to participate directly, the system can use **random selection of jurors**, where individuals with higher conviction weight have a greater chance of being chosen, while still remaining resistant to Sybil attacks.



You vote / The majority votes	YES	NO
YES	+0.1	-0.1
NO	-0.1	+0.1

Figure 1: Payoff table for a basic Schelling game

💡 Why It’s Called a “Schelling” Mechanism

Economist Thomas Schelling introduced the idea of focal points — outcomes that people naturally converge on because they expect others to do the same.

In the SchellingCoin system:

The “focal point” is truth.

Each person expects others to report the truth.

Since everyone thinks this way, being honest becomes the best strategy.

This creates a self-reinforcing honesty loop:

```
I expect you to be honest →
You expect me to be honest →
So both of us are honest →
And truth wins collectively.
```

Too Much UI, Too Little Engagement

A more complex or multi-step user interface (UI) can significantly hinder user experience by increasing the cognitive load on users. When users are required to navigate through multiple steps or screens to complete a task, it can lead to frustration, confusion, and a higher likelihood of abandonment. Complex UIs often require users to remember multiple steps, which can be especially challenging for first-time users or those who are not tech-savvy. This can result in a poor user experience, ultimately leading to lower engagement and satisfaction with the product or service.

The Cost of Complexity: Extended Development Cycles and Missed Opportunities

Increased complexity in software development leads to longer development cycles, which in turn delays the time to market. When a project becomes more complex, it often requires more detailed planning, more resources, and more rigorous testing. These factors all contribute to extended development timelines, making it harder to deliver a product quickly. As a result, the ability to create value for users or customers is postponed, which can be detrimental in fast-moving markets where early adoption and competitive advantage are crucial.

Moreover, added complexity can introduce more potential points of failure, requiring additional time for debugging and quality assurance. This not only extends the development phase but also increases the risk of project overruns and budget constraints. The longer a project takes to develop, the more likely it is that market conditions, user needs, or business priorities may have changed, further reducing the product's relevance or value upon launch.

- **"We don't even know how the simplified version will work. Should we test the simplified version first, and only add extra complexity if needed by creating another future app?"**