

Governance

Overview

This document outlines the governance process of the Seamless protocol, it is subject to change by the governance process itself. On-chain governance for Seamless protocol is on Base.

Process

0. Discuss proposal with the community in Discourse.
1. Proposal is submitted on Snapshot labs for off-chain temperature check.
2. Proposal is submitted on chain (including the code to be executed on chain by governor and timelock).
 - Voting begins after a 2 day delay
3. Voting.
 - Snapshot of voting power occurs at the beginning of the voting period. Voting power (including delegation) for this proposal may not change after this snapshot.
4. Timelock executes proposal code on chain.

Governance Thresholds

- **Protocol changes**
 - **Quorum:** 4% of token supply (For, No and Abstain votes count toward quorum)
 - **Vote differential:** Number of For votes > Number of Against votes
 - **Timelock:** 2 days (48 hours)
 - **Proposal voting power requirement:** 0.5% of token supply
 - **Voting period:** 3 days
- **Governance changes**
 - **Quorum:** 4% of token supply (For, No and Abstain votes count toward quorum)
 - **Vote differential:** Number of For votes > 2 x Number of Against votes
 - **Timelock:** 5 days (120 hours)
 - **Proposal voting power requirement:** 0.5% of token supply
 - **Voting period:** 10 days
 - * Voting period may be extending if a quorum is reached late in the voting period, this is to allow time for additional votes to come in if quorum is reach unexpectedly near the end of a voting period

Voting Power

- SEAM is an ERC20 token that entitles a user to participate in on-chain governance. SEAM balance is voting power.

- Implementation notes:
 - * ERC20 SEAM token. Governor and delegation work out of box
 - * ERC20 veSEAM token. Accepts SEAM tokens and returns a veSEAM (balanceOf to return voting power). Snapshot the SEAM amount and lock end time, based on this the vote power can be calculated at any previous snapshot for delegation and governor vote count
 - * Governor contract overrides `__getVotes` to aggregate voting power from both tokens
 - * Note: you could hypothetically delegate SEAM tokens to one delegate and veSEAM to another.
- Users can lock SEAM governance tokens to obtain veSEAM for significantly increased voting power. The balance of veSEAM is an account's voting power. veSEAM is non-transferable
 - Voting power decays linearly, i.e.: voting power is at its highest at the beginning of a lockup and 0 at the end of a lockup
 - Tokens can be locked for any length from 6 months to 4 years. **Voting power = balance of SEAM locked * (-1 * slope * seconds remaining in lock period + voting power at lock start)**
 - Voting power does not decay until user starts unlock cooldown.
- Voting power must be delegated to be used, either to self or to another address (EOA or smart contract).
 - Voting power can only be delegated to a single address at once.
- Voting power for a specific proposal is determined by a snapshot of voting power state on chain when the proposal voting period begins.
- Voting power and proposal power must be maintained until execution of a proposal on chain. // TODO(wes): can we enforce voting power of each voter and delegate before execution easily? Can we enforce proposal power at least?

SEAM Gov Token

SEAM is only earned by using the protocol and performing behaviours that push the community and protocol forward. i.e.: emissions or airdrop determined by governance.

Emissions

| Program | Amount | Description |
|----------------|--|-------------|
| veSEAM holders | Min: 5% (Min per pool: $x / (\# \text{ pools}) * x$, $x = \text{min}$). Max: 40%. Amount chosen by governance. | |

| Program | Amount | Description |
|-------------------------|--|---|
| DEX liquidity providers | Min: 7.5% (Min per pool: $x / (\# \text{ pools}) * x$, $x = \text{min}$). Max: 40%. Each DEX pool amount chosen by governance. This is the rewards program that should have the highest rewards emissions. | Yes, locked only. Users that timelock their DEX LP tokens (allowlist determined by governance). Protocol to provide token locker contracts. |
| sToken LPs | Min: 5% (Min per pool: $x / (\# \text{ pools}) * x$, $x = \text{min}$). Max: 40%. Each market chosen by governance. | Yes, locked and not locked (locked should receive higher emissions). Locked cannot be used as collateral (protocol to provide locker contracts). Unlocked rewards are emitted directly through Aave rewards platform. |
| ILM LPs | Min: 5% (Min per pool: $x / (\# \text{ pools}) * x$, $x = \text{min}$). Max: 40%. Each ILM chosen by governance. | Yes locked only. Protocol to provide locker contracts. |
| debt tokens | Min: 7.5% (Min per pool: $x / (\# \text{ pools}) * x$, $x = \text{min}$). Max: 40%. Each debt market chosen by governance. | Yes. No locking required. |

// TODO(wes/daryl): Should we set some emissions guardrails (i.e.: maximums for each program maybe minimum too)? Requires a constitutional change to override these limits. Daryl: we should maybe have a 1 year timelock on emissions changes, i.e.: emissions guardrails cannot be changed in the first year.

// TODO(wes): Should emissions programs be in weeks? Should emissions continue perpetually until changed (i.e.: no reloading of contracts it's just a perpetual stream)?

Emissions will continue at predefined rate until governance changes it. We should define the rate of emissions at the top of funnel.

The intention is to have governance choose the rewards emission rates for each program and each pool within a program. These may change over time as

governance allocates based on current needs. Governance should decide the total SEAM rewards rate (per second), start, and end time for each program.

Participants in a program will receive a share of the total rewards allocated to the program per second based on the size of their contribution (e.g.: amount of sTokens). All emissions are emitted in units of SEAM but are automatically locked for 1 year as veSEAM.

Delegation

Each unit of voting power can be delegated to any address, including smart contracts. A smart contract may implement its own governance rules for use of the voting power delegated to it. One such governance process may be the execution of an off chain governance process.

It is in the best interest of protocol governance that users do not centralize delegation to small number of governance delegates. Concentration/centralization of voting power poses systemic risks to protocol governance. As such each delegate has a max voting power they can receive.

Option: Beyond this max the voting power of a delegate is reduced quadratically (i.e.: $\text{votingpower} = \min(\text{voting_power_threshold}, \text{voting_power_delegated}) + \sqrt{\text{voting_power_threshold} - \text{voting_power_delegated}}$). *This could be abused by creating multiple addresses, without Sybil resistance this may be pointless. Or maybe it creates enough friction and awareness for users that they would be less likely to delegate to a duplicate delegate (for example when looking up delegate profiles and seeing a duplicate).*

Governance Actions

1. Propose and vote on changes to the protocol.
2. Propose and vote on changes to governance itself.
3. Propose and vote on SEAM token emissions.

Emergency procedures

To prevent negative impacts to the protocol and its users, governance appoints guardians, which have the authority to make scoped emergency changes to the protocol that are too time sensitive to run through the governance process. The actions that can be taken by guardians are pre-determined by governance and especially are not all encompassing. The role of guardians is to prevent damage to the protocol and its users until the governance process can implement more significant changes to resolve issues. For example, a guardian may be granted permission to pause protocol contract activity if a security issue or major bug is discovered, this pause would last until governance can approve and deploy a remedy.

Guardians are smart contracts wallets (e.g.: multisig) that are configured by governance. For example, governance would appoint members of the community to act as signers on a guardian multisig.

Implementation

On chain governance contracts will be based on the Openzeppelin v5 governance contracts ref.