BIP-36: Silo V3

Proposer

Beanstalk Farms

Proposer Wallet: **TBD**

Summary

- Remove the Withdrawal Freeze from the Silo entirely;
- Delay the issuance of newly Earned Beans to 10 blocks after the beginning of a Season;
- Upgrade the accounting system around Seeds per BDV to Stems such that the Seeds per BDV for a whitelisted asset can be changed after whitelisting;
- Change the Seeds per BDV rewards for both Unripe Bean (urBEAN) and Unripe BEAN:3CRV LP (urBEAN3CRV) to 0; and
- Issue Silo Deposits as ERC-1155 tokens.

Links

- BIP-36 GitHub PR
- GitHub Commit Hash: 55813422bc515a5e36469108d4c4f835158fa8fd
- Safe Transaction (**TBD**)

Problem

Withdrawal Freeze Removal

Currently there is a Withdrawal Freeze that lasts from the time of Withdrawal until the end of the Season. This provides an unnecessarily complex UX as Farmers are forced to perform 2 on-chain actions at different times (Withdraw then Claim). For example, this limits the potential for using Deposits throughout DeFi via Pipeline as Deposits cannot atomically be swapped for the underlying. Furthermore, the supply overhang creates an economic inefficiency for Beanstalk and the lockup release at the same time creates an economic disadvantage for small Farmers when exiting the Silo.

Delayed Earned Bean Seigniorage

If the Withdrawal Freeze is removed with no changes to Earned Bean distribution, Beanstalk becomes susceptible to an attack where an account can Deposit whitelisted assets in the Silo, call `gm` and Withdraw the assets in the same transaction, gaining risk-free Bean seigniorage in the process.

Stems Upgrade

The Seeds per BDV reward for a whitelisted asset is static and can only be set upon whitelisting. This makes it such that (1) Seeds per BDV rewards cannot easily be updated via BIP and (2) no system for autonomously updating Seeds per BDV rewards can exist.

Indexing Deposits by Season results in scenarios where Farmers may forfeit Stalk when Converting from a whitelisted asset to another with a lower Seeds per BDV.

Seeds are implemented without decimals, limiting the economic flexibility of Silo incentives.

Currently Beanstalk does not have a gas efficient way to read the total Deposited BDV in the Silo. In a future Seed Gauge System, Beanstalk may need to calculate average Seeds per BDV across all Deposited assets.

Unripe Seed Parity at 0

Currently Beanstalk rewards 2 and 4 Seeds per BDV for urBEAN Deposits and urBEAN3CRV Deposits, respectively. There is a significant amount of Deposited urBEAN3CRV that could be Converted to repeg the Bean price. Beanstalk does not benefit from incentivizing Farmers to keep their Unripe assets in the Silo. Furthermore, the excessive accrual of Grown Stalk to Unripe asset holders during this extended period without Bean minting continues to make new Deposits less attractive.

ERC-1155 Deposits

Silo Deposits do not implement any standard interface, significantly limiting the composability of Bean's positive carry. Protocols such as Tractor, Depot, etc. need to implement Deposit specific functionality in order to support them. Protocols such as Seaport do not support Deposits.

Proposed Solution

Withdrawal Freeze Removal

Remove the Withdrawal Freeze entirely, allowing Farmers to Withdraw assets in the Silo directly to Circulating or Farm balances in a single transaction.

Specification

- Modify `withdrawDeposit(s)` to send the Withdrawn assets to Farm or Circulating balances using `LibTransfer.sendToken`, instead of creating a Withdrawal.
- Move `claimWithdrawal(s)` to a new `LegacyClaimWithdrawalFacet` to be used by Farmers that have unclaimed Withdrawals at the time of this upgrade.

Delayed Earned Bean Seigniorage

Revise the distribution of Earned Beans such that they Vest after the Vesting Period after the `gm` function is called. Set the Vesting Period to 10 blocks.

In practice, this means that a Farmer who Mows their Grown Stalk during the Vesting Period will be credited with that Stalk when Earned Beans are distributed next.

Specification

- Split `earnedBeans` into two `uint128` variables: `earnedBeans` and `newEarnedStalk`.
- In `rewardToSilo`, set `newEarnedStalk = amount.mul(C.getStalkperBean())` (i.e., the number of new Earned Stalk is equal to the number of Beans rewarded to the Silo).
- Modify `_balanceOfEarnedBeans` to properly exclude vesting Earned Stalk when computing a Farmer's Stalk balance.
- More info: RFC: Delayed Earned Bean Seigniorage.

Stems Upgrade

Introduce Stems for each token on the Deposit Whitelist, where Stems represent the Stalk Grown per BDV of Deposited value. This value starts at 0 at the time of Stem deployment (the Silo V3 BIP), or when a new token is added to the Deposit Whitelist.

Index Deposits based on Stems instead of by Season. A higher Stem count corresponds to an older Deposit. Storing Deposits based on Seasons does not allow for flexibility in Seeds per BDV values. With Stems, Seeds per BDV values can now be changed via governance and can have up to 6 decimals.

Store the total Deposited BDV of each whitelisted asset in the Silo. This enables future work on a Seed Gauge System that could need to calculate the average Seeds per BDV across all Deposited assets.

Storing Deposits based on Stems enables the cleanup of the Silo V1 and V2 storage systems into a single storage method that supports ERC-1155s. This requires a migration function (`mowAndMigrate`), in which Farmers submit all of their Deposits and the function that verifies their existence, removes them from the old storage slots, and migrates them into the Stem-based slots with ERC-1155 support.

A gas-saving migration function (`mowAndMigrateNoDeposits`) is available for accounts with no current Deposits.

As a result of the Stems migration, Farmers can now `mow` their Grown Stalk for each individual whitelisted asset. `mowMultiple` is added to allow Farmers to Mow their Grown Stalk for multiple whitelisted assets at once.

There are no penalties for waiting to migrate from Silo V2 (or V1) to Silo V3, but an account must be migrated in order to interact with the Silo again after the commitment of the Silo V3 BIP.

Specification

- Introduce decimals at the `stalkEarnedPerSeason` level, such that Seeds can now have up to 6 decimals.
- Modify existing `SiloFacet` , `ConvertFacet` and other facet functions from using Seeds to Stems.
- Add the `mowAndMigrate` and `mowAndMigrateNoDeposits` migration functions that migrate an account from Silo V2 (or V1) to Silo V3.
- Split up `SiloFacet` logic into `SiloFacet` , `MigrationFacet` and `ApprovalFacet` to reduce the `SiloFacet` size.
- Add a `token` input to `mow` and introduce `mowMultiple` in `SiloFacet` .

Unripe Seed Parity

Set the Seeds per BDV for both urBEAN and urBEAN3CRV to 0.

The Beanstalk DAO expressed a preference for setting the Unripe Seeds per BDV rewards to be equal in Temp-Check-1, and specifically expressed a preference for them both being set to 0 in Temp-Check-2.

ERC-1155 Deposits

Implement Silo Deposits using the ERC-1155 standard. ERC-1155s support an `id` and a `value` , matching the semi-fungible nature of Silo Deposits wherein they are non-fungible across Stems and fungible within the same number of Stems.

The ERC-20 and ERC-721 standards would not suffice for Silo Deposits:

- The ERC-20 standard would require a new deployment for each new Deposit.
- The ERC-721 standard does not support fungibility within a given `id` .

ERC-1155 tokens will be distributed to Farmers upon `mowAndMigrate` .

Specification

- For each ERC-1155 token, the `id` is the concatenation of the token address (20 bytes) and the Stem of the Deposit (12 bytes). The `value` is the number of tokens.
- Add ERC-1155 events to each function that creates, removes or transfers Deposits.
- Introduce a `MetadataFacet`, which contains on-chain metadata for each ERC-1155 token.

Technical Rationale

Withdrawal Freeze Removal

Adding the ability to send Withdrawn assets directly to the Farmer's Farm or Circulating balance without the need for a separate transaction reduces the friction of interacting with Beanstalk.

Introducing a `LegacyClaimWithdrawalFacet` facilitates backwards compatibility for Farmers with unclaimed Withdrawals at the time of this upgrade.

Stems Upgrade

Implementing Stems allows for Seeds per BDV to be changed via governance and set with greater precision, improving the economic flexibility of Silo incentives.

Providing a way for Beanstalk to calculate the average Seeds per BDV across all Deposited assets in a gas-efficient manner can be used to facilitate a future Seed Gauge System.

Allowing Farmers to `mow` or `mowMultiple` with distinct whitelisted assets increases the customizability of interacting with the Silo.

ERC-1155 Deposits

Implementing Silo Deposits as ERC-1155 tokens improves the composability of Beanstalk.

Economic Rationale

Withdrawal Freeze Removal

Improving the composability of Beanstalk and reducing the number of separate transactions required to interact with it improves the user experience and utility of Beanstalk and Beans.

Delayed Earned Bean Seigniorage

Removing the Withdrawal Freeze with no changes to Earned Bean distribution would create the opportunity to receive Bean seigniorage without taking on any exposure to the Bean price.

Implementing a vesting period for Earned Beans mitigates this attack vector.

Stems Upgrade

Improving the efficiency of the Silo and customizability of Seeds per BDV rewards improves the utility of Beanstalk and Beans.

Eliminating situations where Stalk is unnecessarily forfeited during Conversions reduces the friction of interacting with Beanstalk.

Unripe Seed Parity

There is an incentive not to Convert urBEAN3CRV to urBEAN given the loss of Seeds. Unripe Seed Parity reduces the incentive not to Convert urBEAN3CRV to urBEAN dramatically. Given that the potential sellable Beans to due to the Chop Penalty is so small compared to the liquidity in the pool, Beanstalk should incentivize more Converts from urBEAN3CRV to urBEAN. This should significantly contribute to peg maintenance in the short term.

There is no need to use Seeds per BDV rewards to create opportunity cost associated with Withdrawing and Redepositing Unripe assets. Excessive Grown Stalk per BDV Deposited associated with Unripe assets is likely hurting Beanstalk's ability to attract new Depositors.

ERC-1155 Deposits

The positive carry of Beans is generally not accessible in DeFi given the lack of composability with Silo Deposits. Improving the composability of Silo Deposits by tokenizing them as ERC-1155s improves the user experience and utility of Beanstalk and Beans.

Contract Changes

Initialization Contract

The `init` function on the following `InitBipNewSilo` contract is called:

- `TBD`

Silo Facet

The following `SiloFacet` is removed from Beanstalk:

- `0xe56607c4396c546cb6a137659e42a5fd16e17cfe`

The following `SiloFacet` is added to Beanstalk:

- TBD

`SiloFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`balanceOf`	`TBD`	Add	View	✓
`balanceOfBatch`	`TBD`	Add	View	✓
`balanceOfEarnedBeans`	`0x3e465a2e`	Replace	View	
`balanceOfEarnedSeeds`	`0x602aa123`	Remove	View	
`balanceOfEarnedStalk`	`0x341b94d5`	Replace	View	
`balanceOfGrownStalk`	`0x249564aa`	Replace	View	✓
`balanceOfPlenty`	`0x896651e8`	Replace	View	✓
`balanceOfRainRoots`	`0x69fbad94`	Replace	View	
`balanceOfRoots`	`0xba39dc02`	Replace	View	
`balanceOfSeeds`	`0x4916bc72`	Remove	View	
`balanceOfSop`	`0xa7bf680f`	Replace	View	
`balanceOfStalk`	`0x8eeae310`	Replace	View	
`claimPlenty`	`0x45947ba9`	Replace	Call	
`deposit`	`0xf19ed6be`	Remove	Call	
`deposit`	`TBD`	Add	Call	✓
`getDeposit`	`0x8a6a7eb4`	Remove	View	
`getDeposit`	`TBD`	Add	View	
`getSeedsPerToken`	`TBD`	Add	View	✓
`getTotalDeposited`	`0x0c9c31bd`	Replace	View	
`getTotalDepositedBDV`	`TBD`	Add	View	✓
`grownStalkForDeposit`	`TBD`	Add	View	✓
`inVestingPeriod`	`TBD`	Add	View	✓
`lastSeasonOfPlenty`	`0xbe6547d2`	Replace	View	

Name	Selector	Action	Type	New Functionality
`lastUpdate`	`0xcb03fb1e`	Replace	View	
`migrationNeeded`	`TBD`	Add	View	✓
`mow`	`TBD`	Add	Call	✓
`mowMultiple`	`TBD`	Add	Call	✓
`plant`	`0x779b3c5c`	Replace	Call	
`safeBatchTransferFrom`	`TBD`	Add	Call	✓
`safeTransferFrom`	`TBD`	Add	Call	✓
`seasonToStem`	`TBD`	Add	View	✓
`stemStartSeason`	`TBD`	Add	View	✓
`stemTipForToken`	`TBD`	Add	View	✓
`tokenSettings`	`0xe923e8d4`	Replace	View	
`totalEarnedBeans`	`0xfd9de166`	Replace	View	
`totalRoots`	`0x46544166`	Replace	View	
`totalSeeds`	`0xd8bd0d9d`	Replace	View	
`totalStalk`	`0x7b52fadf`	Replace	View	
`transferDeposit`	`0x9e32d261`	Replace	Call	✓
`transferDeposits`	`0x0d2615b1`	Replace	Call	✓
`update`	`0x1c1b8772`	Remove	Call	
`withdrawDeposit`	`TBD`	Add	Call	✓
`withdrawDeposit`	`0x7af9a0ce`	Remove	Call	
`withdrawDeposits`	`TBD`	Add	Call	✓
`withdrawDeposits`	`0xb189d9c8`	Remove	Call	
`withdrawFreeze`	`0x55926690`	Remove	View	

`SiloFacet` Event Changes

Name	Change
`AddDeposit`	Updated to reflect new Deposit accounting system
`AddWithdrawal`	Removed
`RemoveDeposit`	Updated to reflect new Deposit accounting system
`RemoveDeposits`	Updated to reflect new Deposit accounting system
`RemoveDeposits`	Updated to reflect new Deposit accounting system
`SeedsBalanceChanged`	Removed
`TransferBatch`	Emitted when multiple Deposits are removed or transferred
`TransferSingle`	Emitted when a Deposit is created, removed or transferred

Legacy Claim Withdrawal Facet

The following `LegacyClaimWithdrawalFacet` is added to Beanstalk:

- TBD

`LegacyClaimWithdrawalFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`claimWithdrawal`	`0x488e94dc`	Replace	Call	
`claimWithdrawals`	`0x764a9874`	Replace	Call	
`getTotalWithdrawn`	`0xb1c7a20f`	Replace	View	
`getWithdrawal`	`0xe23c96a4`	Replace	View	

`LegacyClaimWithdrawalFacet` Event Changes

None.

Season Facet

The following `SeasonFacet` is removed from Beanstalk:

- `0x9c9360C85cd020D4eF38775F6ADEdD38931f1731`

The following `SeasonFacet` is added to Beanstalk:

- TBD

`SeasonFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`abovePeg`	`0x2a27c499`	Replace	View	
`gm`	`0x64ee4b80`	Replace	Call	
`paused`	`0x5c975abb`	Replace	View	
`plentyPerRoot`	`0xe60d7a83`	Replace	View	
`poolDeltaB`	`0x471bcdbe`	Replace	View	
`rain`	`0x43def26e`	Replace	View	
`season`	`0xc50b0fb0`	Replace	View	
`seasonTime`	`0xca7b7d7b`	Replace	View	
`sunrise`	`0xfc06d2a6`	Replace	Call	
`sunriseBlock`	`0x3b2ecb70`	Replace	View	
`time`	`0x16ada547`	Replace	View	
`totalDeltaB`	`0x06c499d8`	Replace	View	
`weather`	`0x686b6159`	Replace	View	

`SeasonFacet` Event Changes

None.

Migration Facet

The following `MigrationFacet` is added to Beanstalk:

- TBD

`MigrationFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`balanceOfGrownStalkUpToStemsDeployment`	`TBD`	Add	View	✓

Name	Selector	Action	Type	New Functionality
`balanceOfLegacySeeds`	`TBD`	Add	View	✓
`getDepositLegacy`	`TBD`	Add	View	✓
`mowAndMigrate`	`TBD`	Add	Call	✓
`mowAndMigrateNoDeposits`	`TBD`	Add	Call	✓

`MigrationFacet` Event Changes

None.

Approval Facet

The following `ApprovalFacet` is added to Beanstalk:

- `0xTODO`

`ApprovalFacet` Function Changes

Here is the table ordered alphabetically according to the Name column and with each Selector cell updated to `TBD`:

Name	Selector	Action	Type	New Functionality
`approveDeposit`	`0x1302afc2`	Replace	Call	
`decreaseDepositAllowance`	`0xd9ee1269`	Replace	Call	
`depositAllowance`	`0x2a6a8ef5`	Remove	View	
`depositAllowance`	`TBD`	Add	View	
`depositPermitDomainSeparator`	`0x8966e0ff`	Replace	View	
`depositPermitNonces`	`0x843bc425`	Replace	View	
`increaseDepositAllowance`	`0x5793e485`	Replace	Call	
`isApprovedForAll`	`TBD`	Add	View	✓
`permitDeposit`	`0x120b5702`	Replace	Call	
`permitDeposits`	`0xd5770dc7`	Replace	Call	

Name	Selector	Action	Type	New Functionality
`setApprovalForAll`	`TBD`	Add	Call	✓

`ApprovalFacet` Event Changes

Name	Change
`ApprovalForAll`	New event

Convert Facet

The following `ConvertFacet` is removed from Beanstalk:

- `0xd24959190e29b13e1accb578d02b15d73a2231f3`

The following `ConvertFacet` is added to Beanstalk:

- `0xTODO`

`ConvertFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`convert`	`0x3b2a1b28`	Remove	Call	✓
`convert`	`TBD`	Add	Call	✓
`enrootDeposit`	`0xd5d2ea8c`	Remove	Call	✓
`enrootDeposit`	`TBD`	Add	Call	✓
`enrootDeposits`	`0x83b9e85d`	Remove	Call	✓
`enrootDeposits`	`TBD`	Add	Call	✓
`getAmountOut`	`0x4aa06652`	Replace	View	
`getMaxAmountIn`	`0x24dd285c`	Replace	View	

`ConvertFacet` Event Changes

Name	Change
`RemoveDeposit`	New event

Name	Change
`RemoveDeposits`	Updated to reflect new Deposit accounting system
`TransferBatch`	New event

Whitelist Facet

The following `WhitelistFacet` is removed from Beanstalk:

- `0xaea0e6e011106968adc7943579c829e49efddad0`

The following `WhitelistFacet` is added to Beanstalk:

- `0xTODO`

`WhitelistFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`dewhitelistToken`	`0x86b40a1b`	Replace	Call	
`updateStalkPerBdvPerSeasonForToken`	`TBD`	Add	Call	✓
`whitelistToken`	`0xd8a6aafe`	Replace	Call	✓

`WhitelistFacet` Event Changes

Name	Change
`UpdatedStalkPerBdvPerSeason`	New event
`WhitelistToken`	Updated to reflect new Deposit accounting system

Metadata Facet

The following `MetadataFacet` is added to Beanstalk:

- `0xTODO`

`MetadataFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`getDepositMetadata`	`0xTODO`	Add	View	✓
`imageURI`	`0xTODO`	Add	View	✓
`uri`	`0xTODO`	Add	View	✓

`MetadataFacet` Event Changes

None.

Token Facet

The following `TokenFacet` (s) are being removed from Beanstalk:

- `0x8d00ef08775872374a327355fe0fdbdece1106cf`
- `0x50eb0085c31dfa8cf86ca16def77520e762ead4a`

The following `TokenFacet` is added to Beanstalk:

- `0xTODO`

`TokenFacet` Function Changes

Name	Selector	Action	Type	New Functionality
`approveToken`	`0xda3e3397`	Replace	Call	
`decreaseTokenAllowance`	`0x0bc33ce4`	Replace	Call	
`getAllBalance`	`0xfdb28811`	Replace	View	
`getAllBalances`	`0xb6fc38f9`	Replace	View	
`getBalance`	`0xd4fac45d`	Replace	View	
`getBalances`	`0x6a385ae9`	Replace	View	
`getExternalBalance`	`0x4667fa3d`	Replace	View	
`getExternalBalances`	`0xc3714723`	Replace	View	
`getInternalBalance`	`0x8a65d2e0`	Replace	View	
`getInternalBalances`	`0xa98edb17`	Replace	View	
`increaseTokenAllowance`	`0xb39062e6`	Replace	Call	

Name	Selector	Action	Type	New Functionality
`onERC1155BatchReceived`	`TBD`	Add	View	✓
`onERC1155Received`	`TBD`	Add	View	✓
`permitToken`	`0x7c516e94`	Replace	Call	
`tokenAllowance`	`0x8e8758d8`	Replace	View	
`tokenPermitDomainSeparator`	`0x1f351f6a`	Replace	View	
`tokenPermitNonces`	`0x4edcab2d`	Replace	View	
`transferInternalTokenFrom`	`0xd3f4ec6f`	Replace	Call	
`transferToken`	`0x6204aa43`	Replace	Call	
`unwrapEth`	`0xbd32fac3`	Replace	Call	
`wrapEth`	`0x1c059365`	Replace	Call	

`TokenFacet` Event Changes

None.

Beans Minted

None.

Audit

The commit hash of this BIP is 55813422bc515a5e36469108d4c4f835158fa8fd.

Halborn has performed an audit of this BIP. You can view the Halborn audit report here (**TBD**).

Effective

Immediately upon commitment.