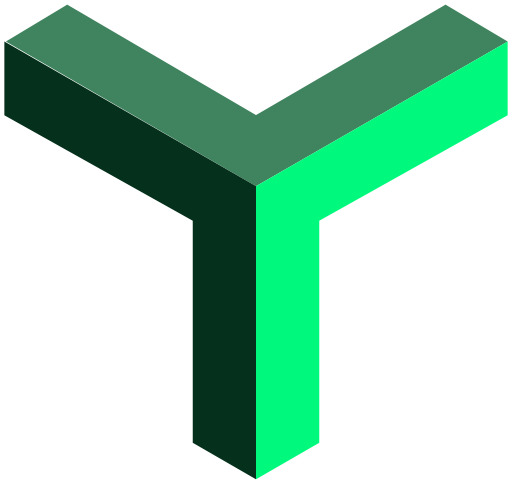
Root: Fungibility for Beanstalk Silo Deposits



Kokonut, Mistermanifold, Sarrdinero and Publius
roottoken@protonmail.com
roottoken.org

Published: November 16, 2022
Modified: November 16, 2022
Whitepaper Version: 1.0.0
Code Version: 1.0.0[1]

"Out of intense complexities, intense simplicities emerge."

- Winston Churchill, The World Crisis

Abstract

Beanstalk[2] *Silo Deposits*[3] are not fungible. A fungible wrapper for *Silo Deposits* can create additional utility for Beans, particularly before a standard interface for other protocols to interact with *Silo Deposits* exists. Root is a fungible wrapper for Beanstalk *Silo Deposits* that implements the ERC-20 Standard. Any Ethereum account can *Mint* and *Redeem* Roots (🌱), the Root ERC-20 Standard token, for *Silo Deposits*. A decentralized autonomous organization (DAO) governed by 🌱 owners facilitates the coordination of Root upgrades and collective participation in Beanstalk governance.

[1] github.com/RootToken/Root
[2] bean.money/docs/beanstalk.pdf
[3] Any italicized terms not defined herein are defined by Beanstalk.

Table of Contents

1 Introduction

Beanstalk is an Ethereum-native permissionless fiat stablecoin protocol. Beanstalk's credit based stability model is a potential solution to the stablecoin carrying cost problem plaguing decentralized finance (DeFi). Whereas other stablecoin protocols rely on collateral to issue stablecoins, which creates centralization and carrying costs, Beanstalk uses infinitely scalable decentralized credit to create a permissionless stablecoin, Bean (Ⓑ), that has carrying costs competitive with off-chain fiat.

While Bean is an ERC-20 Standard token, in order to receive Beanstalk-native passive interest, Beans must be *Deposited* in the *Silo*, the Beanstalk DAO, directly or wrapped in whitelisted[4] liquidity pool (LP) tokens. Beanstalk evaluates the flash-loan-resistant Bean-denominated-value (BDV) of each *Deposit*, at the time of *Deposit*. However, *Silo Deposits* have two qualities that make them non-fungible: the *Stalk* and *Seeds* per BDV of each *Deposit*. The non-fungibility of *Silo Deposits* appears necessary to Beanstalk's peg maintenance mechanism, but currently comes at the cost of composability.

Composability is one of the core value propositions of blockchains. In order to facilitate composability, there are a variety of Ethereum token Standards that are widely adopted by various DeFi protocols. The ERC-20 Standard is the benchmark fungible token standard of the Ethereum network. Root is an Ethereum-native permissionless wrapper that implements the ERC-20 token Standard to create fungibility and composability for Beanstalk *Silo Deposits*.

2 Previous Work

Root uses the OpenZeppelin[5] implementation of the ERC-20 Standard.

EIP-4626 formalized a Standard for token vaults, but only supports a single underlying ERC-20 Standard token.

Root is implemented on top of Beanstalk.

3 Root

Root is a permissionless fungible wrapper that enables collective farming for Beanstalk *Silo Deposits*.

Any Ethereum account can *Mint* and *Redeem* 🌱 for Beanstalk *Silo Deposits* on the *Minting Whitelist* through Root at anytime. Anytime 🌱 are *Minted* or *Redeemed*, the BDV, *Stalk* and *Seeds* per 🌱 either remain the same or increase.

All value in Root is owned pro rata by 🌱. Any account can contribute to collective farming by calling the `earn`, `mow`, `updateBDV` and `updateBDVs` functions.

3.1 Minting Whitelist

Any ERC-20 Standard token that is on the Beanstalk *Deposit Whitelist* can be added to and removed from the *Minting Whitelist* via Root governance.

Any Beanstalk *Silo Deposit* of a token on the *Minting Whitelist* can be used to *Mint* 🌱.

[4] bean.money/docs/beanstalk.pdf#subsection.14.2
[5] github.com/OpenZeppelin/openzeppelin-contracts-upgradeable

3.2 Mint

Any account can *Mint* Υ at anytime by calling the `mint` function with (1) the minimum Υ to *Mint* (Υ^{min}), such that $\Upsilon^{\text{min}} \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, and (2) a list of Beanstalk *Silo Deposits* of tokens on the *Minting Whitelist* not currently owned by Root. The Υ received upon *Minting* (Υ^{+}_{∂}), such that $\Upsilon^{+}_{\partial} \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, is the maximum of (1) Υ^{min} and (2) the minimum of the percentage change in the BDV, *Stalk* and *Seeds* of Root as a result of the *Mint*.

Therefore, we define Υ^{+}_{∂} for a given Υ^{min} and list of *Silo Deposits* with cumulative BDV (Σl), cumulative *Stalk* (Σk) and cumulative *Seeds* (Σc), and Root total BDV before the *Mint* ($L_{<\partial}$), total *Stalk* before the *Mint* ($K_{<\partial}$), such that $\Sigma k,\ K_{<\partial} \in \{j \times 10^{-10} \mid j \in \mathbb{N}\}$, and total *Seeds* before the *Mint* ($C_{<\partial}$), such that $\Sigma l,\ \Sigma c,\ L_{<\partial},\ C_{<\partial} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as:

$$\Upsilon^{+}_{\partial} = \max\left(\Upsilon^{\text{min}},\ \min\left(\frac{\Sigma l}{L_{<\partial}},\ \frac{\Sigma k}{K_{<\partial}},\ \frac{\Sigma c}{C_{<\partial}}\right)\right)$$

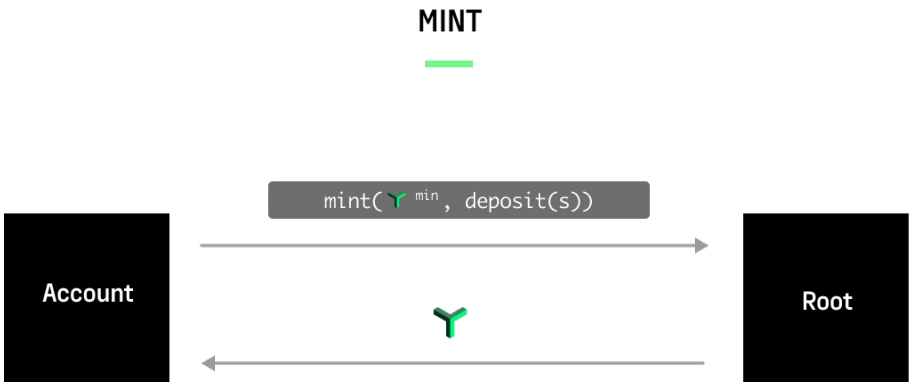


Figure 1: Mint

3.3 Redeem

Any account that owns Υ can *Redeem* them at anytime by calling the `redeem` function with (1) the maximum Υ to *Redeem* (Υ^{max}), such that $\Upsilon^{\text{max}} \in \{j \times 10^{-10} \mid j \in \mathbb{N},\ j \leq 10^6\}$, and (2) a list of Beanstalk *Silo Deposits* currently owned by Root. The Υ necessary to *Redeem* a list of *Silo Deposits* (Υ^{-}_{∂}), such that $\Upsilon^{-}_{\partial} \in \{j \times 10^{-18} \mid j \in \mathbb{Z}\}$, is the minimum of (1) Υ^{max} and (2) the maximum of the percentage change in the BDV, *Stalk* and *Seeds* of Root as a result of the *Redemption*.

Therefore, we define Υ^{-}_{∂} for a given Υ^{max} and list of *Silo Deposits* with Σl, Σk and Σc, and Root total BDV after the *Redemption* ($L_{\partial>}$), total *Stalk* after the *Redemption* ($K_{\partial>}$), such that $K_{\partial>} \in \{j \times 10^{-10} \mid j \in \mathbb{N}\}$, and total *Seeds* after the *Redemption* ($C_{\partial>}$), such that $L_{\partial>}\, C_{\partial>} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as:

$$\Upsilon^{-}_{\partial} = \min\left(\Upsilon^{\text{max}}, \max\left(\frac{\Sigma l}{L_{\partial>}}, \frac{\Sigma k}{K_{\partial>}}, \frac{\Sigma c}{C_{\partial>}}\right)\right)$$

Figure 2: Redeem

3.4 Earn

Any account can (1) *Mow* all of Root's *Grown Stalk*, (2) *Plant* the *Seeds* associated with Root's *Earned* ⅊and (3) *Deposit* Root's *Earned* ⅊ in the current *Season*, at anytime by calling the `earn` function. This is the only instance the Stalk or Seed per BDV ratios of Root may decrease.

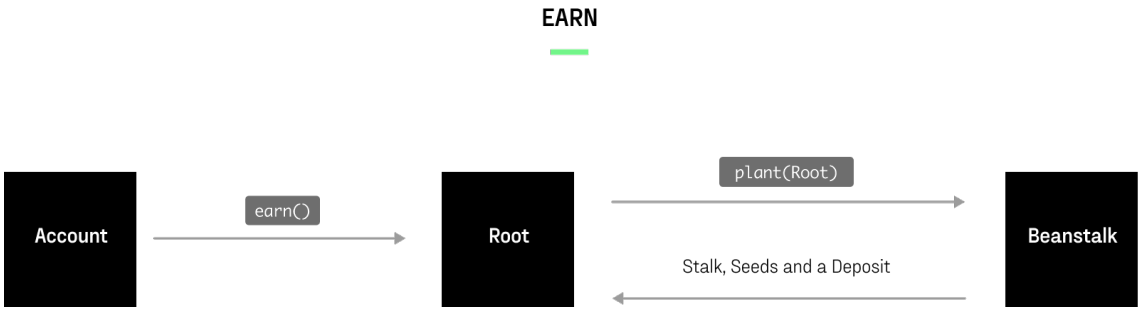


Figure 3: Earn

3.5 Mow

Any account can *Mow* Root's *Grown Stalk* at anytime by calling the `mow` function on behalf of Root directly to Beanstalk.

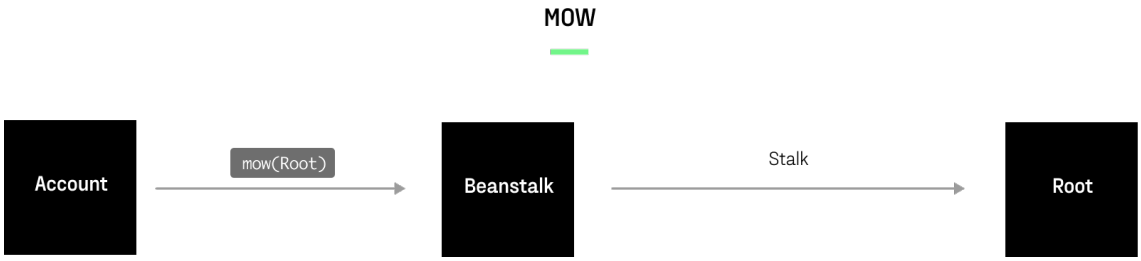


Figure 4: Mow

3.6 Update BDV

Any account can update the BDV of one or multiple of Root's *Silo Deposits* at anytime by calling the `updateBdv` or `updateBdvs` functions, respectively.

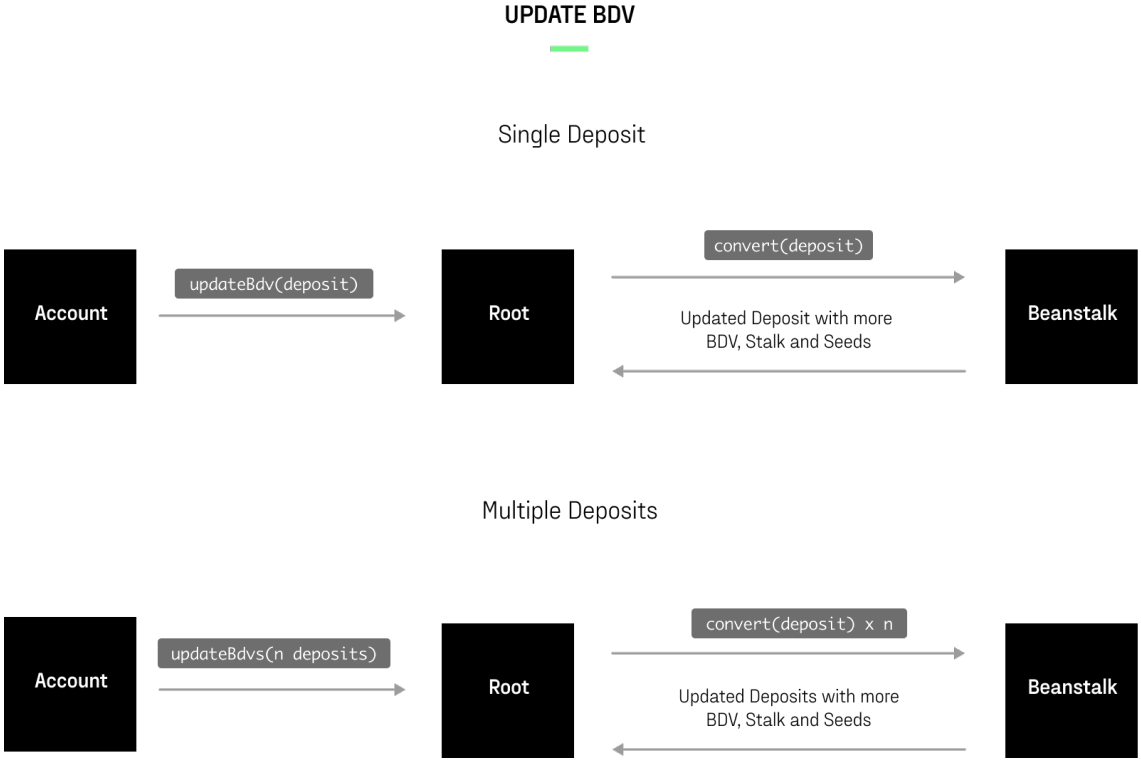


Figure 5: Update BDV

4 Governance

Root is an upgradable contract. Root's *Stalk* ownership gives Root a vote in Beanstalk governance.

In theory, Root governance should balance (1) ensuring sufficient time for all 🌱 owners to consider a *Root Improvement Proposal* (*RIP*) and cast their votes or exit the system, with (2) the abilities to be quickly upgraded in cases of emergency and participate in Beanstalk governance.

In practice, the owner of Root has the exclusive and unilateral ability to commit *RIPs*, and vote in Beanstalk governance, at anytime. The owner of each implementation of Root enforces a governance mechanism.

A RIP has 2 inputs: (1) a new contract address and (2) an optional function call.

5 Economics

🌱 owners share all value earned by Root pro rata across all 🌱. The pro rata distribution creates a zero-fee shared collective farming option for passive Beanstalk *Silo Depositors* that (1) maximizes Beanstalk-native yield and (2) generates additional yield without creating significant risks to Beanstalk or Root.

The *Stalk* and *Seeds* owned by Root increase each time an account successfully calls the `earn` function. The *Stalk* owned by Root increases each time an account successfully calls the `mow` function. The BDV, *Stalk* and *Seeds* owned by Root, and *Redeemable* per 🌱, increase each time an account successfully calls the `updateBdv` or `updateBdvs` functions.

Each time an account *Mints* 🌱, unless they do so such that the percentage change in the BDV, *Stalk* and *Seeds* of Root are identical (*i.e.*, $\frac{\Sigma l}{L_{<\partial}} = \frac{\Sigma k}{K_{<\partial}} = \frac{\Sigma c}{C_{<\partial}}$), there is some excess BDV, *Stalk* or *Seeds* that is transferred to Root. Similarly, each time an account *Redeems* 🌱, unless they do so such that the percentage change in the BDV, *Stalk* and *Seeds* of Root are identical (*i.e.*, $\frac{\Sigma l}{L_{\partial>}} = \frac{\Sigma k}{K_{\partial>}} = \frac{\Sigma c}{C_{\partial>}}$), there is some excess BDV, *Stalk* or *Seeds* left in Root.

Conversions of Beanstalk *Silo Deposits* within Root are permitted only to increase the BDV of *Silo Deposits*, which prevents manipulation of Beanstalk and Root through price manipulation of Beans.

All non-Beanstalk yield paid to Root is held by Root.

6 Risk

There are numerous risks associated with Root. This is not an exhaustive list.

The Root code base is novel and has not been tested in the "real world" prior to the initial Root deployment. The open source nature of Root means that others can take advantage of any bugs, flaws or deficiencies in Root and launch identical or very similar Beanstalk fungible *Silo Deposit* wrapper token implementations.

Root is deployed on the Ethereum network. The security of the Ethereum network is assumed.

Root is dependent on Beanstalk, and therefore inherits all of the risks associated with Beanstalk. The security of Beanstalk is assumed. For an exhaustive list, consult the Beanstalk whitepaper[6] and Beanstalk DAO Disclosures.[7]

The potential friction to *Mint* 🌱 due to the *Stalk* or *Seed* per 🌱 ratios can make doing so unattractive, thereby limiting supply.

The permissionless and pro rata nature of Root makes it impossible to regulate the ownership distribution of 🌱.

The owner of Root is potentially vulnerable to compromise.

7 Future Work

Root is a work in progress. The following are potential improvements that can be incorporated into Root as one or more *RIPs*, or via one or more forks:

- A gas-efficient method can be implemented to allow 🌱 owners to permissionlessly claim non-Beanstalk-native yield Root earns, pro rata.
- The governance process around *RSPs* can be further refined to maximize Root's ability to participate in non-Root governance.
- A system that facilitates *Converts* within Root, beyond solely increasing the BDV of *Silo Deposits*, without creating the potential for manipulation can be implemented.
- Root should vote only the portion of its *Stalk* corresponding to the percent of 🌱 that vote for a given *BIP* once Beanstalk supports partial votes.
- Root should implement on-chain governance once Beanstalk returns to it.

[6] bean.money/docs/beanstalk.pdf#section.12
[7] arweave.net/g1fij0syFvuxNR83I2VKOe1auNxtjEUwevkihxoRUXg

8 Appendix

8.1 Implementations

The following is an incomplete list of implementations of Root.

8.1.1 0x77700005BEA4DE0A78b956517f099260C2CA9a26

The Root contract address is 0x77700005BEA4DE0A78b956517f099260C2CA9a26 (*i.e.*, $\Root$ = 0x77700005BEA4DE0A78b956517f099260C2CA9a26).

8.1.1.1 Current Parameters

The following are the current parameters of 0x77700005BEA4DE0A78b956517f099260C2CA9a26:

- The owner of 0x77700005BEA4DE0A78b956517f099260C2CA9a26 is 0xb7774ec5031e1d903152E96BbC1601e5D0D83Ca2 (*i.e.*, the *Root DAO Multisig*);
- The address of Beanstalk is 0xC1E088fC1323b20BCBee9bd1B9fC9546db5624C5; and
- $\Root^{\text{min}} = 0.1\%$.

8.1.1.2 Minting Whitelist

The following ERC-20 Standard tokens are on the Beanstalk *Deposit Whitelist* and *Whitelisted* for *Minting* $\Root$:

- 0xBEA0000029AD1c77D3d5D23Ba2D8893dB9d1Efab (*i.e.*, $\emptyset$).

8.1.1.3 Governance

A robust decentralized governance mechanism must balance the principles of decentralization with resistance to attempted protocol changes, both malicious and ignorant, and the ability to quickly adapt to changing information.

0x77700005BEA4DE0A78b956517f099260C2CA9a26 is governed by Root DAO. Root DAO governs upgrades to 0x77700005BEA4DE0A78b956517f099260C2CA9a26, the use of *Stalk* in Beanstalk governance and the treatment of non-Beanstalk-native yield earned by 0x77700005BEA4DE0A78b956517f099260C2CA9a26. Any account that owns $\Root$ can participate in Root DAO governance.

8.1.1.3.1 RIPs

Any account that owns more than $\Root^{\text{RIP}}$, such that $\Root^{\text{RIP}} \in \{j \times 10^{-18} \mid j \in \mathbb{N},\ j \leq 10^{18}\}$, of total outstanding $\Root$ can submit a *RIP* to the Root DAO Snapshot[8] via the *Root DAO Multisig*.

[8] snapshot.org/#/rootsmoney.eth

Root DAO only accepts votes in favor of *RIPs*. An account's vote for a given *RIP* is counted as the minimum of its 🌱 between the beginning and end of the *Voting Period*.

A *Voting Period* begins when a vote for a *RIP* can be submitted to Snapshot and ends at approximately the beginning of the 169th Beanstalk *Season* after it begins, or when the *RIP* is committed with a supermajority.

If at the end of the *Voting Period*:

- Less than or equal to half of the total outstanding eligible 🌱 votes in favor of the *RIP*, it fails; and
- More than half of the total outstanding eligible 🌱 votes in favor of the *RIP*, it passes.

If at any time before the end of the *Voting Period* more than two-thirds of the total outstanding eligible 🌱 votes in favor of the *RIP*, it passes and can be committed to the Ethereum blockchain.

8.1.1.3.2 Stalk Use

In its capacity as a *Stalkholder*, 0x77700005BEA4DE0A78b956517f099260C2CA9a26 is entitled to vote on (1) each *Beanstalk Improvement Proposal* (*BIP*) and (2) other, non-*BIP*, governance processes, and may be entitled to non-Beanstalk-native yields.

Any 🌱 owner can submit a *Root Stalk Proposal* (*RSP*) to the *RSP* Snapshot[9] to determine how Root should use its *Stalk* in each governance process, and distribute yields, if ever. *RSPs* follow the same structure as *RIPs*, except that the length of the *Voting Period* can be reasonably shortened to ensure the will of Root DAO can be reflected in the governance process.

Any *Beanstalk Improvement Proposal* (*BIP*) will automatically qualify to be proposed to Root DAO as an *RSP*.

The *Root DAO Multisig* will participate on each governance process as determined by the *RSP*.

8.1.1.3.3 Root DAO Multisig

The *Root DAO Multisig* is a 4-of-7 Gnosis[10] multisig wallet with anonymous signers consisting of community members and contributors to Root and Beanstalk. The *Root DAO Multisig* will execute the will of Root DAO. The *Root DAO Multisig* will provide sufficient notice of the submission of a *RIP*, its contents and the beginning of its *Voting Period* before submitting a *RIP* to Snapshot, and will repost *BIPs* on the *RSP* Snapshot as soon as possible after they are posted to the Beanstalk DAO Snapshot.[11] In the future, we expect *RIPs* will implement permissionless governance and revoke these abilities from the *Root DAO Multisig*.

Thus, Root creates a robust decentralized governance mechanism.

[9] snapshot.org/#/rootstalkproposals.eth
[10] app.safe.global/eth:0xa9bA2C40b263843C04d344727b954A545c81D043/balances
[11] snapshot.org/#/beanstalkdao.eth

8.2 Glossary

The following conventions are adopted from the Beanstalk whitepaper and used throughout this paper:

- Lower case Latin letters are unique values;
- Upper case Latin letters are totals or rates;
- Subscripts are time, where $\ni$ is the current transaction; and
- Superscripts are modifiers.

The following variables and terms are used throughout this paper:

$\mathbb{B}$ - Bean;

BDV - Bean-denominated-value;

Beanstalk - The issuer of $\mathbb{B}$ and *Silo Deposits*;

Beanstalk Improvement Proposal - A Beanstalk governance proposal;

BIP - A *Beanstalk Improvement Proposal*;

Σc - The cumulative *Seeds* for a given list of *Silo Deposits*;

$C_{<\ni}$ - Root's total *Seeds* before a given *Mint*;

$C_{\ni>}$ - Root's total *Seeds* after a given *Redemption*;

Convert - Turn one or more *Silo Deposits* into another, within the *Silo*;

DAO - Decentralized autonomous organization;

DeFi - Decentralized finance;

Deposit - An asset in the *Silo*;

Deposit Whitelist - The Beanstalk whitelist that permissions which tokens can be *Deposited* into the *Silo*;

Depositors - An account that has *Deposited* assets into the *Silo*;

Earned $\mathbb{B}$ - Beans paid to a *Stalkholder* after the last *Season* the *Stalkholder* called the `plant()` function;

Grown Stalk - *Stalk* that has *Grown* from *Seeds* but not yet *Mown*;

Σk - The cumulative *Stalk* for a given list of *Silo Deposits*;

$K_{<\ni}$ - Root's total *Stalk* before a given *Mint*;

$K_{\ni>}$ - Root's total *Stalk* after a given *Redemption*;

Σl - The cumulative BDV for a given list of *Silo Deposits*;

$L_{<\ni}$ - Root's total BDV before a given *Mint*;

$L_{\ni>}$ - Root's total BDV after a given *Redemption*;

LP - Liquidity pool;

Mint - Turn one or more Beanstalk *Silo Deposits* into 🌱;

Minting Whitelist - The Root whitelist that permissions using Beanstalk *Silo Deposits* to *Mint* Υ;

Mow - Turn *Grown Stalk* into *Stalk*;

Plant - Turn *Plantable Seeds* associated with *Earned* $\not\Phi$ into *Seeds* by *Depositing* the *Earned* $\not\Phi$ in the current *Season*;

Redeem - Turn Υ into one or more Beanstalk *Silo Deposits*;

RIP - A *Root Improvement Proposal*;

Root Improvement Proposal - A Root governance proposal;

Root Stalk Proposal - A Root governance proposal to determine how Root should use its *Stalk* in a given instance unrelated to *RIPs*;

RSP - A *Root Stalk Proposal*;

Root DAO Multisig - The owner of the Root contract;

Υ - Roots; Root's ERC 20 Standard token;

Υ^{max} - The maximum Υ to *Redeem*;

Υ^{min} - The minimum Υ to *Mint*;

Υ^{RIP} - The percentage of Υ ownership necessary to submit a *RIP*;

Season - Beanstalk-native discrete time;

Seed - A Beanstalk-native asset that *Grows* 1×10^{-4} *Stalk* each *Season*;

Silo - The Beanstalk DAO;

Stalk - The Beanstalk-native governance asset;

Stalkholder - An owner of *Stalk*; a Beanstalk DAO member; and

Voting Period - The period of time Υ owners can vote on a *RIP* or *RSP*.