Pinto: A Scalable Leviathan-Free Low-Volatility Money Protocol

Brendan Sanderson
@brendaann__

Ben Weintraub
@bwein_

Brian Nguyen
@fr1j00

Austin Moninger
@isthisaustin

Drew Davis
zsdrew@proton.me

Nikolaos Katsios
nkatsios@protonmail.com

Ryan Ham
@gkaehddls

Calvin Ham
calvinham@proton.me

John Deere
johndeere8r420@protonmail.com

Soil King
soilking435@gmail.com

Motasem Ahmed
@itsmod323

Aaron Ash
aaron@aaronash.com

Gabe Weintraub
gabeweintraub1@proton.me

pinto.money

Whitepaper Version: 1.0.0
Published: February 24, 2025
Code Version: 1.0.0[1]
Deployed: November 19, 2024

"You broke the heads of Leviathan in pieces, and gave him to be meat to the people inhabiting the wilderness."

- Psalm 74:14[2]

Abstract

The ideal money is trustless and volatility free. Bitcoin is trustless, but has no native volatility minimization scheme. The tremendous success and adoption of Bitcoin as a store of value (SoV) has demonstrated the power of trustless technology and the primacy of trustlessness with respect to storing value. However, the widespread adoption of trusted stablecoins (*e.g.*, USDT, USDC) as media of exchange (MoE) and units of account (UoA) on trustless computer networks has demonstrated the primacy of freedom from volatility with respect to exchanging and accounting for value. Because value is subjective, dynamic and relative, (1) money will never be entirely free from volatility, and (2) a money protocol optimizing for low volatility can not be trustless without the development of a trustless value index oracle. We propose a trust-minimized, low-volatility money protocol that issues a fungible token, Pinto, optimized to be a MoE and UoA. A bounty based timekeeping mechanism, non-custodial deposit and credit facilities, a variable supply and autonomously adjusting interest rates incentivize user behavior that minimizes the volatility of Pinto with respect to its value target over time, without ever forcing users to act or forcibly removing Pinto from their accounts. Future development of a trustless value index oracle can remove the only trust required by the protocol.

[1] github.com/pinto-org/protocol
[2] biblegateway.com/verse/en/Psalm%2074:14

Table of Contents

1 Introduction

The creation of Bitcoin began a technological revolution still in its infancy. Bitcoin proved the ability of open source software with endogenous incentives to create positive sum externalities (*i.e.*, endogenous value), without requiring trust in any party or set of parties (*i.e.*, a leviathan). Building an economy on such leviathan-free software that radically (1) empowers individuals with sovereignty and community, and (2) realigns incentives to encourage creating value instead of extracting it, is humanity's best bet to avoid being consumed by the ever more present technology-enabled Orwellian dystopia.

Money is the root of economy. It is no wonder, therefore, that Bitcoin (*i.e.*, leviathan-free money) rang the bell to announce the revolution. However, money has historically fulfilled three unique but deeply complimentary roles in an economy: storing value, acting as a media through which to exchange value, and accounting for value. Bitcoin, incredible as it is, has only found product market fit as a SoV, not as a MoE or UoA. The volatility of the value of Bitcoin is the cause of the shortcoming. This is clearly demonstrated by the massive success of leviathan-dependent stablecoin protocols that issue tokens pegged to some exogenous unit of value, typically the US Dollar (USD) (*i.e.*, the globally dominant MoE, UoA and, prior to the creation of Bitcoin, SoV) on trustless computer networks.

Trust is a least common denominator problem: a system requires as much trust as its component that requires the most trust. As of today, the reliance of users of, and protocols on, trustless computer networks (*e.g.*, Ethereum, Solana) on leviathan-dependent MoE and UoA is so great that, in practice, the networks themselves lose their trustlessness: the authorities that control these protocols have so much power they are de facto the ultimate determinants of the state of the networks.[3] Without a leviathan-free MoE and UoA, trustless systems will struggle to remain free from leviathans.

Credit is the ideal tool to minimize volatility because it (1) enables using future value in the present, (2) is infinitely scalable, and (3) does not require a leviathan. During periods of excess supply (*i.e.*, insufficient demand), Pinto uses credit to call forward future demand to the present. In practice, the protocol borrows Pinto from the market and then burns the borrowed excess Pinto to decrease the supply, thereby reducing downside volatility. During periods of excess demand (*i.e.*, insufficient supply) the protocol mints new Pinto and pays back creditors with interest, thereby reducing upside volatility. The more debt Pinto repays with interest, the more (1) creditworthy it is perceived, (2) it can borrow at lower interest rates, and (3) it can sustainably reduce future downside volatility.

Pinto consists of four interconnected components: (1) a bounty based timekeeping and execution facility, (2) a liquid deposit facility, (3) a first in, first out (FIFO) credit facility, and (4) a suite of tools that reduce the friction of interacting with Pinto and other Base-native protocols. Protocol-native incentives coordinate the components to create low-volatility money with competitive carrying costs and deep liquidity in a permissionless and (almost) leviathan-free fashion.

Pinto is designed from economic first principles to create low-volatility money. The following principles inspire Pinto:

- Incentive structures determine behaviors;
- Low concentration of ownership;
- Strong credit;
- Deep liquidity;
- The marginal rate of substitution;
- Low friction; and
- Equilibrium.

[3] haseebq.com/ethereum-is-now-unforkable-thanks-to-defi

2 Previous Work

Pinto is the culmination of previous development, evolution and experimentation within the DeFi ecosystem.

Pinto requires a robust, trustless computer network that supports composability and both fungible and semi-fungible token standards (*e.g.*, Ethereum and Base) with a network-native automated market maker (AMM) decentralized exchange with a manipulation-resistant oracle (*e.g.*, Pinto Exchange[4]).

Pinto is a fork of Beanstalk[5].

3 Farm

Well designed decentralized protocols create utility for end users without requiring, but never preventing, participation in protocol maintenance. Protocol-native incentives encourage performance of work to create utility for end users. Low barriers to and variety in work enable a diverse set of participants. A diverse set of well incentivized workers can create censorship resistant utility.

Pinto peg maintenance takes place on the *Farm*. Anyone can join the *Farm* to profit from participation in protocol maintenance. The *Farm* has four primary components: the *Sun*, *Silo*, *Field* and *Toolshed*. Protocol-native financial incentives coordinate the components to regularly cross the price of 1 over its value target without collateral.

The *Sun* offers payment for participating in timekeeping and code execution. Time on the *Farm* is kept in *Seasons*. Anyone can earn Pinto for successfully calling the `gm` function to begin the next *Season* at the top of each hour.

The *Silo* offers passive yield opportunities to owners of and other assets (λ) on the *Deposit Whitelist* (Λ) (*i.e.*, $\subset \lambda \in \Lambda$) for contributing to peg maintenance. Anyone can become a *Stalkholder* by *Depositing* λ into the *Silo* and earning *Stalk*. *Stalkholders* are rewarded Pinto when the Pinto supply increases on a pro rata basis. Active contributions to peg maintenance within the *Silo* can earn additional *Stalk*.

The *Field* offers yield opportunities to creditors for supporting peg maintenance. Anyone can become a *Sower* by lending Pinto that are not in the *Silo* to the protocol. *Sowers* are repaid with interest when the Pinto supply increases on a first in, first out (FIFO) basis.

The *Toolshed* offers a suite of tools that decrease friction to using the protocol and participating in peg maintenance.

4 Sun

The Pinto peg maintenance mechanism requires a protocol-native timekeeping mechanism and regular code execution on the Base network. The *Sun* creates a cost-efficient protocol-native timekeeping mechanism and incentivizes cost-efficient code execution on Base at regular intervals. In general, Pinto uses Base block timestamps (E), such that $E \in \mathbb{Z}^+$.

4 pinto.exchange
5 bean.money

We define a *Season* (t), such that $t \in \mathbb{Z}^+$, as an approximately 3,600 second (1 hour) interval. The first *Season* begins when a successful transaction on the Base network that includes a `gm` function call is committed. When Pinto accepts the `gm` function call, the necessary code is executed.

Pinto only accepts one `gm` function call per *Season*. Pinto accepts the first `gm` function call provided that the timestamp in the Base block containing it is sufficiently distant from the timestamp in the Base block containing the Pinto deployment (E_1).

The minimum timestamp Pinto accepts a `gm` function call for a given t (E_t^{min}), $\forall\ E_t^{\text{min}}$ such that $1 < t$, and E_1 is:

$$E_t^{\text{min}} = 3600\left(\left\lfloor \frac{E_1}{3600} \right\rfloor + t\right)$$

The cost to execute the `gm` function changes depending on the traffic on the Base network and the state of Pinto. Pinto covers the transaction cost by rewarding the sender of an accepted `gm` function call with newly minted Pinto.

To encourage regular `gm` function calls even during periods of congestion on the Base network while minimizing cost, the Pinto reward for successfully calling the `gm` function for t (a_t), starts at ☼, such that a_t, $\text{☼} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, and compounds 2.01% every additional 2 seconds that elapse past E_t^{min} for 300 seconds.

Therefore, we define a_t for a given timestamp of the current block (E_Ξ) and E_t^{min} as:

$$a_t = \text{☼} \times 1.0201^{\min\left(2 \times \left\lceil \frac{E_\Xi - E_t^{\text{min}}}{2} \right\rceil,\ 300\right)}$$

To minimize the cost of calculating a_t, the *Sun* uses a binomial estimation with a margin of error of less than 0.05% of a_t. Thus, Pinto creates a cost-efficient protocol-native timekeeping mechanism and ensures cost-efficient code execution on the Base network at regular intervals.

5 Silo

The *Silo* is the protocol-native *Deposit* facility.

The *Silo* uses the *Stalk System* to create protocol-native financial incentives that contribute to security and peg maintenance. The *Silo* uses the *Seed Gauge System* to create protocol-native financial incentives that contribute to peg maintenance. Anyone can become a *Stalkholder* by *Depositing* assets on the *Deposit Whitelist* into the *Silo* to earn *Stalk* and *Seeds*.

5.1 Deposits, Withdrawals, Conversions and Transfers

λ can be *Deposited* into, *Withdrawn* from and *Transferred* within the *Silo* at any time. λ can be *Converted* within the *Silo* under permitted conditions.

The protocol rewards *Stalk* to *Depositors* upon the completion of a *Germination* period. The protocol rewards *Seeds* to *Depositors* immediately upon *Depositing* λ into the *Silo*. *Deposits* implement the ERC-1155 Standard.[6]

[6] ethereum.org/en/developers/docs/standards/tokens/erc-1155

Upon a *Deposited* asset's *Withdrawal* from the *Silo*, the *Deposit* token is burned in addition to the number of *Stalk* and *Seeds* rewarded to it.

Conversions of *Deposited* λ to *Deposited* λ' are permissioned by a *Conversion Whitelist*. Upon an asset's *Conversion* within the *Silo*, *Stalk* and *Seeds* may be rewarded or burned depending on the details of the *Conversion*.

The number of *Stalk* and *Seeds* rewarded to a *Deposit* are included in its *Transfer* to another address.

5.1.1 Deposit Whitelist

Any ERC-20 Standard token can be added to and removed from Λ via the *Pinto Contract Multisig* (*PCM*) (see Appendix). is always on the *Deposit Whitelist*.

In order for a given λ to be added to Λ, the protocol requires (1) its token address, (2) a function to calculate the flash-loan-resistant Pinto-Denominated-Value (PDV) of a given number of *Deposited* λ, ($f^{L^\lambda}(z^\lambda)$), such that $f^{L^\lambda}: \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^+\} \rightarrow \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, where z^λ is the number of λ *Deposited*, (3) the number of *Stalk* per PDV of λ *Deposited* (k^λ), such that $k^\lambda \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^+\}$, (4) the initial number of *Seeds* per PDV of λ *Deposited* ($c^\lambda_\circ$), (5) the initial *LP Seed Gauge Points* of λ ($\mathscr{L}^\lambda_\circ$), such that $\mathscr{L}^\lambda_\circ \in \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, (6) a function to calculate the *LP Seed Gauge Points* of λ ($f^{\mathscr{L}^\lambda}$), (7) a function to calculate the USD price of the non-Pinto asset in λ ($f^{\$^\lambda}$), such that $f^{\$^\lambda}: \mathbb{U} \rightarrow \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, (8) a function to calculate the *Liquidity Weight* of λ ($f^{\mathscr{W}^\lambda}$), such that $f^{\mathscr{L}^\lambda},\ f^{\mathscr{W}^\lambda}: \mathbb{U} \rightarrow \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, and (9) the optimal percent of *Deposited* LP PDV of λ ($\mathscr{B}^{\lambda^*}$), such that $z^\lambda,\ c^\lambda_\circ,\ \mathscr{B}^{\lambda^*} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

5.1.2 Earned

Pinto minted to the *Silo* are automatically *Deposited* into the *Silo* and distributed to *Stalkholders* as *Earned* (η), such that $\eta \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$.

Earned are rewarded *Stalk* upon *Deposit*, but (1) are not rewarded *Seeds*, (2) cannot be *Converted* and (3) cannot be *Transferred* until the *Stalkholder* calls the `plant` function. Upon calling the `plant` function, (1) the *Earned* become and (2) the *Seeds* associated with the *Earned* are *Planted* to start *Growing Stalk*.

5.1.3 Germination

Every λ *Deposit* except *Earned* are subject to a *Germination* period during which *Deposits* (1) cannot be *Converted*, (2) are not eligible for mints and (3) are not included in *Seed Gauge System* PDV calculations.

Germination ends upon the completion of a full *Season* after *Deposit*.

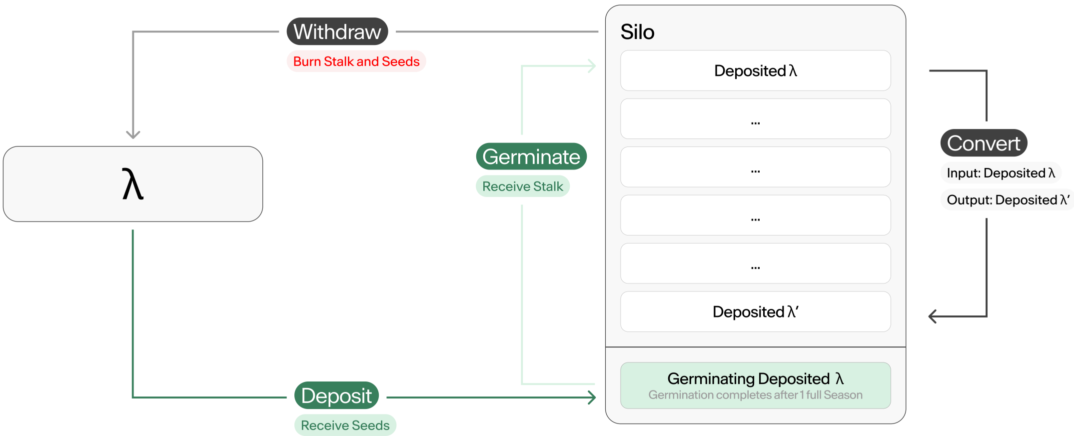

Figure 1: Silo

5.1.4 Conversion Whitelist

Conversions can be added to and removed from the *Conversion Whitelist* via the *PCM*.

In order for a given *Conversion* to be added to the *Conversion Whitelist*, the protocol requires (1) the from token address, (2) the to token address, (3) a list of conditions under which the *Conversion* is permitted and (4) a function to determine the number of λ' received for *Converting* a given number of λ ($f^{\lambda \to \lambda'}(z^{\lambda})$), such that $f^{\lambda \to \lambda'}: \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^{+}\} \to \{j \times 10^{-\lambda'} \mid j \in \mathbb{Z}^{+}\}$, where z^{λ} is the number of λ *Converted* or (5) a function to determine the number of *Mown Stalk* rewarded or burned for *Converting* a given number of λ to λ' ($f^{\lambda \to \lambda'}(z^{\lambda} \to z^{\lambda'})$), such that $f^{\lambda \to \lambda'}: \{j \times 10^{-\lambda} \to j' \times 10^{-\lambda'} \mid j,\ j' \in \mathbb{Z}^{+}\} \to \{j \times 10^{-16} \mid j \in \mathbb{Z}^{+}\}$, where z^{λ} is the number of λ *Converted* and where $z^{\lambda'}$ is the number of λ output by the *Conversion*.

5.2 Stalk System

The *Stalk System* decentralizes ownership of over time and creates protocol-native financial incentives to (1) *Deposit* assets into the *Silo*, (2) leave assets *Deposited* in the *Silo* and (3) align *Depositors*' interests with the health of the protocol.

Stalk are not liquid. Every *Season*, $\frac{1}{10000}$ additional *Stalk Grows* from each *Seed*. *Grown Stalk* become *Stalk* when *Mown*. Upon a *Deposited* asset's *Withdrawal* from the *Silo*, the number of *Stalk* and *Grown Stalk* rewarded to it must be burned.

Stalkholders are entitled to a portion of Pinto mints. The distribution of Pinto paid to a *Stalkholder* are proportional to their *Stalk* holdings relative to total outstanding *Stalk*. *Stalk* holdings become less concentrated over time. Active contributions to peg maintenance within the *Silo* can earn additional *Stalk* via *Conversions*.

Grown Stalk from λ *Deposits* are automatically *Mown* each time a *Stalkholder* interacts with λ in the Silo (*i.e.*, *Deposit*, *Withdraw*, *Convert*, *Transfer* and *Plant*), or anytime a `mow` function is called with λ and their address.

5.3 Seed Gauge System

The *Seed Gauge System* decentralizes ownership of 🫘 over time and creates protocol-native financial incentives to (1) *Deposit* assets into the *Silo*, (2) leave assets *Deposited* in the *Silo* and (3) align *Depositors'* interests with the health of the protocol.

Seeds are not liquid. At the beginning of each *Season*, the total number of *Seeds* is calculated such that a new *Deposit* with an average number of *Seeds* per PDV, averaged across all *Deposits* in the *Silo* that are not *Germinating*, will catch up to the average *Mown Stalk* per PDV across all *Deposits* in the *Silo* at the beginning of t in $\mathscr{T}$ *Seasons*, such that $\mathscr{T} \in \mathbb{Z}^+$.

The *Crop Ratio* during t ($\mathscr{C}_t$), such that $\mathscr{C}_t \in \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, determines the ratio between (1) the *Seeds* per 🫘 *Deposited* in the *Silo* that are not *Germinating* at the beginning of t ($\mathbf{L}_t^{🫘}$) (*i.e.*, $\frac{c_t^{🫘}}{\mathbf{L}_t^{🫘}}$) and (2) the *Seeds* per PDV of the λ with the highest *LP Seed Gauge Points* per PDV of λ *Deposited* in the *Silo* that are not *Germinating* at the beginning of t ($\mathbf{L}_t^{\lambda^{\max}}$) (*i.e.*, $\frac{c_t^{\lambda}}{\mathbf{L}_t^{\lambda^{\max}}}$), such that $\mathbf{L}_t^{🫘}$, $\mathbf{L}_t^{\lambda^{\max}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

The distribution of *Seeds* across λ during t is a function of (1) the *Crop Ratio* and (2) the distribution of *LP Seed Gauge Points*.

5.4 Calculating Stalk

A *Stalkholder's* total *Stalk* is the sum of the *Stalk* for each of their *Deposits* that are not *Germinating*.

When a *Stalkholder Deposits* λ during i, they update their total number of λ *Deposited* during *Season* i (Z_i^{λ}), such that $Z_i^{\lambda} \in \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^+\}$, and its total PDV ($L_i^{\lambda}$), such that $L_i^{\lambda} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, as $Z_i^{\lambda} \mathrel{+}= z^{\lambda}$ and $L_i^{\lambda} = f^{L^{\lambda}}(Z_i^{\lambda})$, where z^{λ} is the number of λ *Deposited*.

The number of *Stalk* for a given λ *Deposit* are determined by its PDV, k^{λ}, *Season* of *Deposit*, the last *Season* the *Stalkholder Mowed* λ *Deposits* ($\varkappa^{\lambda}$), such that $\varkappa^{\lambda} \in \mathbb{Z}^+$, and the number of *Seeds* per PDV of λ *Deposited* during t (c_t^{λ}), such that $c_t^{\lambda} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, in every *Season* from when it was *Deposited* until $\varkappa^{\lambda}$.

The number of *Stalk* during t for a given λ *Deposit Deposited* during i and last *Mowed* during $\varkappa^{\lambda}$ ($k_{t,i,\varkappa^{\lambda}}^{\lambda}$), such that $k_{t,i,\varkappa^{\lambda}}^{\lambda} \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^+\}$, is:

$$k_{t,i,\varkappa^{\lambda}}^{\lambda} = L_i^{\lambda}\left(k^{\lambda} + \sum_{i}^{\varkappa^{\lambda}}\left(\frac{c_i^{\lambda}}{10000}\right)\right)$$

A *Stalkholder's* total *Stalk* during t (K_t), such that $K_t \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^+\}$, is:

$$K_t = \sum_{\lambda \in \Lambda} \sum_{i=1}^{\varkappa^{\lambda}} k_{t,i,\varkappa^{\lambda}}^{\lambda}$$

The total *Stalk* in the *Silo* during t ($\mathbf{K}_t$), such that $\mathbf{K}_t \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^+\}$, is the sum of every *Stalkholder's* total *Stalk* during t:

$$\mathbf{K}_t = \sum K_t$$

The total *Mown Stalk* in the *Silo* during t ($\widehat{\mathbf{K}_t}$), such that $\widehat{\mathbf{K}_t} \in \{j \times 10^{-16} \mid j \in \mathbb{N}\}$, is the sum of every *Stalkholder's* total *Stalk* during t, less the *Stalk* rewarded upon *Deposit* across all *Deposits*:

$$\widehat{\mathbf{K}_t} = \mathbf{K}_t - \sum \sum_{\lambda \in \Lambda} \sum_{i=1}^{t} L_i^\lambda \times k^\lambda$$

The number of *Grown Stalk* from λ *Deposits* that can be *Mown* during t to start earning Pinto seigniorage for a given *Deposit Deposited* during i and last *Mowed* during $\varkappa^\lambda$ ($g_{t,i,\varkappa^\lambda}^\lambda$), such that $g_{t,i,\varkappa^\lambda}^\lambda \in \{j \times 10^{-16} \mid j \in \mathbb{N}\}$, is:

$$g_{t,i,\varkappa^\lambda}^\lambda = L_i^\lambda \left(\sum_{\varkappa^\lambda+1}^{t} \frac{c_i^\lambda}{10000} \right)$$

A *Stalkholder's* total *Grown Stalk* that can be *Mown* during t (G_t), such that $G_t \in \{j \times 10^{-16} \mid j \in \mathbb{N}\}$, is:

$$G_t = \sum_{\lambda \in \Lambda} \sum_{\varkappa^\lambda+1}^{t} g_{t,i,\varkappa^\lambda}^\lambda$$

When a *Stalkholder Converts* multiple λ *Deposits*, their *Mown Stalk* per PDV amounts are averaged together, weighted by their PDVs, and rounded up.

5.5 Calculating Seeds

A *Stalkholder's* total *Seeds* is the sum of the *Seeds* for each of their *Deposits*.

The number of *Seeds* per PDV during t for a given λ are determined by (1) the total *Seeds* that *Season* ($\mathbf{C}_t$), (2) the *Seed Gauge Points* of λ ($\mathscr{S}_t^\lambda$), (3) $\mathbf{L}_t^\lambda$, and (4) the total *Seed Gauge Points* ($\mathscr{S}_t$), such that $\mathbf{C}_t, \mathscr{S}_t^\lambda, \ \mathscr{S}_t \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

We define $\mathbf{C}_t$ for a given initial ($\overline{\mathbf{C}}^\circ$) and minimum average *Seeds* per PDV ($\overline{\mathbf{C}}^{\min}$), such that $\overline{\mathbf{C}}^\circ, \ \overline{\mathbf{C}}^{\min} \in \{j \times 10^{-12} \mid j \in \mathbb{Z}^+\}$, total PDV of λ *Deposited* in the *Silo* that are not *Germinating* at the beginning of t ($\mathbf{L}_t^\lambda$), such that $\mathbf{L}_t^\lambda \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, $\forall \ \lambda \in \Lambda$, $\widehat{\mathbf{K}_t}$ and $\mathscr{T}$ as:

$$\mathbf{C}_t = \begin{cases} \frac{\overline{\mathbf{C}}^\circ}{\sum_{\lambda \in \Lambda} \mathbf{L}_i^\lambda} & \text{if } t < \mathscr{T} \mid \frac{10000 \times \widehat{\mathbf{K}_t}}{\mathscr{T} \times \sum_{\lambda \in \Lambda} \mathbf{L}_t^\lambda} < \frac{\overline{\mathbf{C}}^\circ}{\sum_{\lambda \in \Lambda} \mathbf{L}_i^\lambda} \ \forall \ \mathscr{T} < t \\ \max\left(\frac{\overline{\mathbf{C}}^{\min}}{\sum_{\lambda \in \Lambda} \mathbf{L}_i^\lambda}, \ \frac{10000 \times \widehat{\mathbf{K}_t}}{\mathscr{T} \times \sum_{\lambda \in \Lambda} \mathbf{L}_t^\lambda} \right) & \text{else} \end{cases}$$

The *Seed Gauge Points* of λ during t are a function of $\mathscr{C}_t$, the *LP Seed Gauge Points* per PDV of λ during t ($\mathscr{V}_t^\lambda$), such that $\mathscr{V}_t^\lambda \in \{j \times 10^{-12} \mid j \in \mathbb{Z}^+\} \ \forall \ \lambda$, and $\mathbf{L}_t^\lambda$.

The *Crop Ratio* is a function of the *Crop Scalar* ($\mathscr{X}_t$), such that $\mathscr{X}_t \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, the minimum *Crop Ratio* ($\mathscr{C}^{\min}$), the maximum *Crop Ratio* ($\mathscr{C}^{\max}$), the *Rain Crop Ratio* ($\mathscr{C}^{\text{rain}}$), such that $\mathscr{C}^{\min},\ \mathscr{C}^{\max},\ \mathscr{C}^{\text{rain}} \in \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, the liquidity and time weighted average price of 🫘1 over the previous *Season* ($P_{\overline{t-1}}$) compared to the protocol value target (V), such that $P_{\overline{t-1}},\ V \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, the *Pod Rate* at the end of the previous *Season* (R^D_{t-1}), and the level below which debt is considered excessively low ($R^{D^{\text{lower}}}$), such that $R^D_{t-1},\ R^{D^{\text{lower}}} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$.

We define $\mathscr{C}_t$ for a given $\mathscr{C}^{\text{rain}}$, V, $P_{\overline{t-1}}$, R^D_{t-1}, $R^{D^{\text{lower}}}$, $\mathscr{C}^{\min}$, $\mathscr{X}_t$ and $\mathscr{C}^{\max}$ as:

$$\mathscr{C}_t = \begin{cases} \mathscr{C}^{\text{rain}} & \text{if } V < P_{\overline{t-1}} \ \&\&\ R^D_{t-1} < R^{D^{\text{lower}}} \\ \mathscr{C}^{\min} + \mathscr{X}_t \ * \ (\mathscr{C}^{\max} - \mathscr{C}^{\min}) & \text{else} \end{cases}$$

We define $\mathscr{V}_t^\lambda$ for a given *LP Seed Gauge Points* of λ during t ($\mathscr{L}_t^\lambda$), such that $\mathscr{L}_t^\lambda \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, and $\mathbf{L}_t^\lambda$ as:

$$\mathscr{V}_t^\lambda = \begin{cases} 0 & \text{if } \lambda == \text{🫘} \\ \frac{\mathscr{L}_t^\lambda}{\mathbf{L}_t^\lambda} & \text{else} \end{cases}$$

We define the *Seed Gauge Points* of λ during t as:

$$\mathscr{S}_t^\lambda = \begin{cases} \mathscr{C}_t \times \max(\mathscr{V}_t^\lambda) \times \mathbf{L}_t^\lambda & \text{if } \lambda == \text{🫘} \\ \mathscr{L}_t^\lambda & \text{else} \end{cases}$$

Therefore, we define c_t^λ given $\mathbf{C}_t$, $\mathscr{S}_t^\lambda$, $\mathbf{L}_t^\lambda$ and $\mathscr{S}_t$ as:

$$c_t^\lambda = \frac{\mathbf{C}_t \times \mathscr{S}_t^\lambda}{\mathbf{L}_t^\lambda \times \mathscr{S}_t}$$

The number of *Seeds* for a given λ *Deposit* are determined by its PDV when *Deposited* and c_t^λ.

Therefore, the number of *Seeds* during t for a given λ *Deposit Deposited* during i ($C_{t,i}^\lambda$), such that $C_{t,i}^\lambda \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, is:

$$C_{t,i}^\lambda = c_t^\lambda \times L_i^\lambda$$

A *Stalkholder's* total *Seeds* during t (C_t), such that $C_t \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, is:

$$C_t = \sum_{\lambda \in \Lambda} \sum_{i=1}^{t} C_{t,i}^\lambda$$

The number of *Plantable Seeds* associated with a *Stalkholder's* $\eta^{\text{🫘}}$ that can be *Planted* during t to start earning *Grown Stalk* (η_t^c), such that $\eta_t^c \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, is:

$$\eta_t^c = c_t^{\text{🫘}} \times \eta^{\text{🫘}}$$

6 Field

The Pinto peg maintenance mechanism requires the ability to borrow Pinto. The *Field* is the protocol-native credit facility.

Anytime there is *Soil* in the *Field*, any owner of Pinto that are not in the *Silo* can *Sow* (lend) Pinto to the protocol in exchange for *Pods* and become a *Sower*. The *Temperature* is the interest rate on Pinto loans. The *Morning* is the first 5 minutes of each *Season*. The protocol changes the *Soil* and *Temperature* every Q, such that $Q \in \mathbb{Z}^+$, seconds during the *Morning* according to the peg maintenance mechanism.

6.1 Soil

We define *Soil* (S), such that $S \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as the current number of Pinto that can be *Sown* in exchange for *Pods*. 1 is *Sown* in one *Soil*. The protocol permanently removes *Sown* from the Pinto supply.

When the protocol is willing to borrow more Pinto to remove them from the Pinto supply, it creates more *Soil*. The protocol changes the *Minimum Soil* ($S_{t_q}^{\min}$), such that $S_{t_q}^{\min} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, in each interval q, such that $q \in \mathbb{Z}^+$, of the *Morning* of t according to the peg maintenance mechanism. During the *Morning* of each *Season*, a Dutch auction determines the *Minimum Soil*.[7]

6.2 Pods

Pods are the primary debt asset of Pinto. The protocol never defaults on debt. *Pods* automatically *Yield* from *Sown* and never expire.

In the future, when the average price of 1 is above its value target over a *Season*, *Pods Ripen* and become *Harvestable* (redeemable) for 1 at anytime. *Pods Ripen* on a FIFO basis: *Pods Yielded* from Pinto that are *Sown* first *Ripen* into *Harvestable Pods* first. *Pod* holders can *Harvest* their *Harvestable Pods* anytime by calling the `harvest` function. There is no penalty for waiting to *Harvest Pods*.

Pods are transferable. In practice, *Pods* are non-callable zero-coupon bonds with priority for maturity represented as a place in line. The number of *Pods* that *Yield* from *Sown* is determined by the *Temperature*.

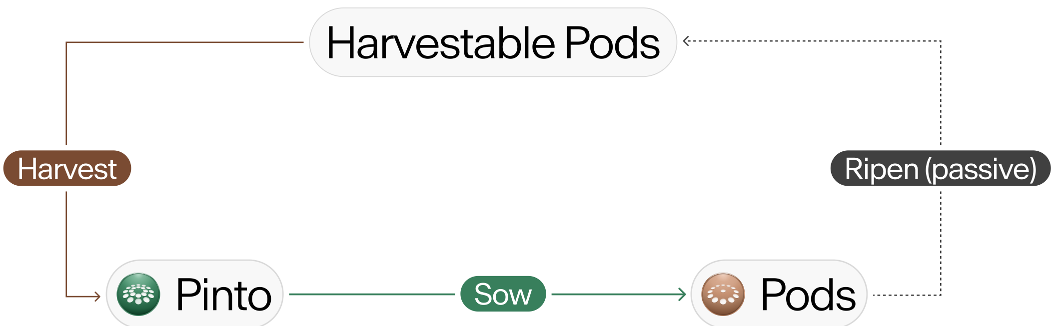

Figure 2: Field

[7] en.wikipedia.org/wiki/Dutch_auction

6.3 Temperature

We define the *Temperature* (h), such that $h \in \mathbb{Z}^+$, as the percentage of additional Pinto ultimately *Harvested* from 1 *Sown*.

The number of *Pods* (d) that *Yield* from a given number of *Sown* (u), such that $d,\ u \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, *Sown* with a given h is:

$$d = u \times \left(1 + \frac{h}{100}\right)$$

The protocol changes the *Maximum Temperature* it is willing to offer each Season (H_t), such that $H_t \in \mathbb{Z}^+$, at the beginning of each *Season* according to the peg maintenance mechanism. During the *Morning* of each *Season*, a Dutch auction determines the *Temperature*.

7 Peg Maintenance

Pinto faces the fundamental limitation that it cannot fix the price of 1 at its value target, but instead must encourage widespread participation in peg maintenance through protocol-native financial incentives. Stability is a function of how frequently and regularly the price of 1 crosses, and the magnitudes of price deviations from, its value target. The protocol attempts to regularly cross the price of 1 over its value target during both long run decreases and increases in demand for Pinto.

The protocol has five primary peg maintenance tools available: (1) increase the Pinto supply, (2) change the *Soil* supply, (3) change the *Maximum Temperature*, (4) change the *Crop Ratio*, and (5) *Flood* (defined below). At the beginning of every *Season*, the protocol evaluates its position (*i.e.*, price, debt level and liquidity level) and current state (*i.e.*, direction and acceleration) with respect to ideal equilibrium, and dynamically adjusts the Pinto supply, *Soil* supply, *Maximum Temperature* and *Crop Ratio* to move closer to ideal equilibrium.

7.1 Ideal Equilibrium

Pinto derive their value from the credit of the protocol and liquidity they trade against. The protocol only fails if it can no longer attract creditors and there is no liquidity trading against Pinto. A healthy level of debt and liquidity attracts creditors and liquidity. Therefore, in addition to the Pinto price, the peg maintenance mechanism considers the protocol debt and liquidity levels (defined below).

The protocol is in ideal equilibrium when the Pinto price and protocol debt and liquidity levels are at their optimal levels, and the demand for *Soil* is steady (defined below). The protocol affects the supply of and demand for Pinto to return to ideal equilibrium in response to the Pinto price, the protocol debt and liquidity levels, and changing demand for *Soil*, by adjusting the Pinto supply, *Soil* supply, *Maximum Temperature* and *Crop Ratio*.

Pinto supply increases and *Soil* supply changes primarily affect the Pinto supply. *Maximum Temperature* and *Crop Ratio* changes primarily affect demand for Pinto. In order to make the proper adjustments, the protocol closely monitors the states of both the Pinto and *Soil* markets.

In practice, maintaining ideal equilibrium is impossible. Deviations from ideal equilibrium along each axis are normal and expected. As the protocol grows, the durations and magnitudes of deviations should decrease.

7.2 Price Level

At the beginning of each *Season*, the protocol calculates a sum of time weighted average shortages and excesses of Pinto across liquidity pools on the *Minting Whitelist* over the previous *Season* ($\Delta B_{\overline{t-1}}$), such that $\Delta B_{\overline{t-1}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$. Liquidity pools can be added to and removed from the *Minting Whitelist* via the *PCM*.

$\Delta B_{\overline{t-1}}$ can be used to infer $P_{\overline{t-1}}$. If there was a time weighted average shortage of Pinto across liquidity pools on the *Minting Whitelist* over the previous *Season* (*i.e.*, $0 < \Delta B_{\overline{t-1}}$), the protocol infers $V < P_{\overline{t-1}}$. If there was a time weighted average excess of Pinto across liquidity pools on the *Minting Whitelist* over the previous *Season* (*i.e.*, $\Delta B_{\overline{t-1}} < 0$), the protocol infers $P_{\overline{t-1}} < V$. If there was neither a time weighted shortage nor excess of Pinto across liquidity pools on the *Minting Whitelist* over the previous *Season* (*i.e.*, $\Delta B_{\overline{t-1}} = 0$), the protocol infers $P_{\overline{t-1}} = V$.

If $0 < \Delta B_{\overline{t-1}}$, the protocol calculates $P_{\overline{t-1}}$ from the pool on the *Minting Whitelist* with the most non-Pinto USD-denominated value at the beginning of the *Season*.

$\Delta B_{\overline{t-1}} = 0$ for each *Season* that contains a *Pause* and *Unpause* (see Appendix).

The protocol requires two P levels to be set: (1) P^*, the optimal price (*i.e.*, $P^* = V$), and (2) P^{upper}, above which the price is considered excessively high, such that $P^*,\ P^{\text{upper}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$. When $P_{\overline{t-1}} \leq P^{\text{upper}}$ and $P_{\overline{t-1}} \neq P^*$ (*i.e.*, not optimal), $P_{\overline{t-1}}$ is considered reasonable.

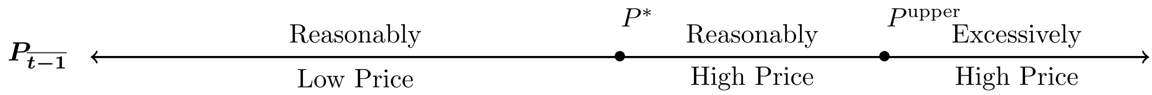

Figure 3: Price Level

7.3 Debt Level

The *Pod Rate* (R^D), such that $R^D \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, represents the protocol debt level relative to the Pinto supply.

The protocol does not consider *Burnt*, *Sown* nor *Unharvestable Pods*, but does consider *Harvestable Pods*, as part of the total Pinto supply.

We define the total Pinto supply (B) for a given total Pinto minted over all *Seasons* (M), such that $B,\ M \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, total *Sown* over all *Seasons* (U), and total *Burnt* over all *Seasons* (N), such that $U,\ N \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as:

$$B = M - (U + N)$$

We define R^D for a given the total number of *Unharvestable Pods* (D), such that $D \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, and B as:

$$R^D = \frac{D}{B}$$

The protocol requires three R^D levels to be set: (1) $R^{D^{\text{lower}}}$, (2) R^{D^*}, an optimal level of debt, and (3) $R^{D^{\text{upper}}}$, above or equal to which debt is considered excessively high, such that R^{D^*}, $R^{D^{\text{upper}}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$. When $R^{D^{\text{lower}}} \leq R^D < R^{D^{\text{upper}}}$ and $R^D \neq R^{D^*}$ (*i.e.*, not optimal), R^D is considered reasonable.

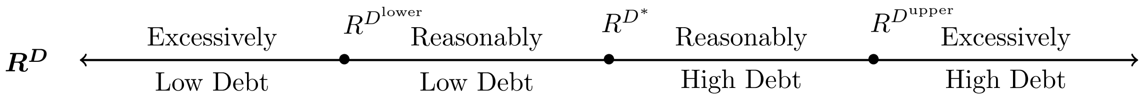

Figure 4: Debt Level

7.4 Liquidity Level

The *Liquidity Rate* (R^W), such that $R^W \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, represents the protocol liquidity level relative to the Pinto supply.

The total non-Pinto USD-denominated value of liquidity in the *Silo* (W), such that $W \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, is a function of $f^{\mathscr{W}^\lambda}$, the function to calculate the time weighted average non-Pinto USD-denominated value in a given number of *Deposited* λ ($f^{\$}(z^\lambda)$), such that $f^{\$}\colon \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^+\} \to \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, and $\mathbf{L}_t^\lambda$.

We define $f^{\$}(z^\lambda)$ for a given inter-block MEV manipulation resistant TWA non-Pinto reserves in the Multi Flow *Pump*[8] of λ from the beginning of the previous *Season* to the current transaction ($\lambda^{\text{SMA}}_{\text{🫘},t-1_0,\text{Ↄ}}$), such that $\lambda^{\text{SMA}}_{\text{🫘},t-1_0,\text{Ↄ}} \in \{j \times 10^{-\text{🫘}} \mid j \in \mathbb{Z}^+\}$, $f^{\$^\lambda}$, and the inter-block MEV manipulation resistant TWA number of λ LP tokens from the beginning of the previous *Season* to the current transaction ($\lambda^{\text{SMA}}_{\lambda,t-1_0,\text{Ↄ}}$), such that $\lambda^{\text{SMA}}_{\lambda,t-1_0,\text{Ↄ}} \in \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^+\}$ as:

$$f^{\$}(z^\lambda) = \frac{z^\lambda \times \lambda^{\text{SMA}}_{\text{🫘},t-1_0,\text{Ↄ}} \times f^{\$^\lambda}(3600)}{\lambda^{\text{SMA}}_{\lambda,t-1_0,\text{Ↄ}}}$$

Therefore, we define W for a given $f^{\mathscr{W}^\lambda}$, $f^{\$}(z^\lambda)$ and $\mathbf{L}_t^\lambda$ as:

$$W = \sum_{\lambda \in \Lambda \setminus \text{🫘}} f^{\mathscr{W}^\lambda} \times f^{\$}(\mathbf{L}_t^\lambda)$$

Therefore, we define R^W for a given W and B as:

$$R^W = \frac{W}{B}$$

[8] Any italicized terms not defined herein are defined by Pinto Exchange.

The protocol requires three R^W levels to be set: (1) $R^{W^{\text{lower}}}$, below which liquidity is considered excessively low, (2) R^{W^*}, an optimal level of liquidity, and (3) $R^{W^{\text{upper}}}$, above or equal to which liquidity is considered excessively high, such that $R^{W^{\text{lower}}}, R^{W^*}, R^{W^{\text{upper}}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$. When $R^{W^{\text{lower}}} \leq R^W < R^{W^{\text{upper}}}$ and $R^W \neq R^{W^*}$ (*i.e.*, not optimal), R^W is considered reasonable.

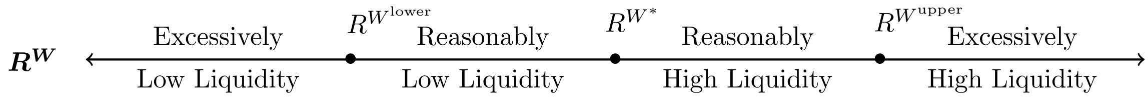

Figure 5: Liquidity Level

7.5 Position

The position of the protocol with respect to ideal equilibrium can be represented on a graph with axes R^D, R^W and P, and ideal equilibrium at the origin (R^{D^*}, R^{W^*}, P^*). The current state of the protocol is determined in part by the position of the protocol with respect to ideal equilibrium.

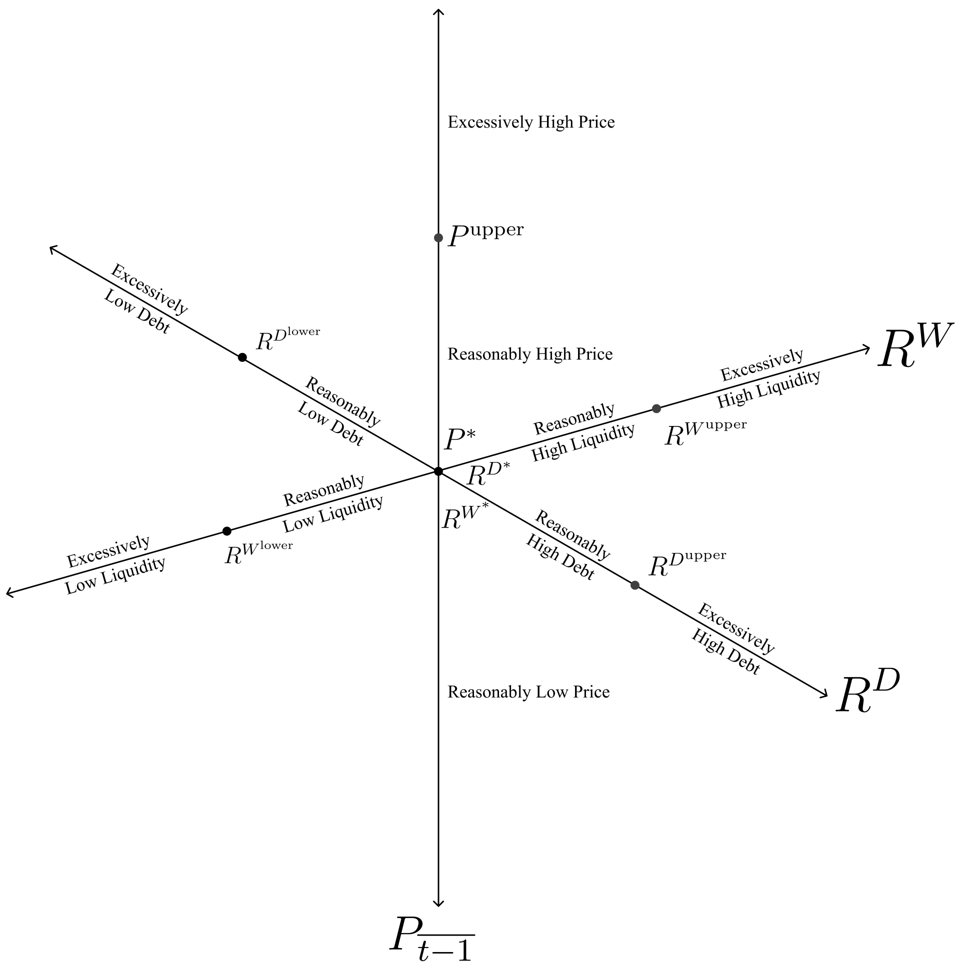

Figure 6: Position

7.6 Direction

The position of the protocol with respect to ideal equilibrium changes at the beginning of each *Season*. The current state of the protocol with respect to ideal equilibrium is determined in part by the direction of this change.

The direction of change in position of the protocol at the beginning of t is considered either toward or away from ideal equilibrium, based on R^D_{t-1} and $P_{\overline{t-1}}$. When $V < P_{\overline{t-1}}$ (*i.e.*, $0 < \Delta B_{\overline{t-1}}$), debt is paid back; when $P_{\overline{t-1}} \leq V$ (*i.e.*, $\Delta B_{\overline{t-1}} \leq 0$), debt can only increase or remain constant.

Therefore, when $R^{D^*} < R^D_{t-1}$ (*i.e.*, there was more than optimal debt):

- If $V < P_{\overline{t-1}}$, the protocol moves toward ideal equilibrium; and
- If $P_{\overline{t-1}} \leq V$, the protocol moves away from ideal equilibrium.

When $R^D_{t-1} \leq R^{D^*}$ (*i.e.*, there was less than optimal or optimal debt):

- If $V < P_{\overline{t-1}}$, the protocol moves away from ideal equilibrium; or
- If $P_{\overline{t-1}} \leq V$, the protocol moves toward ideal equilibrium.

		$\boldsymbol{R^D_{t-1}}$			
	Direction	**Excessively Low Debt**	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
$\boldsymbol{P_{\overline{t-1}}}$	$\boldsymbol{V < P_{\overline{t-1}}}$	Away From	Away From	Toward	Toward
	$\boldsymbol{P_{\overline{t-1}} \leq V}$	Toward	Toward	Away From	Away From

Figure 7: Direction

7.7 Acceleration

The current state of the protocol with respect to ideal equilibrium is also determined by the rate of change of direction of the protocol at the beginning of each *Season* (*i.e.*, its acceleration).

The acceleration of the protocol is considered decelerating, steady or accelerating, based on $P_{\overline{t-1}}$, R^D_{t-1} and changing demand for *Soil*. Demand for *Soil* is considered decreasing, steady or increasing.

When demand for *Soil* is decreasing:

- If $V < P_{\overline{t-1}}$, the protocol is decelerating; and
- If $P_{\overline{t-1}} \leq V$, the protocol is accelerating.

When demand for *Soil* is steady, the protocol is steady.

When demand for *Soil* is increasing:

- If $V < P_{\overline{t-1}}$, the protocol is accelerating; and
- If $P_{\overline{t-1}} \leq V$, the protocol is decelerating.

		Demand for Soil		
	Acceleration	**Decreasing Demand**	**Steady Demand**	**Increasing Demand**
$\boldsymbol{P_{\overline{t-1}}}$	$\boldsymbol{V < P_{\overline{t-1}}}$	Decelerating	Steady	Accelerating
	$\boldsymbol{P_{\overline{t-1}} \leq V}$	Accelerating	Steady	Decelerating

Figure 8: Acceleration

7.8 Demand for Soil

In order to properly classify its acceleration, the protocol must accurately measure changing demand for *Soil*.

The number of *Sown* each *Season* (u_t), such that $u_t \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, indicates demand for *Soil* over the course of that *Season*. The rate of change of u_t from *Season* to *Season* ($\frac{\partial u_t}{\partial t}$), such that $\frac{\partial u_t}{\partial t} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, indicates changing demand for *Soil*.

We define $\frac{\partial u_t}{\partial t}$ over the previous two *Seasons*, u_{t-1} and u_{t-2}, respectively, as:

$$\frac{\partial u_t}{\partial t} = \frac{u_{t-1}}{u_{t-2}}$$

The protocol requires two $\frac{\partial u_t}{\partial t}$ levels to be set: (1) $\frac{\partial u_t}{\partial t}^{\text{lower}}$, below which demand for *Soil* is considered decreasing, and (2) $\frac{\partial u_t}{\partial t}^{\text{upper}}$, above or equal to which demand for *Soil* is considered increasing, such that $\frac{\partial u_t}{\partial t}^{\text{lower}}, \frac{\partial u_t}{\partial t}^{\text{upper}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$. When $\frac{\partial u_t}{\partial t}^{\text{lower}} \leq \frac{\partial u_t}{\partial t} < \frac{\partial u_t}{\partial t}^{\text{upper}}$, demand for *Soil* is considered steady.

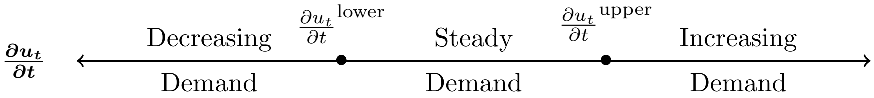

Figure 9: Soil Demand Changes From $\frac{\partial u_t}{\partial t}$

However, when Pinto are *Sown* in all *Soil* in a *Season* (defined as $S_t^{\text{end}} \leq S^{\text{all}}$), $\frac{\partial u_t}{\partial t}$ can inaccurately measure changing demand for *Soil*. The first time Pinto are *Sown* in all *Soil* in a *Season*, after one or more *Seasons* where Pinto were not *Sown* in all *Soil*, demand for *Soil* is considered increasing. When Pinto are *Sown* in all *Soil* in consecutive *Seasons* (*i.e.*, $t-1$ and $t-2$), the difference in time it took for the Pinto to be *Sown* in all *Soil* over the previous two *Seasons* (ΔE_t^u), such that $\Delta E_t^u \in \mathbb{Z}$, can provide a more accurate measurement.

In order to measure ΔE_t^u, the protocol logs the time of the first *Sow* such that Pinto are *Sown* in all *Soil* in each *Season* ($\Delta E_t^{u^{\text{first}}}$), such that $\Delta E_t^{u^{\text{first}}} \in \mathbb{N}$, as the difference between the Base timestamp of the first *Sow* in t such that there is at most one *Soil* ($E_t^{u^{\text{first}}}$) and E_Ξ.

We define $\Delta E_t^{u^{\text{first}}}$ for a given $E_t^{u^{\text{first}}}$ and E_Ξ as:

$$\Delta E_t^{u^{\text{first}}} = E_t^{u^{\text{first}}} - E_\Xi$$

If Pinto were *Sown* in all *Soil* in the first 10 minutes of the previous *Season* (*i.e.*, $\Delta E_{t-1}^{u^{\text{first}}} < \Delta E_{t-1}^{u^{\text{first}^{\max}}}$), demand for *Soil* is considered increasing. If Pinto were *Sown* in all *Soil* in both $t-1$ and $t-2$, but $\Delta E_{t-1}^{u^{\text{first}^{\max}}} \leq \Delta E_{t-1}^{u^{\text{first}}}$, at the beginning of t the protocol compares $\Delta E_{t-1}^{u^{\text{first}}}$ with $\Delta E_{t-2}^{u^{\text{first}}}$ to calculate ΔE_t^u.

We define ΔE_t^u for a given $\Delta E_{t-1}^{u^{\text{first}}}$ and $\Delta E_{t-2}^{u^{\text{first}}}$ as:

$$\Delta E_t^u = \Delta E_{t-2}^{u^{\text{first}}} - \Delta E_{t-1}^{u^{\text{first}}}$$

If the above condition is met, changing demand for *Soil* is measured by ΔE_t^u. the protocol requires two ΔE_t^u levels to be set: (1) $\Delta E_t^{u^{\text{lower}}}$, below which demand for *Soil* is considered decreasing, and (2) $\Delta E_t^{u^{\text{upper}}}$, above or equal to which demand for *Soil* is considered increasing, such that $\Delta E_t^{u^{\text{lower}}}, \Delta E_t^{u^{\text{upper}}} \in \mathbb{Z}$. When $\Delta E_t^{u^{\text{lower}}} \leq \Delta E_t^u < \Delta E_t^{u^{\text{upper}}}$, demand for *Soil* is considered steady.

Thus, the protocol measures changing demand for *Soil*.

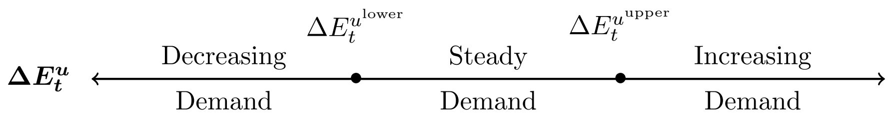

Figure 10: Soil Demand Changes From ΔE_t^u

7.9 Current State

We define the current state of the protocol with respect to ideal equilibrium as the combination of its direction and acceleration with respect to ideal equilibrium. With two potential directions and three potential accelerations, the protocol has six potential current states:

- Accelerating away from ideal equilibrium;
- Steady away from ideal equilibrium;
- Decelerating away from ideal equilibrium;
- Accelerating toward ideal equilibrium;
- Steady toward ideal equilibrium; and
- Decelerating toward ideal equilibrium.

		Acceleration		
	Current State	**Decelerating**	**Steady**	**Accelerating**
Direction	**Away From**	Decelerating Away From	Steady Away From	Accelerating Away From
Direction	**Toward**	Decelerating Toward	Steady Toward	Accelerating Toward

Figure 11: Current State

7.10 Optimal State

An optimal state of the protocol is an optimal current state and determined by R^D_{t-1}.

We define an optimal state of the protocol as accelerating toward ideal equilibrium, or either steady or decelerating toward ideal equilibrium. When R^D_{t-1} is excessively high or low, the optimal state is accelerating toward ideal equilibrium. When R^D_{t-1} is reasonably high or low, the optimal state is either steady or decelerating toward ideal equilibrium.

	R^D_{t-1}			
Optimal State	**Excessively Low Debt**	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
	Accelerating Toward	Steady or Decelerating Toward	Steady or Decelerating Toward	Accelerating Toward

Figure 12: Optimal State

7.11 Pinto Supply

At the beginning of each *Season*, if $V < P_{\overline{t-1}}$, the protocol increases the Pinto supply based on $\Delta B_{\overline{t-1}}$ in addition to the reward for successfully calling the `gm` function. Up to 48.5% of the additional Pinto supply increase is used to pay off *Pods*, up to 3% of the additional Pinto supply increase is distributed to the Pinto development budget[9] or to repay Beanstalk debt[10], and the remainder is distributed to *Stalkholders*. If the Pinto supply is less than 1 billion, 3% is distributed to the Pinto development budget. If the Pinto supply is greater than 1 billion, up to 3% is distributed to repay Beanstalk debt.

At the beginning of each *Season*, the protocol mints m_t Pinto, such that $m_t \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

We define m_t for a given $\Delta B_{\overline{t-1}}$ and a_t as:

$$m_t = \max(0,\ \Delta B_{\overline{t-1}}) + a_t$$

The distribution of the additional mint is dependent on $\Delta B_{\overline{t-1}}$, D and B. If $0 < \frac{\Delta B_{\overline{t-1}} \times 485}{1000} < D$ (*i.e.*, there are at least $\frac{\Delta B_{\overline{t-1}} \times 485}{1000}$ *Unharvestable Pods*), $\frac{\Delta B_{\overline{t-1}} \times 485}{1000}$ *Pods Ripen* and become *Harvestable*, up to $\frac{\Delta B_{\overline{t-1}} \times 30}{1000}$ newly minted Pinto are distributed to the Pinto development budget or to repay Beanstalk debt, and the remaining newly minted Pinto are distributed to *Stalkholders*. If $0 < D < \frac{\Delta B_{\overline{t-1}} \times 485}{1000}$ (*i.e.*, there are less *Unharvestable Pods* than $\frac{\Delta B_{\overline{t-1}} \times 485}{1000}$), D *Pods Ripen* and become *Harvestable*, up to $\frac{\Delta B_{\overline{t-1}} \times 30}{1000}$ newly minted Pinto are distributed to the Pinto development budget or to repay Beanstalk debt, and the remaining newly minted Pinto are distributed to *Stalkholders*.

[9] basescan.org/address/0xb0cdb715D8122bd976a30996866Ebe5e51bb18b0
[10] pinto.money/repay-beanstalk-debt

Therefore, the number of *Pods* that *Ripen* and become *Harvestable* at the beginning of each *Season* (ΔD_t), such that $\Delta D_t \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, is:

$$\Delta D_t = \min\left(\max\left(0,\ \frac{\Delta B_{\overline{t-1}}\ \times\ 485}{1000}\right),\ D\right)$$

7.12 Soil Supply

Pinto is willing to issue debt every *Season*. When $V \leq P_{\overline{t-1}}$, the *Soil* supply is based on (1) the number of *Pods* that *Ripen* and became *Harvestable* at the beginning of the *Season*, (2) the *Temperature* in each interval of the *Morning* of t (h_{t_q}), such that $h_{t_q} \in \mathbb{Z}^+$, and (3) R^D_{t-1}. When $P_{\overline{t-1}} < V$, the *Soil* supply is also based on $\Delta B_{\overline{t-1}}$ and the sum of shortages and excesses of Pinto across liquidity pools on the *Minting Whitelist* calculated using the inter-block MEV manipulation resistant instantaneous reserves in Multi Flow *Pump* ($\Delta B^{\text{EMA}}_{\Xi-1}$), such that $\Delta B^{\text{EMA}}_{\Xi-1} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$.

We define $S^{\min}_{t_q}$ for a given ΔD_t, h_{t_q}, $R^{D^{\text{upper}}}$, R^D_{t-1} and $R^{D^{\text{lower}}}$ as:

$$S^{\min}_{t_q} = \begin{cases} \dfrac{0.5 \times \Delta D_t}{1 + \frac{h_{t_q}}{100}} & \text{if } R^{D^{\text{upper}}} \leq R^D_{t-1} \\ \dfrac{\Delta D_t}{1 + \frac{h_{t_q}}{100}} & \text{if } R^{D^{\text{lower}}} < R^D_{t-1} \\ \dfrac{1.5 \times \Delta D_t}{1 + \frac{h_{t_q}}{100}} & \text{else} \end{cases}$$

The protocol calculates the *Maximum Soil* ($S^{\max}_{t_q}$), such that $S^{\max}_{t_q} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, in each interval q of the *Morning* of t for a given $\Delta B_{\overline{t-1}}$, $\Delta B^{\text{EMA}}_{\Xi-1}$ and $S^{\min}_{t_q}$ as:

$$S^{\max}_{t_q} = \begin{cases} -\max(\Delta B^{\text{EMA}}_{\Xi-1}, \Delta B_{\overline{t-1}}) & \text{if } \Delta B_{\overline{t-1}} < 0 \ \&\&\ \Delta B^{\text{EMA}}_{\Xi-1} < 0 \\ -\Delta B_{\overline{t-1}} & \text{if } \Delta B_{\overline{t-1}} < 0 \ \&\&\ \Delta B^{\text{EMA}}_{\Xi-1} > 0 \\ S^{\min}_{t_q} & \text{else} \end{cases}$$

7.13 Temperature

The protocol attempts to regularly cross the price of 1 over its value target during long run decreases and increases in demand for Pinto by adjusting the *Maximum Temperature* in an attempt to maintain an optimal state, or to move from its current state into an optimal state. During the *Morning* of each *Season*, a Dutch auction determines the *Temperature* increases at the beginning of each interval.

7.13.1 Maximum Temperature

The *Maximum Temperature* change at the beginning of t is determined by R^D_{t-1}, R^W_{t-1}, $P_{\overline{t-1}}$ and changing demand for *Soil*.

The protocol changes the *Maximum Temperature* in both a relative (H^m), such that $H^m \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, and absolute manner ($H^b$), such that $H^b \in \mathbb{Z}$, at the beginning of t as:

$$H_t = \max(H^m \times H_{t-1} + H^b, 1)$$

H^m and H^b are shown in Figures 13, 14, 15 and 16 as (H^m, H^b).

$P_{\overline{t-1}}$ & Demand for Soil Changes	Excessively Low R^W_{t-1}	R^D_{t-1}: Excessively Low Debt	R^D_{t-1}: Reasonably Low Debt	R^D_{t-1}: Reasonably High Debt	R^D_{t-1}: Excessively High Debt
$P_{\overline{t-1}} > P^{\text{upper}}$	Increasing	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} > V$	Increasing	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} \leq V$	Increasing	(1, 1)	(1, 1)	(1, 0)	(1, 0)
	Steady	(1, 3)	(1, 3)	(1, 1)	(1, 1)
	Decreasing	(1, 3)	(1, 3)	(1, 3)	(1, 3)

Figure 13: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with an Excessively Low R^W_{t-1}

$P_{\overline{t-1}}$ & Demand for Soil Changes	Reasonably Low R^W_{t-1}	R^D_{t-1}: Excessively Low Debt	R^D_{t-1}: Reasonably Low Debt	R^D_{t-1}: Reasonably High Debt	R^D_{t-1}: Excessively High Debt
$P_{\overline{t-1}} > P^{\text{upper}}$	Increasing	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} > V$	Increasing	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} \leq V$	Increasing	(1, 1)	(1, 1)	(1, 0)	(1, 0)
	Steady	(1, 3)	(1, 3)	(1, 1)	(1, 1)
	Decreasing	(1, 3)	(1, 3)	(1, 3)	(1, 3)

Figure 14: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with a Reasonably Low R^W_{t-1}

Reasonably High R^W_{t-1}		R^D_{t-1} **Excessively Low Debt**	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, 1)	(1, 1)	(1, 0)	(1, 0)
	Steady	(1, 3)	(1, 3)	(1, 1)	(1, 1)
	Decreasing	(1, 3)	(1, 3)	(1, 3)	(1, 3)

$P_{\overline{t-1}}$ & Demand for Soil Changes

Figure 15: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with a Reasonably High R^W_{t-1}

Excessively High R^W_{t-1}		R^D_{t-1} **Excessively Low Debt**	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -3)	(1, -3)	(1, -3)	(1, -3)
	Steady	(1, -3)	(1, -3)	(1, -1)	(1, -1)
	Decreasing	(1, -1)	(1, -1)	(1, 0)	(1, 0)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, 1)	(1, 1)	(1, 0)	(1, 0)
	Steady	(1, 3)	(1, 3)	(1, 1)	(1, 1)
	Decreasing	(1, 3)	(1, 3)	(1, 3)	(1, 3)

$P_{\overline{t-1}}$ & Demand for Soil Changes

Figure 16: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with an Excessively High R^W_{t-1}

7.13.2 Morning

The *Temperature* increases logarithmically in each interval of the *Morning* of t based on H_t, the number of intervals in the *Morning* ($\mathfrak{Q}$), such that $\mathfrak{Q} \in \mathbb{Z}^+$, and a control variable (σ), such that $\sigma \in \{j \times 10^{-\infty} \mid j \in \mathbb{Z}^+\}$, as:

$$h_{t_q} = \begin{cases} 1 & \text{if } q = 0 \\ \max(H_t * \log_{\mathfrak{Q}\sigma+1}(q\sigma + 1),\ 1) & \text{if } 0 < q < \mathfrak{Q} \\ H_t & \text{else} \end{cases}$$

Thus, the protocol changes the *Temperature* in an attempt to regularly cross the price of 1 over its value target during long run decreases and increases in demand for Pinto.

7.14 Crop Ratio

The protocol attempts to regularly cross the price of 1 over its value target during long run decreases and increases in demand for Pinto by adjusting the *Crop Ratio*. In practice, the protocol adjusts the *Crop Ratio* by changing the *Crop Scalar*. The *Crop Scalar* change at the beginning of t is determined by R^D_{t-1}, R^W_{t-1}, $P_{\overline{t-1}}$ and changing demand for *Soil*.

The protocol changes the *Crop Scalar* in both a relative ($\mathscr{X}^m$), such that $\mathscr{X}^m \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, and absolute manner ($\mathscr{X}^b$), such that $\mathscr{X}^b \in \{j \times 10^{-18} \mid j \in \mathbb{Z}\}$, at the beginning of t as:

$$\mathscr{X}_t = \min(\max(\mathscr{X}^m \times \mathscr{X}_{t-1} + \mathscr{X}^b, 0), 1)$$

$\mathscr{X}^m$ and $\mathscr{X}^b$ are shown in Figures 17, 18, 19 and 20 as ($\mathscr{X}^m$, $\mathscr{X}^b$).

$P_{\overline{t-1}}$ & Demand for Soil Changes	Excessively Low R^W_{t-1}	R^D_{t-1}: Excessively Low Debt	R^D_{t-1}: Reasonably Low Debt	R^D_{t-1}: Reasonably High Debt	R^D_{t-1}: Excessively High Debt
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, –0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)

Figure 17: Crop Scalar Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with an Excessively Low R^W_{t-1}

$P_{\overline{t-1}}$ & Demand for Soil Changes	Reasonably Low R^W_{t-1}	R^D_{t-1}			
		Excessively Low Debt	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.01)	(1, -0.01)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.01)	(1, -0.01)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.01)	(1, -0.01)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, 0.01)	(1, 0.01)
	Steady	(1, -0.5)	(1, -0.5)	(1, 0.01)	(1, 0.01)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, 0.01)	(1, 0.01)

Figure 18: Crop Scalar Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with a Reasonably Low R^W_{t-1}

$P_{\overline{t-1}}$ & Demand for Soil Changes	Reasonably High R^W_{t-1}	R^D_{t-1}			
		Excessively Low Debt	**Reasonably Low Debt**	**Reasonably High Debt**	**Excessively High Debt**
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
	Steady	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
	Decreasing	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, 0.01)	(1, 0.01)	(1, 0.01)	(1, 0.01)
	Steady	(1, 0.01)	(1, 0.01)	(1, 0.01)	(1, 0.01)
	Decreasing	(1, 0.01)	(1, 0.01)	(1, 0.01)	(1, 0.01)

Figure 19: Crop Scalar Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with a Reasonably High R^W_{t-1}

Excessively High R^W_{t-1}		R^D_{t-1}: Excessively Low Debt	Reasonably Low Debt	Reasonably High Debt	Excessively High Debt
$P_{\overline{t-1}}$ & Demand for Soil Changes					
$P_{\overline{t-1}} > P^{\text{upper}}$	**Increasing**	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Steady	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
	Decreasing	(1, -0.5)	(1, -0.5)	(1, -0.5)	(1, -0.5)
$P_{\overline{t-1}} > V$	**Increasing**	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
	Steady	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
	Decreasing	(1, -0.01)	(1, -0.01)	(1, -0.01)	(1, -0.01)
$P_{\overline{t-1}} \leq V$	**Increasing**	(1, 0.01)	(1, 0.01)	(1, 0.02)	(1, 0.02)
	Steady	(1, 0.01)	(1, 0.01)	(1, 0.02)	(1, 0.02)
	Decreasing	(1, 0.01)	(1, 0.01)	(1, 0.02)	(1, 0.02)

Figure 20: Crop Scalar Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1} with an Excessively High R^W_{t-1}

7.15 Flood

During long run increases in demand for Pinto, at the beginning of each *Season*, the protocol sells newly minted Pinto on the open market if increasing the Pinto supply, lowering the *Maximum Temperature* and lowering the *Crop Ratio* have not crossed the average nor current prices of 1 over its value target.

If $V < P_{\overline{t-1}}$ and $R^D_{t-1} < R^{D^{\text{lower}}}$, it is *Raining*. If it has *Rained* for F, such that $F \in \mathbb{Z}^+$, consecutive full *Seasons* and at the beginning of the next *Season* it is still *Raining*, it *Floods*.

At the beginning of each *Season* in which it *Floods*, the protocol mints additional Pinto and sells them directly in the pools on the *Flood Whitelist* with the highest shortage of Pinto at the end of the previous *Season*. Proceeds from the sale are distributed to *Stalkholders* at the beginning of t in proportion to their *Stalk* holdings when it began to *Rain* that they still have. At the beginning of each *Season* in which it Floods, up to 0.1% of the Pinto supply worth of *Pods* that grew from Pinto *Sown* before it began to *Rain Ripen* and become *Harvestable*.

The number of Pinto that are minted and sold to return the price of 1 to its value target is calculated from the sum of differences between the optimal number of Pinto and the number of Pinto in each liquidity pool on the *Flood Whitelist* in the current transaction ($\Delta B_{Ↄ}$), such that $\Delta B_{Ↄ} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$. For each marginal Pinto minted to be sold, the protocol sells it into the pool with the highest shortage of Pinto until all Pinto are sold.

In a *Flood*, m_t for a given number of *Unharvestable Pods* that grew prior to the *Flood* (D_γ), the maximum percent of the Pinto supply worth of *Pods* that grew from Pinto *Sown* before it began to *Rain* that become *Harvestable* (D^F_γ), such that $D_\gamma,\ D^F_\gamma \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, B, a_t, $\Delta B_{\overline{t-1}}$ and $\Delta B_{Ↄ}$ is:

$$m_t = \min(D_\gamma, B \times D^F_\gamma) + a_t + \Delta B_{\overline{t-1}} + \Delta B_{Ↄ}$$

Thus, the protocol attempts to regularly cross the price of 1 over its value target during both long run increases and decreases in demand for Pinto.

8 Toolshed

The *Toolshed* offers a suite of tools that decrease friction to using the protocol and participating in peg maintenance.

8.1 Pod Market

The *Pod Market* is a peer-to-peer marketplace that allows *Pods* to be bought and sold in a trustless fashion without trading fees. Specifications of the *Pod Market* are outside the scope of this whitepaper. For information on the *Pod Market*, refer to the Appendix.

8.2 Depot

Complex interactions with Base-native protocols can be tedious, cumbersome and expensive. The *Depot* facilitates complex, gas-efficient interactions with other Base-native protocols in a single transaction. Any external protocol with a *Pipeline* to the *Depot* can be used through the protocol in a single transaction. *Pipelines* to the *Depot* can be added and removed via the *PCM*. The specifications of specific *Pipelines* are outside the scope of this whitepaper. For information on the *Depot*, refer to the Appendix.

8.3 Farm Balances

Farm Balances are assets stored on the *Farm* that are not *Deposited* in the *Silo* or locked in *Pod Orders*. *Farm Balances* enable the protocol to store any ERC-20 token on behalf of a user.

All interactions with the protocol that use a user's ERC-20 tokens can use tokens from both *Farm* and *Wallet Balances*.

8.4 Tractor

Tractor is a peer-to-peer transaction marketplace that allows third parties to perform pre-authorized actions through the protocol on behalf of a user. Specifications of *Tractor* are outside the scope of this whitepaper.

9 Economics

The protocol is designed from economic first principles to increase trustlessness, stability and liquidity over time.

9.1 Ownership Concentration

A design that lowers the Gini coefficient[11] of Pinto and *Stalk* over time is essential to censorship resistance.

Older *Deposits* have their *Stalk* diluted relative to newer *Deposits* every *Season*. Therefore, newly minted Pinto are more widely distributed over time.

The protocol does not require a pre-mine. The first 1000 Pinto are created when the `init` function is called to deploy the protocol.

9.2 Strong Credit

Pinto is credit based and only fails if it can no longer attract creditors. A reasonable level of debt, strong credit history, competitive interest rate and reasonable level of liquidity attract creditors.

The protocol attempts to change the *Temperature* and *Crop Ratio* to return R^D to R^{D^*} while regularly crossing the price of 1 over its value target. The protocol acts more aggressively when R^D is excessively high or low.

The protocol never defaults on debt and is willing to issue *Pods* every *Season*.

9.3 Deep Liquidity

The value of is derived from the liquidity they trade against. Reasonable levels of debt and liquidity attract more liquidity.

The protocol attempts to change the *Temperature* and *Crop Ratio* to return R^W to R^{W^*} while regularly crossing the price of 1 over its value target. The protocol acts more aggressively when R^W is excessively high or low.

9.4 Marginal Rate of Substitution

There are a wide variety of opportunities Pinto has to compete with for creditors and liquidity providers. Therefore, the protocol does not define an optimal *Temperature* or *Crop Ratio*, but instead adjusts them to move closer to ideal equilibrium.

9.5 Low Friction

Minimizing the cost of using Pinto and barriers to the *Farm* maximizes utility for users and appeal to creditors. The *Depot* realizes the benefits of composability on Base.

The FIFO *Pod Harvest* schedule allows smaller *Sowers* to participate in peg maintenance and decreases the benefit of large scale price manipulation. The combination of non-expiry, the FIFO *Harvest* schedule, the Dutch auction, transferability, a liquid secondary market and *Tractor* enables *Sowers* to *Sow* Pinto as efficiently as possible. By maximizing the efficiency of the *Soil* market, Pinto minimizes its cost to attract creditors, the durations and magnitudes of price deviations below its value target, and excess *Pod* issuance.

[11] wikipedia.org/wiki/Gini_coefficient

9.6 Equilibrium

Equilibrium is a state of equivalent marginal quantity supplied and demanded. The protocol affects the supply of and demand for Pinto in attempt to regularly cross the equilibrium price of 1 over its value target.

While the protocol can arbitrarily increase the Pinto supply when the equilibrium price of 1 is above its value target, it cannot arbitrarily decrease the Pinto supply when the equilibrium price of 1 is below it. The protocol relies on the codependence between the equilibria of Pinto and *Soil* to work around this limitation.

9.7 Incentives

Protocol-native financial incentives are designed to increase trustlessness, stability and liquidity over time by coordinating independently financially motivated actors (*i.e*, *Stalkholders* and *Sowers*).

The *Stalk* and *Seed Gauge Systems* incentivize (1) *Depositing* assets in the *Silo*, (2) leaving assets *Deposited* in the *Silo* continuously by creating opportunity cost to *Withdraw* assets from the *Silo*, (3) adding value to liquidity pools with Pinto by rewarding at least as many *Seeds* to at least one *Deposited* LP token as *Deposited* , and (4) returning the price of 1 to its value target by allowing *Conversions* within the *Silo* without forfeiting *Stalk*.

When $V < P_{\bar{t}}$, there is an incentive to buy Pinto to earn a portion of the upcoming Pinto seigniorage. This is exacerbated when R^D is lower and R^W is higher. The *Germination* period reduces this incentive significantly. The combination of the commitment to (1) automatically return the price of 1 to its value target and (2) distribute proceeds from the sale to current *Stalkholders* based on *Stalk* ownership when it began to *Rain*, removes this incentive entirely during *Seasons* where it is *Raining*.

Thus, the protocol is designed to consistently increase trustlessness, stability and liquidity over time.

10 Risk

There are numerous risks associated with Pinto.[12] This is not an exhaustive list.

The protocol code base and peg maintenance mechanism are novel. Neither had been tested in the "real world" prior to the initial protocol deployment. Portions of the Pinto code base are unaudited.[13] The open source nature of Pinto means that anyone can take advantage of any bugs, flaws or deficiencies in Pinto and launch identical or very similar low-volatility money implementations.

A trustless implementation of Pinto has four external dependencies:

(1) A trustless computer network that supports composability and both fungible and semi-fungible token standards (*e.g.*, Base and the ERC-20 and ERC-1155 Standards, respectively);

(2) A trustless exchange protocol with an inter-block MEV manipulation resistant oracle that runs on (1) (*e.g.*, Pinto Exchange[14] and Multi Flow, respectively);

(3) A liquid, trustless network-native asset with endogenous value (*e.g.*, ETH); and

(4) A trustless oracle for the value of (3) in V, or a non-network-native exogenous value convertible stablecoin protocol native to (1) that offers convertibility to its non-network-native exogenous value collateral (*e.g.*, USDC, USDT) that trades on (2) against (3) with sufficient liquidity.

[12] docs.pinto.money/disclosures
[13] docs.pinto.money/resources/audits
[14] pinto.exchange

The current implementation of Pinto lacks (4) and has three additional external dependencies:

(5) The non-network-native assets in each λ on the *Deposit Whitelist*;

(6) Non-network-native data feeds that facilitate reading inter-block MEV manipulation resistant USD-denominated prices of (3) and (5) (*e.g.*, the ETH/USD[15] and SOL/USD[16] Chainlink data feeds); and

(7) Pipeline[17], in order to facilitate complex, gas-efficient interactions with other Base-native protocols in a single transaction.

To date, the Base network is one of the most developed and trust-minimized smart contract platforms and has an active community. The ERC-20 and ERC-1155 Standards are the most widely used fungible and semi-fungible token standards, respectively. ETH is the most censorship resistant and liquid asset on the Base network. Chainlink is the most widely used oracle network on Base. In general, open source protocols with large amounts of value on them are high value targets for exploits. Long track records indicate security.

The code bases of Pinto Exchange, Multi Flow and Pipeline are novel. They had not been tested in the "real world" prior to their initial deployments. Their open source nature means anyone can exploit any bugs, flaws or deficiencies. Although Pinto Exchange, Multi Flow and Pipeline have been audited[18], it is no guarantee of security.

Pinto inherits the security of the Base network, ERC-20 Standard, ERC-1155 Standard, Pinto Exchange, Multi Flow, non-Pinto assets in each λ on the *Deposit Whitelist*, Chainlink and Pipeline.

The Pinto price oracle contains exposure to risk related to the centralized nature of Chainlink. There is no guarantee the node operators for the Chainlink data feeds used by assets on the *Deposit Whitelist* report price data accurately. The protocol assumes the accuracy of these data feeds.

11 Future Work

Pinto is a work in progress. The following are potential improvements that can be incorporated into Pinto as one or more *Pinto Improvements* (see Appendix):

- Implement a *Fork Migration System* (see Appendix);
- Add a mechanism to pay back old Beanstalk holders after the Pinto supply reaches 1 billion;
- Change parameters until 2 weeks after the first time the Pinto supply reaches 500M (*e.g.*, $\mathscr{T}$, R^D and R^W thresholds, Λ, $\mathscr{B}^{\lambda^*}$, etc.);
- The price oracle can be replaced with a trustless one; and
- Contracts can be made immutable.

[15] data.chain.link/feeds/base/base/eth-usd
[16] data.chain.link/feeds/base/base/sol-usd
[17] evmpipeline.org
[18] github.com/BeanstalkFarms/Beanstalk-Audits#ecosystem-reports

12 Appendix

12.1 Current Parameters

The following are the current parameters of Pinto:

12.1.1 Sun

- $☼ = 10^6$ (*i.e.*, 1)

12.1.2 Silo

- $\overline{\mathbf{C}}^{\circ} = 3 \times 10^{12}$ (*i.e.*, 3)
- $\overline{\mathbf{C}}^{\min} = 2 \times 10^{12}$ (*i.e.*, 2)
- $\mathscr{C}^{\max} = 10^{18}$ (*i.e.*, 100%)
- $\mathscr{C}^{\min} = 5 \times 10^{17}$ (*i.e.*, 50%)
- $\mathscr{C}^{\text{rain}} = 5 \times 10^{17}$ (*i.e.*, 50%)
- $\mathscr{C}_1 = \frac{2}{3} \times 10^{18}$ (*i.e.*, $66.\overline{7}$%)
- $\mathscr{T} = 4.32 \times 10^3$ (*i.e.*, 4320 *Seasons* or 6 months)

12.1.3 Field

- $H_1 = 1$ (*i.e.*, 1%)
- $Q = 1.2 \times 10$ (*i.e.*, 12)

12.1.4 Peg Maintenance

- $V = 10^6$ (*i.e.*, \$1)
- $P^* = 10^6$ (*i.e.*, \$1)
- $P^{\text{upper}} = 1.05 \times 10^6$ (i.e., \$1.05)
- $R^{D^{\text{lower}}} = 5 \times 10^4$ (*i.e.*, 5%)
- $R^{D^*} = 1.5 \times 10^5$ (*i.e.*, 15%)
- $R^{D^{\text{upper}}} = 2.5 \times 10^5$ (*i.e.*, 25%)
- $R^{W^{\text{lower}}} = 1.2 \times 10^5$ (*i.e.*, 12%)
- $R^{W^*} = 4 \times 10^5$ (*i.e.*, 40%)
- $R^{W^{\text{upper}}} = 8 \times 10^5$ (*i.e.*, 80%)
- $\frac{\partial u_t}{\partial t}^{\text{lower}} = 9.5 \times 10^5$ (*i.e.*, 95%)
- $\frac{\partial u_t}{\partial t}^{\text{upper}} = 1.05 \times 10^6$ (*i.e.*, 105%)

- $\Delta E_t^{u^{\text{lower}}} = -6 \times 10$ (*i.e.*, -60)
- $\Delta E_t^{u^{\text{upper}}} = 6 \times 10$ (*i.e.*, 60)
- $S^{\text{all}} = 10^6$ (*i.e.*, 1)
- $\Delta E_{t-1}^{u^{\text{first}^{\max}}} = 6 \times 10^2$ (*i.e.*, 600)
- $\mathfrak{Q} = 2.5 \times 10$ (*i.e.*, 25)
- $\sigma = 1 \times 10^{-1}$ (*i.e.*, 0.1)
- $F = 1$
- $D_\gamma^F = 1 \times 10^{-3}$ (*i.e.*, 0.1%)

12.1.5 Appendix

- $\mathscr{L}^{\lambda'^{\max}} = 10^{21}$ (*i.e.*, 1000)
- $\mathfrak{w}^{\mathrm{min}(\,)} = 10^7$ (*i.e.*, 10)
- $\Delta b^{\mathrm{MAX}^{\mathrm{m}}} = 2 \times 10^4$ (*i.e.*, 2%)
- $\Delta b^{\mathrm{MAX}^{\mathrm{b}}} = 2 \times 10^{11}$ (*i.e.*, 200,000)
- $\Delta B^{\mathrm{MAX}^{\mathrm{m}}} = 4 \times 10^4$ (*i.e.*, 4%)
- $\Delta B^{\mathrm{MAX}^{\mathrm{b}}} = 8 \times 10^{11}$ (*i.e.*, 800,000)

12.2 Upgradability

Pinto does not have governance. While Pinto is the first Beanstalk fork, additional development must be completed in order to create a generalized *Fork Migration System* that replaces the need for contract upgrades. In the meantime, limited upgrades to Pinto may be implemented by the *PCM*, the owner of the Pinto contract.

The *PCM* address has the exclusive and unilateral ability to *Pause* and *Unpause* the protocol, and commit *Pinto Improvements* (*PIs*). The *PCM* is a 5-of-9 Safe[19] multisig wallet with anonymous signers. In the future, we expect *PIs* will remove upgradability entirely, revoking these abilities from the *PCM*.

The following are potential improvements that can be incorporated into Pinto as one or more *PIs*[20]:

- Implement a *Fork Migration System* that allows users to migrate assets from one or more Pinto deployments to another under conditions defined by the Pinto deployment being migrated to;
- Add a mechanism to pay back old Beanstalk holders after the Pinto supply reaches 1 billion[21];
- Change parameters until 2 weeks after the first time the Pinto supply reaches 500M (*e.g.*, $\mathscr{T}$, R^D and R^W thresholds, Λ, $\mathscr{B}^{\lambda^*}$, etc.);
- Mint Pinto to fund a bug bounty program;
- The price oracle can be replaced with a trustless one; and
- Contracts can be made immutable.

12.2.1 Pause

The *PCM* can *Pause* and *Unpause* the protocol via *PIs*. When *Paused*, the protocol does not accept a `gm` function call. When *Unpaused*, the `gm` function can be called at the beginning of the next hour.

For a given timestamp of last *Unpause* (E_Ψ) during *Season* $t^{'}$, we define $E_t^{\min}$ $\forall$ $E_t^{\min}$ such that $t^{'} < t$ as:

$$E_t^{\min} = 3600\left(\left\lfloor \frac{E_\Psi}{3600} \right\rfloor + t - t^{'}\right)$$

12.2.2 Hypernative

Hypernative[22] is a proactive threat prevention and real-time monitoring platform that has the ability to remove facets upon detecting an impending or in-progress attack on the protocol. The *PCM* has the ability remove Hypernative protections when necessary for the security or censorship resistance of the protocol.

[19] app.safe.global/base:0x2cf82605402912C6a79078a9BBfcCf061CbfD507
[20] pinto.money/upgradability
[21] pinto.money/repay-beanstalk-debt
[22] hypernative.io

12.3 Δb Calculations

The following Δb calculations are used throughout this whitepaper.

12.3.1 $\Delta b^{\mathfrak{w}}_{\text{Ↄ}}$

The excess or shortage of Pinto in a *Well* at the end of the current transaction ($\Delta b^{\mathfrak{w}}_{\text{Ↄ}}$), such that $\Delta b^{\mathfrak{w}}_{\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, is calculated as the difference between the optimal number of Pinto ($\mathfrak{w}^{\circledast^*}_{\text{Ↄ}}$) and the number of Pinto ($\mathfrak{w}^{\circledast}_{\text{Ↄ}}$) in $\mathfrak{w}$ at the end of the current transaction, such that $\mathfrak{w}^{\circledast^*}_{\text{Ↄ}}$, $\mathfrak{w}^{\circledast}_{\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

The protocol calculates $\mathfrak{w}^{\circledast^*}_{\text{Ↄ}}$ by calling the *Well Function* `calcReserveAtRatioSwap` function with the number of Pinto ($\mathfrak{w}^{\circledast}_{\text{Ↄ}}$), such that $\mathfrak{w}^{\circledast}_{\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, and the number of non-Pinto in $\mathfrak{w}$ at the end of the current transaction ($\mathfrak{w}^{\circledcirc}_{\text{Ↄ}}$), such that $\mathfrak{w}^{\circledcirc}_{\text{Ↄ}} \in \{j \times 10^{-\circledcirc} \mid j \in \mathbb{Z}^+\}$, $f^{\$^{\mathfrak{w}}}$, and the data associated with the *Well Function* ($\mathfrak{w}^*$) as:

$$\mathfrak{w}^{\circledast^*}_{\text{Ↄ}} = \begin{cases} \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\circledast}_{\text{Ↄ}},\ \mathfrak{w}^{\circledcirc}_{\text{Ↄ}}],\ 0,\ [10^6,\ \frac{10^{24}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*) & \text{if } n > 8\colon \mathfrak{w}^{\circledcirc}_{\text{Ↄ}} \in \{j \times 10^{-n} \mid j \in \mathbb{Z}^+\} \\ \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\circledast}_{\text{Ↄ}},\ \mathfrak{w}^{\circledcirc}_{\text{Ↄ}}],\ 0,\ [10^{12},\ \frac{10^{30}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*) & \text{else} \end{cases}$$

The protocol calculates $\Delta b^{\mathfrak{w}}_{\text{Ↄ}}$ for a given $\mathfrak{w}^{\circledast}_{\text{Ↄ}}$, the minimum number of Pinto required in a *Well* in order to calculate PDV or Δb ($\mathfrak{w}^{\min(\circledast)}$), such that $\mathfrak{w}^{\min(\circledast)} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, and $\mathfrak{w}^{\circledast^*}_{\text{Ↄ}}$ as:

$$\Delta b^{\mathfrak{w}}_{\text{Ↄ}} = \begin{cases} 0 & \text{if } \mathfrak{w}^{\circledast}_{\text{Ↄ}} < \mathfrak{w}^{\min(\circledast)} \\ \mathfrak{w}^{\circledast^*}_{\text{Ↄ}} - \mathfrak{w}^{\circledast}_{\text{Ↄ}} & \text{else} \end{cases}$$

12.3.2 $\Delta B_{\text{Ↄ}}$

The protocol calculates $\Delta B_{\text{Ↄ}}$ for a given shortage or excess of Pinto in λ at the end of the current transaction ($\Delta b^{\lambda}_{\text{Ↄ}}$), such that $\Delta b^{\lambda}_{\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, $\forall \lambda \in$ *Minting Whitelist*, as:

$$\Delta B_{\text{Ↄ}} = \sum_{\lambda \in \Lambda} \Delta b^{\lambda}_{\text{Ↄ}}$$

12.3.3 $\Delta b^{\mathfrak{w}}_{<\text{Ↄ}}$

The excess or shortage of Pinto in a *Well* at the beginning of the current transaction ($\Delta b^{\mathfrak{w}}_{<\text{Ↄ}}$), such that $\Delta b^{\mathfrak{w}}_{<\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, is calculated as the difference between the optimal number of Pinto ($\mathfrak{w}^{\circledast^*}_{<\text{Ↄ}}$) and the number of Pinto ($\mathfrak{w}^{\circledast}_{\text{Ↄ}}$) in $\mathfrak{w}$ at the beginning of the current transaction, such that $\mathfrak{w}^{\circledast^*}_{<\text{Ↄ}}$, $\mathfrak{w}^{\circledast}_{<\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

The protocol calculates $\mathfrak{w}^{\circledast^*}_{<\text{Ↄ}}$ by calling the *Well Function* `calcReserveAtRatioSwap` function with $\mathfrak{w}^{\circledast}_{<\text{Ↄ}}$, $\mathfrak{w}^{\circledcirc}_{<\text{Ↄ}}$, $f^{\$^{\mathfrak{w}}}$ and $\mathfrak{w}^*$ as:

$$\mathfrak{w}^{\circledast^*}_{<\text{Ↄ}} = \begin{cases} \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\circledast}_{<\text{Ↄ}},\ \mathfrak{w}^{\circledcirc}_{<\text{Ↄ}}],\ 0,\ [10^6,\ \frac{10^{24}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*) & \text{if } n > 8\colon \mathfrak{w}^{\circledcirc}_{<\text{Ↄ}} \in \{j \times 10^{-n} \mid j \in \mathbb{Z}^+\} \\ \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\circledast}_{<\text{Ↄ}},\ \mathfrak{w}^{\circledcirc}_{<\text{Ↄ}}],\ 0,\ [10^{12},\ \frac{10^{30}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*) & \text{else} \end{cases}$$

The protocol calculates $\Delta b^{\mathfrak{w}}_{<\text{Ↄ}}$ for a given $\mathfrak{w}^{\circledcirc}_{<\text{Ↄ}}$, $\mathfrak{w}^{\min(\circledcirc)}$ and $\mathfrak{w}^{\circledcirc^*}_{<\text{Ↄ}}$ as:

$$\Delta b^{\mathfrak{w}}_{<\text{Ↄ}} = \begin{cases} 0 & \text{if } \mathfrak{w}^{\circledcirc}_{<\text{Ↄ}} < \mathfrak{w}^{\min(\circledcirc)} \\ \mathfrak{w}^{\circledcirc^*}_{<\text{Ↄ}} - \mathfrak{w}^{\circledcirc}_{<\text{Ↄ}} & \text{else} \end{cases}$$

12.3.4 $\Delta b^{\mathfrak{w}^{\text{LAST}}}_{\Xi-1}$

The excess or shortage of Pinto in a *Well* at the end of the last block ($\Delta b^{\mathfrak{w}^{\text{LAST}}}_{\Xi-1}$), such that $\Delta b^{\mathfrak{w}^{\text{LAST}}}_{\Xi-1} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, is calculated as the difference between the optimal number of Pinto ($\mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}}$) and the number of Pinto ($\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}}$) in $\mathfrak{w}$ at the end of the last block, such that $\mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}}$, $\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

The protocol calculates $\mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}}$ by calling the *Well Function* `calcReserveAtRatioSwap` function with $\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}}$, $\mathfrak{w}^{\text{LAST}}_{\circledast,\text{Ↄ}}$, $f^{\$^{\mathfrak{w}}}$ and $\mathfrak{w}^*$ as:

$$\mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}} = \begin{cases} \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}},\ \mathfrak{w}^{\text{LAST}}_{\circledast,\text{Ↄ}}],\ 0,\ [10^6,\ \frac{10^{24}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{if } n > 8\colon \mathfrak{w}^{\text{LAST}}_{\circledast,\text{Ↄ}} \in \{j \times 10^{-n} \mid j \in \mathbb{Z}^+\} \\ \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}},\ \mathfrak{w}^{\text{LAST}}_{\circledast,\text{Ↄ}}],\ 0,\ [10^{12},\ \frac{10^{30}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{else} \end{cases}$$

The protocol calculates $\Delta b^{\mathfrak{w}^{\text{LAST}}}_{\Xi-1}$ for a given $\mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}}$, $\mathfrak{w}^{\min(\circledcirc)}$ and $\mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}}$ as:

$$\Delta b^{\mathfrak{w}^{\text{LAST}}}_{\Xi-1} = \begin{cases} 0 & \text{if } \mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}} < \mathfrak{w}^{\min(\circledcirc)} \\ \mathfrak{w}^{\text{LAST}^*}_{\circledcirc,\text{Ↄ}} - \mathfrak{w}^{\text{LAST}}_{\circledcirc,\text{Ↄ}} & \text{else} \end{cases}$$

12.3.5 $\Delta b^{\mathfrak{w}^{\text{EMA}}}_{\Xi-1}$

The excess or shortage of Pinto in a *Well* based on the inter-block MEV manipulation resistant instantaneous reserves in the Multi Flow *Pump* ($\Delta b^{\mathfrak{w}^{\text{EMA}}}_{\Xi-1}$) of $\mathfrak{w}$, such that $\Delta b^{\mathfrak{w}^{\text{EMA}}}_{\Xi-1} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, is calculated as the difference between the optimal number of Pinto ($\mathfrak{w}^{\text{EMA}^*}_{\circledcirc,\text{Ↄ}}$) and the number of Pinto ($\mathfrak{w}^{\text{EMA}}_{\circledcirc,\text{Ↄ}}$) in $\mathfrak{w}$ based on the instantaneous reserves in the Multi Flow *Pump*, such that $\mathfrak{w}^{\text{EMA}^*}_{\circledcirc,\text{Ↄ}}$, $\mathfrak{w}^{\text{EMA}}_{\circledcirc,\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$.

The protocol calculates $\mathfrak{w}^{\text{EMA}^*}_{\circledcirc,\text{Ↄ}}$ by calling the *Well Function* `calcReserveAtRatioSwap` function with $\mathfrak{w}^{\text{EMA}}_{\circledcirc,\text{Ↄ}}$, $\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}$, $f^{\$^{\mathfrak{w}}}$ and $\mathfrak{w}^*$ as:

$$\mathfrak{w}^{\text{EMA}^*}_{\circledcirc,\text{Ↄ}} = \begin{cases} \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\text{EMA}}_{\circledcirc,\text{Ↄ}},\ \mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}],\ 0,\ [10^6,\ \frac{10^{24}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{if } n > 8\colon \mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}} \in \{j \times 10^{-n} \mid j \in \mathbb{Z}^+\} \\ \texttt{calcReserveAtRatioSwap}([\mathfrak{w}^{\text{EMA}}_{\circledcirc,\text{Ↄ}},\ \mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}],\ 0,\ [10^{12},\ \frac{10^{30}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{else} \end{cases}$$

The protocol calculates $\Delta b_{\Xi-1}^{\mathfrak{w}^{\text{EMA}}}$ for a given $\mathfrak{w}_{\text{🫘},\text{Ↄ}}^{\text{EMA}}$, $\mathfrak{w}^{\min(\text{🫘})}$ and $\mathfrak{w}_{\text{🫘},\text{Ↄ}}^{\text{EMA}^*}$ as:

$$\Delta b_{\Xi-1}^{\mathfrak{w}^{\text{EMA}}} = \begin{cases} 0 & \text{if } \mathfrak{w}_{\text{🫘},\text{Ↄ}}^{\text{EMA}} < \mathfrak{w}^{\min(\text{🫘})} \\ \mathfrak{w}_{\text{🫘},\text{Ↄ}}^{\text{EMA}^*} - \mathfrak{w}_{\text{🫘},\text{Ↄ}}^{\text{EMA}} & \text{else} \end{cases}$$

12.3.6 $\Delta B_{\Xi-1}^{\textbf{EMA}}$

The protocol calculates $\Delta B_{\Xi-1}^{\text{EMA}}$ for a given shortage or excess of Pinto in λ calculated using the inter-block MEV manipulation resistant instantaneous reserves in the Multi Flow *Pump* of λ ($\Delta b_{\Xi-1}^{\lambda^{\text{EMA}}}$), such that $\Delta b_{\Xi-1}^{\lambda^{\text{EMA}}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, $\forall \lambda \in$ *Minting Whitelist*, as:

$$\Delta B_{\Xi-1}^{\text{EMA}} = \sum_{\lambda \in \Lambda} \Delta b_{\Xi-1}^{\lambda^{\text{EMA}}}$$

12.3.7 $\Delta b_{\overline{t-1}}^{\mathfrak{w}}$

The time weighted average shortage or excess of Pinto in a *Well* over the previous *Season* ($\Delta b_{\overline{t-1}}^{\mathfrak{w}}$), such that $\Delta b_{\overline{t-1}}^{\mathfrak{w}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, is calculated as the difference between the optimal time weighted average number of Pinto in $\mathfrak{w}$ over the previous *Season* ($\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*}$), such that $\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, and $\mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}}$.

The protocol calculates $\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*}$ by calling the *Well Function* `calcReserveAtRatioSwap` function with $\mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}}$, $\mathfrak{w}_{\text{⬤},t-1_0,\text{Ↄ}}^{\text{SMA}}$, $f^{\$^{\mathfrak{w}}}$ and $\mathfrak{w}^*$ as:

$$\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*} = \begin{cases} \texttt{calcReserveAtRatioSwap(}[\mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}},\ \mathfrak{w}_{\text{⬤},t-1_0,\text{Ↄ}}^{\text{SMA}}], & \\ \qquad 0,\ [10^6,\ \frac{10^{24}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{if } n > 8\colon \mathfrak{w}_{\text{⬤},t-1_0,\text{Ↄ}}^{\text{SMA}} \in \{j \times 10^{-n} \mid j \in \mathbb{Z}^+\} \\ \texttt{calcReserveAtRatioSwap(}[\mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}},\ \mathfrak{w}_{\text{⬤},t-1_0,\text{Ↄ}}^{\text{SMA}}],\ 0,\ [10^{12},\ \frac{10^{30}}{f^{\$^{\mathfrak{w}}}(0)}],\ \mathfrak{w}^*\texttt{)} & \text{else} \end{cases}$$

The protocol calculates $\Delta b_{\overline{t-1}}^{\mathfrak{w}}$ for a given $\mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}}$, $\mathfrak{w}^{\min(\text{🫘})}$, $\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*}$, the Pinto supply at the end of the previous *Season* (B_{t-1}), such that $B_{t-1} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, the relative per *Well* minting cap ($\Delta b^{\text{MAX}^{\text{m}}}$), and the absolute per *Well* minting cap ($\Delta b^{\text{MAX}^{\text{b}}}$), such that $\Delta b^{\text{MAX}^{\text{m}}},\ \Delta b^{\text{MAX}^{\text{b}}} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as:

$$\Delta b_{\overline{t-1}}^{\mathfrak{w}} = \begin{cases} 0 & \text{if } \mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}} < \mathfrak{w}^{\min(\text{🫘})} \\ \max\left(\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*} - \mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}},\ \min(-\Delta b^{\text{MAX}^{\text{m}}} \times B_{t-1}, -\Delta b^{\text{MAX}^{\text{b}}})\right) & \text{if } \mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*} - \mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}} < 0 \\ \min\left(\mathfrak{w}_{\overline{t-1}}^{\text{🫘}^*} - \mathfrak{w}_{\text{🫘},t-1_0,\text{Ↄ}}^{\text{SMA}},\ \max(\Delta b^{\text{MAX}^{\text{m}}} \times B_{t-1}, \Delta b^{\text{MAX}^{\text{b}}})\right) & \text{else} \end{cases}$$

12.3.8 $\Delta B_{\overline{t-1}}$

The protocol calculates $\Delta B_{\overline{t-1}}$ for a given shortage or excess of Pinto in λ over the previous *Season* ($\Delta b^{\lambda}_{\overline{t-1}}$), such that $\Delta b^{\lambda}_{\overline{t-1}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, $\forall \lambda \in$ *Minting Whitelist*, the total relative minting cap ($\Delta B^{\mathrm{MAX}^{\mathrm{m}}}$), and the total absolute minting cap ($\Delta b^{\mathrm{MAX}^{\mathrm{b}}}$), such that $\Delta B^{\mathrm{MAX}^{\mathrm{m}}}$, $\Delta B^{\mathrm{MAX}^{\mathrm{b}}} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, as:

$$\Delta B_{\overline{t-1}} = \begin{cases} \max\left(\sum_{\lambda \in \Lambda} \Delta b^{\lambda}_{\overline{t-1}}, \min(-\Delta B^{\mathrm{MAX}^{\mathrm{m}}} \times B_{t-1}, -\Delta B^{\mathrm{MAX}^{\mathrm{b}}})\right) & \text{if } \sum_{\lambda \in \Lambda} \Delta b^{\lambda}_{\overline{t-1}} < 0 \\ \min\left(\sum_{\lambda \in \Lambda} \Delta b^{\lambda}_{\overline{t-1}}, \max(\Delta B^{\mathrm{MAX}^{\mathrm{m}}} \times B_{t-1}, \Delta B^{\mathrm{MAX}^{\mathrm{b}}})\right) & \text{else} \end{cases}$$

12.4 PDV Calculations

The following PDV function can be used by assets on the *Deposit Whitelist*.

12.4.1 $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$

The PDV of a number of *Well* LP tokens is calculated using the using the inter-block MEV manipulation resistant instantaneous Pinto reserves ($\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}$), such that $\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^{+}\}$, and non-Pinto reserves ($\mathfrak{w}^{\text{EMA}}_{\bigstar,\text{Ↄ}}$), such that $\mathfrak{w}^{\text{EMA}}_{\bigstar,\text{Ↄ}} \in \{j \times 10^{-\bigstar} \mid j \in \mathbb{Z}^{+}\}$, in the Multi Flow *Pump* of $\mathfrak{w}$ in the current transaction.

The protocol calculates the inter-block MEV manipulation resistant derivative of the $\mathfrak{w}$ LP token supply with respect to Pinto ($\frac{\partial \mathfrak{w}}{\partial \circledast}$), such that $\frac{\partial \mathfrak{w}}{\partial \circledast} \in \{j \times 10^{-18} \mid j \in \mathbb{Z}^{+}\}$, by calling the *Well Function* `calcLpTokenSupply` function with $\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}$, $\mathfrak{w}^{\text{EMA}}_{\bigstar,\text{Ↄ}}$ and $\mathfrak{w}^{*}$ as:

$$\frac{\partial \mathfrak{w}}{\partial \circledast} = \texttt{calcLpTokenSupply}([\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}} - 1,\ \mathfrak{w}^{\text{EMA}}_{\bigstar,\text{Ↄ}}],\ \mathfrak{w}^{*}) - \texttt{calcLpTokenSupply}([\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}},\ \mathfrak{w}^{\text{EMA}}_{\bigstar,\text{Ↄ}}],\ \mathfrak{w}^{*})$$

Therefore, we define $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$ for a given $\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}$, $\mathfrak{w}^{\min(\circledast)}$ and $\frac{\partial \mathfrak{w}}{\partial \circledast}$ as:

$$f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}}) = \begin{cases} \text{REVERT} & \text{if } \mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}} < \mathfrak{w}^{\min(\circledast)} \\ \frac{z^{\mathfrak{w}} \times 10^{6}}{\frac{\partial \mathfrak{w}}{\partial \circledast}} & \text{else} \end{cases}$$

12.5 LP Seed Gauge Point Functions

The following *LP Seed Gauge Point* function can be used by assets on the *Deposit Whitelist*.

12.5.1 $f^{\mathscr{L}^{\lambda'}}$

$f^{\mathscr{L}^{\lambda'}}$ increases the magnitude of the change of *LP Seed Gauge Points* of λ' the further away the percent of *Deposited* LP PDV of λ' ($\mathscr{B}^{\lambda'}$), such that $\mathscr{B}^{\lambda'} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, is from $\mathscr{B}^{\lambda'^*}$.

$f^{\mathscr{L}^{\lambda'}}$ requires three $\mathscr{L}^{\lambda'}$ levels to be set: (1) $\mathscr{L}^{\lambda'^{\text{close}}}$, (2) $\mathscr{L}^{\lambda'^{\text{far}}}$, and (3) $\mathscr{L}^{\lambda'^{\text{vfar}}}$, such that $\mathscr{L}^{\lambda'^{\text{close}}}$, $\mathscr{L}^{\lambda'^{\text{far}}}$, $\mathscr{L}^{\lambda'^{\text{vfar}}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, such that:

- If $\mathscr{B}^{\lambda'}$ is between $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{close}}})$ and $\mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{close}}})$ (*i.e.*, $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{close}}}) < \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{close}}})$), $\mathscr{B}^{\lambda'}$ is considered excessively close;
- Otherwise, if $\mathscr{B}^{\lambda'}$ is between $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{far}}})$ and $\mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{far}}})$ (*i.e.*, $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{far}}}) < \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{far}}})$), $\mathscr{B}^{\lambda'}$ is considered reasonably close;
- Otherwise, if $\mathscr{B}^{\lambda'}$ is between $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{vfar}}})$ and $\mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{vfar}}})$ (*i.e.*, $\mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{vfar}}}) < \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{vfar}}})$), $\mathscr{B}^{\lambda'}$ is considered reasonably far;
- Otherwise, (*i.e.*, $\mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{vfar}}}) \mid\mid \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{vfar}}}) < \mathscr{B}^{\lambda'}$), $\mathscr{B}^{\lambda'}$ is considered excessively far.

The *LP Seed Gauge Points* of λ' during t is calculated using the maximum *LP Seed Gauge Points* for a given λ' ($\mathscr{L}^{\lambda'^{\max}}$), such that $\mathscr{L}^{\lambda'^{\max}} \in \{j \times 10^{-18} \mid j \in \mathbb{Z}^+\}$, $\mathscr{L}^{\lambda'}_{t-1}$, $\mathscr{B}^{\lambda'}$, $\mathscr{B}^{\lambda'^*}$, $\mathscr{L}^{\lambda'^{\text{vfar}}}$, $\mathscr{L}^{\lambda'^{\text{far}}}$, and $\mathscr{L}^{\lambda'^{\text{close}}}$ as:

$$f^{\mathscr{L}^{\lambda'}} = \min\left(\mathscr{L}^{\lambda'^{\max}},\ \max\left(0, \begin{cases} \mathscr{L}^{\lambda'}_{t-1} + 5 & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{vfar}}}) \\ \mathscr{L}^{\lambda'}_{t-1} + 3 & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{far}}}) \\ \mathscr{L}^{\lambda'}_{t-1} + 1 & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} - (\mathscr{B}^{\lambda'^*} \times \mathscr{L}^{\lambda'^{\text{close}}}) \\ \mathscr{L}^{\lambda'}_{t-1} & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{close}}}) \\ \mathscr{L}^{\lambda'}_{t-1} - 1 & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{far}}}) \\ \mathscr{L}^{\lambda'}_{t-1} - 3 & \text{if } \mathscr{B}^{\lambda'} < \mathscr{B}^{\lambda'^*} + ((1 - \mathscr{B}^{\lambda'^*}) \times \mathscr{L}^{\lambda'^{\text{vfar}}}) \\ \mathscr{L}^{\lambda'}_{t-1} - 5 & \text{else} \end{cases}\right)\right)$$

12.6 Non-Pinto USD Price Functions

The following non-Pinto USD price function can be used by assets on the *Deposit Whitelist*.

12.6.1 $f^{\$^{\lambda'}}(\mathfrak{K})$

If $\mathfrak{K} = 0$, such that $\mathfrak{K} \in \mathbb{N}$, the protocol calculates the USD price of the non-Pinto asset in λ' by calling the Chainlink `latestRoundData`, otherwise the protocol calculates it as the time weighted average of Chainlink `getRoundData` over the last $\mathfrak{K}$ seconds.

12.7 Liquidity Weight Functions

The following *Liquidity Weight* function can be used by assets on the *Deposit Whitelist*.

12.7.1 $f^{\mathscr{W}^{\lambda'}}$

We define the *Liquidity Weight* of λ' as:

$$f^{\mathscr{W}^{\lambda'}} = 1 \times 10^8 \ (i.e.,\ 100\%)$$

12.8 Deposit Whitelist

The following ERC-20 Standard tokens are *Whitelisted* for *Deposit* in the *Silo*.

12.8.1 🫘

1. **Token Address:** The 🫘 token address (🫘$^{@}$), such that 🫘$^{@} \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, 🫘$^{@} =$ 0xb170000aeeFa790fa61D6e837d1035906839a3c8).
2. **PDV Function:** The PDV of 1 🫘 is 1.
   We define $f^{L^{🫘}}(z^{🫘})$ as:
   $$f^{L^{🫘}}(z^{🫘}) = z^{🫘}$$
3. **Stalk per PDV:** 🫘 *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^{🫘} = 10^{16}$).
4. **Initial Seeds per PDV:** 🫘 *Deposits* received 2 *Seeds* per PDV at deployment (*i.e.*, $c_{\circ}^{🫘} = 2 \times 10^{6}$).
5. **Initial LP Seed Gauge Points:** N/A
6. **LP Seed Gauge Point Function:** N/A
7. **Non-Pinto USD Price Function:** N/A
8. **Liquidity Weight Function:** N/A
9. **Optimal Percent of Deposited LP PDV:** N/A

12.8.2 ϑ (PINTO:WETH)

1. **Token Address:** The ϑ token address ($\vartheta^{@}$), such that $\vartheta^{@} \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\vartheta^{@} =$ 0x3e11001CfbB6dE5737327c59E10afAB47B82B5d3).
2. **PDV Function:** The PDV of ϑ is calculated using $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$.
3. **Stalk per PDV:** ϑ *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^{\vartheta} = 10^{16}$).
4. **Initial Seeds per PDV:** ϑ *Deposits* received 3 *Seeds* per PDV upon deployment (*i.e.*, $c_{\circ}^{\vartheta} = 3 \times 10^{6}$).
5. **Initial LP Seed Gauge Points:** The initial *LP Seed Gauge Points* of ϑ was 250 (*i.e.*, $\mathscr{L}_{\circ}^{\vartheta} = 2.5 \times 10^{20}$).
6. **LP Seed Gauge Point Function:** The number of *LP Seed Gauge Points* of ϑ during t is calculated using $f^{\mathscr{L}^{\lambda'}}$.
7. **Non-Pinto USD Price Function:** The USD price of the non-Pinto asset in ϑ is calculated using $f^{\$^{\lambda'}}(\mathfrak{K})$.
8. **Liquidity Weight Function:** The *Liquidity Weight* of ϑ is calculated using $f^{\mathscr{W}^{\lambda'}}$.
9. **Optimal Percent of Deposited LP PDV:** The optimal percent of *Deposited* LP PDV of ϑ is 25% (*i.e.*, $\mathscr{B}^{\vartheta^*} = 2.5 \times 10^{7}$).

10. **Additional Information:**
    - Non-Pinto USD Price Oracle: Chainlink ETH/USD data feed[23]
    - *Well Function*: Constant Product 2[24]
    - *Well Implementation*: Upgradable *Well*[25] (owned by the *PCM*)
    - *Pump*: Multi Flow[26]
        - The exchange rate cap per block (ρ), such that $\rho \in \{j \times 10^{-64} \mid j \in \mathbb{Z}^+\}$, is 10^{63} (*i.e.*, 0.1%) (see Multi Flow whitepaper).
        - The LP token supply change cap per block (λ^ρ), such that $\lambda^\rho \in \{j \times 10^{-64} \mid j \in \mathbb{Z}^+\}$, is 3×10^{64} (*i.e.*, 3%) (see Multi Flow whitepaper).

12.8.3 φ (PINTO:cbETH)

1. **Token Address:** The φ token address ($\varphi^@$), such that $\varphi^@ \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\varphi^@ =$ 0x3e111115A82dF6190e36ADf0d552880663A4dBF1).
2. **PDV Function:** The PDV of φ is calculated using $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$.
3. **Stalk per PDV:** φ *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^\varphi = 10^{16}$).
4. **Initial Seeds per PDV:** φ *Deposits* received 3 *Seeds* per PDV upon deployment (*i.e.*, $c^\varphi_\circ = 3 \times 10^6$).
5. **Initial LP Seed Gauge Points:** The initial *LP Seed Gauge Points* of φ was 250 (*i.e.*, $\mathscr{L}^\varphi_\circ = 2.5 \times 10^{20}$).
6. **LP Seed Gauge Point Function:** The number of *LP Seed Gauge Points* of φ during t is calculated using $f^{\mathscr{L}^{\lambda'}}$.
7. **Non-Pinto USD Price Function:** The USD price of the non-Pinto asset in φ is calculated using $f^{\$^{\lambda'}}(\mathfrak{K})$.
8. **Liquidity Weight Function:** The *Liquidity Weight* of φ is calculated using $f^{\mathscr{W}^{\lambda'}}$.
9. **Optimal Percent of Deposited LP PDV:** The optimal percent of *Deposited* LP PDV of φ is 25% (*i.e.*, $\mathscr{B}^{\varphi^*} = 2.5 \times 10^7$).
10. **Additional Information:**
    - Non-Pinto USD Price Oracle: Chainlink cbETH/USD data feed[27]
    - *Well Function*: Constant Product 2
    - *Well Implementation*: Upgradable *Well* (owned by the *PCM*)
    - *Pump*: Multi Flow
        - $\rho = 10^{63}$ (*i.e.*, 0.1%)
        - $\lambda^\rho = 3 \times 10^{64}$ (*i.e.*, 3%)

[23] basescan.org/address/0x71041dddad3595F9CEd3DcCFBe3D1F4b0a16Bb70
[24] basescan.org/address/0xBA510C289fD067EBbA41335afa11F0591940d6fe
[25] basescan.org/address/0xBA510990a720725Ab1F9a0D231F045fc906909f4
[26] basescan.org/address/0xBA51AA73ab1b8720E6D5602Bd3cBaaedB6399133
[27] basescan.org/address/0xd7818272B9e248357d13057AAb0B417aF31E817d

12.8.4 ϱ (PINTO:cbBTC)

1. **Token Address:** The ϱ token address ($\varrho^{@}$), such that $\varrho^{@} \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\varrho^{@} =$ 0x3e11226fe3d85142B734ABCe6e58918d5828d1b4).
2. **PDV Function:** The PDV of ϱ is calculated using $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$.
3. **Stalk per PDV:** ϱ *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^{\varrho} = 10^{16}$).
4. **Initial Seeds per PDV:** ϱ *Deposits* received 3 *Seeds* per PDV upon deployment (*i.e.*, $c^{\varrho}_{\circ} = 3 \times 10^{6}$).
5. **Initial LP Seed Gauge Points:** The initial *LP Seed Gauge Points* of ϱ was 500 (*i.e.*, $\mathscr{L}^{\varrho}_{\circ} = 5 \times 10^{20}$).
6. **LP Seed Gauge Point Function:** The number of *LP Seed Gauge Points* of ϱ during t is calculated using $f^{\mathscr{L}^{\lambda'}}$.
7. **Non-Pinto USD Price Function:** The USD price of the non-Pinto asset in ϱ is calculated using $f^{\$^{\lambda'}}$.
8. **Liquidity Weight Function:** The *Liquidity Weight* of ϱ is calculated using $f^{\mathscr{W}^{\lambda'}}$.
9. **Optimal Percent of Deposited LP PDV:** The optimal percent of *Deposited* LP PDV of ϱ is 25% (*i.e.*, $\mathscr{B}^{\varrho^*} = 2.5 \times 10^{7}$).
10. **Additional Information:**
    - Non-Pinto USD Price Oracle: Chainlink cbBTC/USD data feed[28]
    - *Well Function*: Constant Product 2
    - *Well Implementation*: Upgradable *Well* (owned by the *PCM*)
    - *Pump*: Multi Flow
      - $\rho = 10^{63}$ (*i.e.*, 0.1%)
      - $\lambda^{\rho} = 3 \times 10^{64}$ (*i.e.*, 3%)

12.8.5 ϕ (PINTO:USDC)

1. **Token Address:** The ϕ token address ($\phi^{@}$), such that $\phi^{@} \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\phi^{@} =$ 0x3e1133aC082716DDC3114bbEFEeD8B1731eA9cb1).
2. **PDV Function:** The PDV of ϕ is calculated using $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$.
3. **Stalk per PDV:** ϕ *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^{\phi} = 10^{16}$).
4. **Initial Seeds per PDV:** ϕ *Deposits* received 3 *Seeds* per PDV upon deployment (*i.e.*, $c^{\phi}_{\circ} = 3 \times 10^{6}$).
5. **Initial LP Seed Gauge Points:** The initial *LP Seed Gauge Points* of ϕ was 500 (*i.e.*, $\mathscr{L}^{\phi}_{\circ} = 5 \times 10^{20}$).
6. **LP Seed Gauge Point Function:** The number of *LP Seed Gauge Points* of ϕ during t is calculated using $f^{\mathscr{L}^{\lambda'}}$.

[28] basescan.org/address/0x07DA0E54543a844a80ABE69c8A12F22B3aA59f9D

7. **Non-Pinto USD Price Function:** The USD price of the non-Pinto asset in ϕ is calculated using $f^{\$^{\lambda'}}$.

8. **Liquidity Weight Function:** The *Liquidity Weight* of ϕ is calculated using $f^{\mathscr{W}^{\lambda'}}$.

9. **Optimal Percent of Deposited LP PDV:** The optimal percent of *Deposited* LP PDV of ϕ is 25% (*i.e.*, $\mathscr{B}^{\phi^*} = 2.5 \times 10^7$).

10. **Additional Information:**
    - Non-Pinto USD Price Oracle: Chainlink USDC/USD data feed[29]
    - *Well Function*: Stable 2[30]
    - *Well Implementation*: Upgradable *Well* (owned by the *PCM*)
    - *Pump*: Multi Flow
      - $\rho = 10^{63}$ (*i.e.*, 0.1%)
      - $\lambda^\rho = 3 \times 10^{64}$ (*i.e.*, 3%)

12.8.6 ϖ (PINTO:WSOL)

1. **Token Address:** The ϖ token address ($\varpi^@$), such that $\varpi^@ \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\varpi^@ =$ 0x3e11444c7650234c748D743D8d374fcE2eE5E6C9).

2. **PDV Function:** The PDV of ϖ is calculated using $f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$.

3. **Stalk per PDV:** ϖ *Deposits* receive 1 *Stalk* per PDV upon *Deposit* (*i.e.*, $k^\varpi = 10^{16}$).

4. **Initial Seeds per PDV:** ϖ *Deposits* received 3 *Seeds* per PDV upon deployment (*i.e.*, $c^\varpi_\circ = 3 \times 10^6$).

5. **Initial LP Seed Gauge Points:** The initial *LP Seed Gauge Points* of ϖ was 500 (*i.e.*, $\mathscr{L}^\varpi_\circ = 5 \times 10^{20}$).

6. **LP Seed Gauge Point Function:** The number of *LP Seed Gauge Points* of ϖ during t is calculated using $f^{\mathscr{L}^{\lambda'}}$.

7. **Non-Pinto USD Price Function:** The USD price of the non-Pinto asset in ϖ is calculated using $f^{\$^{\lambda'}}$.

8. **Liquidity Weight Function:** The *Liquidity Weight* of ϖ is calculated using $f^{\mathscr{W}^{\lambda'}}$.

9. **Optimal Percent of Deposited LP PDV:** The optimal percent of *Deposited* LP PDV of ϖ is 25% (*i.e.*, $\mathscr{B}^{\varpi^*} = 2.5 \times 10^7$).

10. **Additional Information:**
    - Non-Pinto USD Price Oracle: Chainlink SOL/USD data feed[31]
    - *Well Function*: Constant Product 2
    - *Well Implementation*: Upgradable *Well* (owned by the *PCM*)
    - *Pump*: Multi Flow
      - $\rho = 10^{63}$ (*i.e.*, 0.1%)
      - $\lambda^\rho = 3 \times 10^{64}$ (*i.e.*, 3%)

---

[29] basescan.org/address/0x7e860098F58bBFC8648a4311b374B1D669a2bc6B
[30] basescan.org/address/0xBA51055a97b40d7f41f3F64b57469b5D45B67c87
[31] basescan.org/address/0x975043adBb80fc32276CbF9Bbcfd4A601a12462D

12.9 Conversion Whitelist

The following *Conversions* within the *Silo* are *Whitelisted*.

12.9.1 $\lambda \to \lambda$

1. **From Token Address:** The from token address must match the to token address.
2. **To Token Address:** The to token address must match the from token address.
3. **Conditions:** *Deposited* λ can be *Converted* to a λ *Deposit* at anytime by the λ *Deposit* holder.
4. **Convert Function:** The number of λ received for *Converting Deposited* λ within the *Silo* is equivalent to the number of λ *Converted*. Therefore, we define the function as:

   $$f^{\lambda \to \lambda}(z^{\lambda}) = z^{\lambda}$$

   The stored PDV of a λ *Deposit* will only update if it increases above the previously stored value.
5. **Mown Stalk Reward / Burn Function:** N/A

12.9.2 $\lambda^{\circ} \to \lambda^{\circ}$

1. **From Token Address:** The from token address must match the to token address.
2. **To Token Address:** The to token address must match the from token address.
3. **Conditions:** *Deposited* λ° can be *Converted* to a λ° *Deposit* at anytime by any account on Base.
4. **Convert Function:** The number of λ received for *Converting Deposited* λ° within the *Silo* is equivalent to the number of λ° *Converted*. Therefore, we define the function as:

   $$f^{\lambda^{\circ} \to \lambda^{\circ}}(z^{\lambda^{\circ}}) = z^{\lambda^{\circ}}$$

   Upon a $\lambda^{\circ} \to \lambda^{\circ}$ *Conversion*, the stored PDV of a λ° *Deposit* is updated.

   Upon a $\lambda^{\circ} \to \lambda^{\circ}$ *Conversion* during t, the number of *Stalk* of the output λ° *Deposit* initially *Deposited* during i ($k^{\lambda^{\circ}}_{t,i,t}$), such that $k^{\lambda^{\circ}}_{t,i,t} \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^{+}\}$, is recalculated to account for the change in PDV of the *Deposit* as:

   $$k^{\lambda^{\circ}}_{t,i,t} = f^{L^{\lambda}}\left(z^{\lambda^{\circ}}\right)\left(k^{\lambda^{\circ}} + \sum_{i}^{t}\left(\frac{c_i^{\lambda^{\circ}}}{10000}\right)\right)$$

   The i associated with a *Converted Deposit* is automatically updated to preserve the correct amount of *Mown Stalk* and rounded down.
5. **Mown Stalk Reward / Burn Function:** N/A

12.9.3 Generalized Convert (*i.e.*, $\lambda \to \lambda'$)

1. **From Token Address:** The from token address can be any token address on the *Deposit Whitelist*.
2. **To Token Address:** The to token address can be any token address on the *Deposit Whitelist*.
3. **Conditions:** *Deposited* λ can be *Converted* to a λ' *Deposit* at anytime by the λ *Deposit* holder.
4. **Convert Function:** N/A
5. **Mown Stalk Reward / Burn Function:**

   Upon a $\lambda \to \lambda'$ *Conversion* during t, the number of *Stalk* of the output λ' *Deposit* initially *Deposited* during i ($k^{\lambda'}_{t,i,t}$) is calculated to account for (1) the change in PDV of the *Deposit* and (2) the *Mown Stalk* penalty ($g^{\lambda'^*}$), such that $k^{\lambda'}_{t,i,t},\ g^{\lambda'^*} \in \{j \times 10^{-16} \mid j \in \mathbb{Z}^+\}$.

   The *Mown Stalk* penalty is a function of (1) the PDV of the output λ' *Deposit*, (2) the maximum change in current Δb (*i.e.*, $|\Delta b_{\text{ↄ}} - \Delta b_{<\text{ↄ}}|$) of (a) λ, (b) λ' and (c) the sum of changes across all $\lambda \in \Lambda$, adjusted for the changes in current supplies of λ, λ' and all $\lambda \in \Lambda$, respectively ($\Delta\mathfrak{B}^{\max}$), and (3) the maximum cumulative change in Δb of (a) λ, (b) λ' and (c) the sum of cumulative changes across all $\lambda \in \Lambda$, adjusted for the changes in supplies of λ, λ' and all $\lambda \in \Lambda$, respectively, based on and relative to the inter-block MEV manipulation resistant last values in the Multi-Flow *Pump* for each λ at the end of the previous block, ($\Delta\mathfrak{B}^{\max^{\text{LAST}}}$), such that $\Delta\mathfrak{B}^{\max},\ \Delta\mathfrak{B}^{\max^{\text{LAST}}} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$.

   $\Delta\mathfrak{B}^{\max}$ is a function of (1) a function to determine the magnitude of the movement away from $\Delta b = 0$ given the Δb before and after a transaction ($f^{\Delta b^*}$), such that $f^{\Delta b^*} : \{(j,\ j') \times 10^{-6} \mid j,\ j' \in \mathbb{Z}^+\} \to \{j \times 10^{-6} \mid j \in \mathbb{Z}^+\}$, (2) $\Delta b^{\lambda}_{<\text{ↄ}}\ \forall\ \lambda \in \Lambda$, (3) $\Delta b^{\lambda}_{\text{ↄ}}\ \forall\ \lambda \in \Lambda$, and (4) the current supply of λ before and after the *Conversion* $\forall\ \lambda \in \Lambda$.

   We define $f^{\Delta b^*}$ for a given Δb before a transaction ($\Delta b_{<\text{ↄ}}$) and Δb after the transaction ($\Delta b_{\text{ↄ}}$), such that $\Delta b_{<\text{ↄ}},\ \Delta b_{\text{ↄ}} \in \{j \times 10^{-6} \mid j \in \mathbb{Z}\}$, as:

$$f^{\Delta b^*}(\Delta b_{<\text{ↄ}},\ \Delta b_{\text{ↄ}}) = \begin{cases} |\Delta b_{\text{ↄ}} - \Delta b_{<\text{ↄ}}| & \text{if } \Delta b_{<\text{ↄ}} \times \Delta b_{\text{ↄ}} > 0\ \&\&\ |\Delta b_{\text{ↄ}}| > |\Delta b_{<\text{ↄ}}| \\ 0 & \text{if } \Delta b_{<\text{ↄ}} \times \Delta b_{\text{ↄ}} > 0\ \&\&\ |\Delta b_{\text{ↄ}}| \leq |\Delta b_{<\text{ↄ}}| \\ |\Delta b_{\text{ↄ}}| & \text{else} \end{cases}$$

   Therefore, we define $\Delta\mathfrak{B}^{\max}$ for a given $f^{\Delta b^*}$, $\Delta b^{\lambda}_{<\text{ↄ}}\ \forall\ \lambda \in \Lambda$, $\Delta b^{\lambda}_{\text{ↄ}}\ \forall\ \lambda \in \Lambda$, and current supply of λ before and after the *Conversion* $\forall\ \lambda \in \Lambda$ as:

$$\begin{aligned} \Delta\mathfrak{B}^{\max} = \max\Bigg(& f^{\Delta b^*}\left(\Delta b^{\lambda}_{<\text{ↄ}},\ \frac{\Delta b^{\lambda}_{\text{ↄ}} \times \texttt{calcLpTokenSupply}([\lambda^{\text{❀}}_{<\text{ↄ}},\ \lambda^{\text{✿}}_{<\text{ↄ}}],\ \lambda^*)}{\texttt{calcLpTokenSupply}([\lambda^{\text{❀}}_{\text{ↄ}},\ \lambda^{\text{✿}}_{\text{ↄ}}],\ \lambda^*)}\right), \\ & f^{\Delta b^*}\left(\Delta b^{\lambda'}_{<\text{ↄ}},\ \frac{\Delta b^{\lambda'}_{\text{ↄ}} \times \texttt{calcLpTokenSupply}([\lambda'^{\text{❀}}_{<\text{ↄ}},\ \lambda'^{\text{✿}}_{<\text{ↄ}}],\ \lambda'^*)}{\texttt{calcLpTokenSupply}([\lambda'^{\text{❀}}_{\text{ↄ}},\ \lambda'^{\text{✿}}_{\text{ↄ}}],\ \lambda'^*)}\right), \\ & \sum_{\lambda \in \Lambda} f^{\Delta b^*}\left(\Delta b^{\lambda}_{<\text{ↄ}},\ \frac{\Delta b^{\lambda}_{\text{ↄ}} \times \texttt{calcLpTokenSupply}([\lambda^{\text{❀}}_{<\text{ↄ}},\ \lambda^{\text{✿}}_{<\text{ↄ}}],\ \lambda^*)}{\texttt{calcLpTokenSupply}([\lambda^{\text{❀}}_{\text{ↄ}},\ \lambda^{\text{✿}}_{\text{ↄ}}],\ \lambda^*)}\right)\Bigg) \end{aligned}$$

$\Delta\mathfrak{B}^{\max^{\text{LAST}}}$ is a function of (1) $f^{\Delta b^*}$, (2) the maximum remaining *Conversions* of λ permitted before a penalty is accrued ($\lambda^{\text{LAST}^*}_{\Xi-1}$), such that $\lambda^{\text{LAST}^*}_{\Xi-1} \in \{j \times 10^{-\lambda} \mid j \in \mathbb{N}\}$, $\forall\ \lambda \in \Lambda$, and (3) $\Delta b^{\lambda}_{\text{ↄ}}\ \forall\ \lambda \in \Lambda$.

In order to properly decrement $\lambda^{\text{LAST}^*}_{\Xi-1}$, at the beginning of each block Pinto stores the inter-block MEV manipulation resistant last values in the Multi-Flow *Pump* of λ ($\Delta b^{\lambda^{\text{LAST}}}_{\Xi-1}$), such that $\Delta b^{\lambda^{\text{LAST}}}_{\Xi-1} \in \{j \times 10^{-\lambda} \mid j \in \mathbb{Z}^+\}$, $\forall\ \lambda \in \Lambda$ (*i.e.*, $\lambda^{\text{LAST}^*}_{\Xi-1} = \Delta b^{\lambda^{\text{LAST}}}_{\Xi-1}$), and the sign of $\Delta b^{\lambda^{\text{LAST}}}_{\Xi-1}$ ($\lambda^{\text{LAST}^{\text{SIGN}}}_{\Xi-1}$), such that $\lambda^{\text{LAST}^{\text{SIGN}}}_{\Xi-1} \in \{1,\ 0\}$, $\forall\ \lambda \in \Lambda$, where 1 indicates a positive value and 0 indicates a negative value.

Upon each successful *Conversion*, the protocol updates $\lambda^{\text{LAST}^*}_{\Xi-1}$ as:

$$\lambda^{\text{LAST}^*}_{\Xi-1} = \begin{cases} \max\left(0,\ \left(|\lambda^{\text{LAST}^*}_{\Xi-1}| - \frac{|(\Delta b^{\lambda}_{\text{ↄ}} - \Delta b^{\lambda}_{<\text{ↄ}})| \times \texttt{calcLpTokenSupply}([\lambda^{\text{LAST}}_{\circledcirc,\text{ↄ}},\ \lambda^{\text{LAST}}_{\circledast,\text{ↄ}}],\ \lambda^*)}{\texttt{calcLpTokenSupply}([\lambda^{\circledcirc}_{\text{ↄ}},\ \lambda^{\circledast}_{\text{ↄ}}],\ \lambda^*)}\right)\right) & \text{if } \lambda^{\text{LAST}^{\text{SIGN}}}_{\Xi-1} == 1 \\ \min\left(0,\ \left(|\lambda^{\text{LAST}^*}_{\Xi-1}| - \frac{|(\Delta b^{\lambda}_{\text{ↄ}} - \Delta b^{\lambda}_{<\text{ↄ}})| \times \texttt{calcLpTokenSupply}([\lambda^{\text{LAST}}_{\circledcirc,\text{ↄ}},\ \lambda^{\text{LAST}}_{\circledast,\text{ↄ}}],\ \lambda^*)}{\texttt{calcLpTokenSupply}([\lambda^{\circledcirc}_{\text{ↄ}},\ \lambda^{\circledast}_{\text{ↄ}}],\ \lambda^*)}\right) \times -1\right) & \text{else} \end{cases}$$

Therefore, we define $\Delta\mathfrak{B}^{\max^{\text{LAST}}}$ for a given $f^{\Delta b^*}$, $\lambda^{\text{LAST}^*}_{\Xi-1}\ \forall\ \lambda \in \Lambda$, and $\Delta b^{\lambda}_{\text{ↄ}}\ \forall\ \lambda \in \Lambda$ as:

$$\Delta\mathfrak{B}^{\max^{\text{LAST}}} = \max\Bigg(f^{\Delta b^*}(\lambda^{\text{LAST}^*}_{\Xi-1},\ \Delta b^{\lambda}_{\text{ↄ}}), \\ f^{\Delta b^*}(\lambda^{\text{LAST}^*}_{\Xi-1},\ \Delta b^{\lambda'}_{\text{ↄ}}), \\ \sum_{\lambda \in \Lambda} f^{\Delta b^*}(\lambda^{\text{LAST}^*}_{\Xi-1},\ \Delta b^{\lambda}_{\text{ↄ}})\Bigg)$$

Therefore, we define the *Mown Stalk* penalty for a given $f^{L^{\lambda'}}(z^{\lambda'})$, $\Delta\mathfrak{B}^{\max}$ and $\Delta\mathfrak{B}^{\max^{\text{LAST}}}$ as:

$$g^{\lambda'^*} = 1 - \left(\frac{\max(f^{L^{\lambda'}}(z^{\lambda'}) - (\Delta\mathfrak{B}^{\max} + \Delta\mathfrak{B}^{\max^{\text{LAST}}}),\ 0)}{f^{L^{\lambda'}}(z^{\lambda'})}\right)$$

Therefore, we define $k^{\lambda'}_{t,i,t}$ for a given $f^{L^{\lambda'}}(z^{\lambda'})$, $k^{\lambda'}$, $f^{L^{\lambda}}(z^{\lambda})$, $g^{\lambda'^*}$, and $c^{\lambda'}_t$ in every *Season* from when it was *Deposited* until t as:

$$k^{\lambda'}_{t,i,t} = \begin{cases} f^{L^{\lambda'}}(z^{\lambda'}) \times k^{\lambda'} + f^{L^{\lambda}}(z^{\lambda}) \times g^{\lambda'^*} \times \sum_{i}^{t} \left(\frac{c^{\lambda}_i}{10000}\right) & \text{if } f^{L^{\lambda'}}(z^{\lambda'}) \geq f^{L^{\lambda}}(z^{\lambda}) \\ f^{L^{\lambda'}}(z^{\lambda'}) \left(k^{\lambda'} + g^{\lambda'^*} \times \sum_{i}^{t} \left(\frac{c^{\lambda}_i}{10000}\right)\right) & \text{else} \end{cases}$$

The i associated with a *Converted Deposit* is automatically updated to preserve the correct amount of *Mown Stalk* and rounded down.

12.9.4 $\circledcirc \rightarrow \mathfrak{w}$

1. **From Token Address:** $\circledcirc^{@}$
2. **To Token Address:** $\mathfrak{w}^{@}$
3. **Conditions:** *Deposited* $\circledcirc$ can be *Converted* to *Deposited* $\mathfrak{w}$ if (1) $\Delta b^{\mathfrak{w}^{\text{EMA}}}_{\Xi-1} > 0$ and (2) the change of Δb in $\mathfrak{w}$ does not exceed $\mathfrak{w}^{\text{LAST}^*}_{\Xi-1}$ (*i.e.*, $|\Delta b_{<\text{ↄ}} - \Delta b_{\text{ↄ}}| \leq \mathfrak{w}^{\text{LAST}^*}_{\Xi-1}$).

4. **Convert Function:** The number of $\mathfrak{w}$ received for *Converting Deposited* Pinto within the *Silo* for a given Pinto contract address ($\text{❁}^{@}$, such that $\text{❁}^{@} \in \{j \in \mathbb{N} \mid j < 16^{40}\}$, $\text{❁}^{@} =$ 0xD1A0D188E861ed9d15773a2F3574a2e94134bA8f) and minimum $\mathfrak{w}$ received ($\mathfrak{w}^{\min}$), such that $\mathfrak{w}^{\min} \in \{j \times 10^{-18} \mid j \in \mathbb{N}\}$, is the result of calling the *Well Implementation* `sync` function on $\mathfrak{w}$ immediately after sending $z^{\text{✿}}$ to $\mathfrak{w}$, as:

$$f^{\text{✿} \rightarrow \mathfrak{w}}(z^{\text{✿}}) = \mathfrak{w}\texttt{.sync(}\text{❁}^{@}, \mathfrak{w}^{\min}\texttt{)}$$

Upon each successful *Conversion*, the protocol updates $\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}$ as:

$$\mathfrak{w}_{\Xi-1}^{\text{LAST}^*} = \begin{cases} \max\left(0, \left(|\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}| - \frac{|(\Delta b_{\text{ↄ}}^{\mathfrak{w}} - \Delta b_{<\text{ↄ}}^{\mathfrak{w}})| \times \texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{✿},\text{ↄ}}^{\text{LAST}},\ \mathfrak{w}_{\text{❁},\text{ↄ}}^{\text{LAST}}\texttt{], }\mathfrak{w}^*\texttt{)}}{\texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{ↄ}}^{\text{✿}},\ \mathfrak{w}_{\text{ↄ}}^{\text{❁}}\texttt{], }\mathfrak{w}^*\texttt{)}}\right)\right) & \text{if } \mathfrak{w}_{\Xi-1}^{\text{LAST}^{\text{SIGN}}} == 1 \\ \min\left(0, \left(|\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}| - \frac{|(\Delta b_{\text{ↄ}}^{\mathfrak{w}} - \Delta b_{<\text{ↄ}}^{\mathfrak{w}})| \times \texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{✿},\text{ↄ}}^{\text{LAST}},\ \mathfrak{w}_{\text{❁},\text{ↄ}}^{\text{LAST}}\texttt{], }\mathfrak{w}^*\texttt{)}}{\texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{ↄ}}^{\text{✿}},\ \mathfrak{w}_{\text{ↄ}}^{\text{❁}}\texttt{], }\mathfrak{w}^*\texttt{)}}\right) \times -1\right) & \text{else} \end{cases}$$

The stored PDV of a *Converted* $\mathfrak{w}$ *Deposit* will only update if it increases above the previously stored value.

The i associated with a *Converted* $\mathfrak{w}$ *Deposit* is automatically updated to preserve the correct amount of *Mown Stalk* and rounded down.

5. **Mown Stalk Reward / Burn Function:** N/A

12.9.5 $\mathfrak{w} \rightarrow \text{✿}$

1. **From Token Address:** $\mathfrak{w}^{@}$
2. **To Token Address:** $\text{✿}^{@}$
3. **Conditions:** *Deposited* $\mathfrak{w}$ can be *Converted* to *Deposited* ✿ if (1) $\Delta b_{\Xi-1}^{\mathfrak{w}^{\text{EMA}}} < 0$ and (2) the change of Δb in $\mathfrak{w}$ does not exceed $\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}$ (*i.e.*, $|\Delta b_{<\text{ↄ}} - \Delta b_{\text{ↄ}}| \leq \mathfrak{w}_{\Xi-1}^{\text{LAST}^*}$).
4. **Convert Function:** The number of ✿ received for *Converting Deposited* $\mathfrak{w}$ within the *Silo* for a given $\text{✿}^{@}$, minimum number of ✿ received ($\text{✿}^{\min}$), such that $\text{✿}^{\min} \in \{j \times 10^{-6} \mid j \in \mathbb{N}\}$, $\text{❁}^{@}$ and `block.timestamp` is the result of calling the *Well Implementation* `removeLiquidityOneToken` function on $\mathfrak{w}$ as:

$$f^{\mathfrak{w} \rightarrow \text{✿}}(z^{\mathfrak{w}}) = \mathfrak{w}\texttt{.removeLiquidityOneToken(}z^{\mathfrak{w}},\ \text{✿}^{@},\ \text{✿}^{\min},\ \text{❁}^{@},\ \texttt{block.timestamp)}$$

Upon each successful *Conversion*, the protocol updates $\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}$ as:

$$\mathfrak{w}_{\Xi-1}^{\text{LAST}^*} = \begin{cases} \max\left(0, \left(|\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}| - \frac{|(\Delta b_{\text{ↄ}}^{\mathfrak{w}} - \Delta b_{<\text{ↄ}}^{\mathfrak{w}})| \times \texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{✿},\text{ↄ}}^{\text{LAST}},\ \mathfrak{w}_{\text{❁},\text{ↄ}}^{\text{LAST}}\texttt{], }\mathfrak{w}^*\texttt{)}}{\texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{ↄ}}^{\text{✿}},\ \mathfrak{w}_{\text{ↄ}}^{\text{❁}}\texttt{], }\mathfrak{w}^*\texttt{)}}\right)\right) & \text{if } \mathfrak{w}_{\Xi-1}^{\text{LAST}^{\text{SIGN}}} == 1 \\ \min\left(0, \left(|\mathfrak{w}_{\Xi-1}^{\text{LAST}^*}| - \frac{|(\Delta b_{\text{ↄ}}^{\mathfrak{w}} - \Delta b_{<\text{ↄ}}^{\mathfrak{w}})| \times \texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{✿},\text{ↄ}}^{\text{LAST}},\ \mathfrak{w}_{\text{❁},\text{ↄ}}^{\text{LAST}}\texttt{], }\mathfrak{w}^*\texttt{)}}{\texttt{calcLpTokenSupply([}\mathfrak{w}_{\text{ↄ}}^{\text{✿}},\ \mathfrak{w}_{\text{ↄ}}^{\text{❁}}\texttt{], }\mathfrak{w}^*\texttt{)}}\right) \times -1\right) & \text{else} \end{cases}$$

The stored PDV of a *Converted* ✿ *Deposit* will only update if it increases above the previously stored value.

The i associated with a *Converted* ✿ *Deposit* is automatically updated to preserve the correct amount of *Mown Stalk* and rounded down.

5. **Mown Stalk Reward / Burn Function:** N/A

12.10 Minting Whitelist

The following liquidity pools are *Whitelisted* for inclusion in the calculation of $\Delta B_{\overline{t-1}}$.

12.10.1 ϑ

1. **Pool Address:** $\vartheta^{@}$
2. $\Delta b_{\overline{t-1}}$ **Calculation:** The protocol calculates $\Delta b^{\vartheta}_{\overline{t-1}}$ using $\Delta b^{\mathfrak{w}}_{\overline{t-1}}$.

12.10.2 φ

1. **Pool Address:** $\varphi^{@}$
2. $\Delta b_{\overline{t-1}}$ **Calculation:** The protocol calculates $\Delta b^{\varphi}_{\overline{t-1}}$ using $\Delta b^{\mathfrak{w}}_{\overline{t-1}}$.

12.10.3 ϱ

1. **Pool Address:** $\varrho^{@}$
2. $\Delta b_{\overline{t-1}}$ **Calculation:** The protocol calculates $\Delta b^{\varrho}_{\overline{t-1}}$ using $\Delta b^{\mathfrak{w}}_{\overline{t-1}}$.

12.10.4 ϕ

1. **Pool Address:** $\phi^{@}$
2. $\Delta b_{\overline{t-1}}$ **Calculation:** The protocol calculates $\Delta b^{\phi}_{\overline{t-1}}$ using $\Delta b^{\mathfrak{w}}_{\overline{t-1}}$.

12.10.5 ϖ

1. **Pool Address:** $\varpi^{@}$
2. $\Delta b_{\overline{t-1}}$ **Calculation:** The protocol calculates $\Delta b^{\varpi}_{\overline{t-1}}$ using $\Delta b^{\mathfrak{w}}_{\overline{t-1}}$.

12.11 Flood Whitelist

At the beginning of each *Season* in which it *Floods*, the protocol mints additional Pinto and sells them directly in the following pools with the highest shortage of Pinto at the end of the previous *Season*.

12.11.1 ϑ

1. **Pool Address:** $\vartheta^{@}$
2. $\Delta b_{\Game}$ **Calculation:** The protocol calculates $\Delta b_{\Game}^{\vartheta}$ using $\Delta b_{\Game}^{\mathfrak{w}}$.

12.11.2 φ

1. **Pool Address:** $\varphi^{@}$
2. $\Delta b_{\Game}$ **Calculation:** The protocol calculates $\Delta b_{\Game}^{\varphi}$ using $\Delta b_{\Game}^{\mathfrak{w}}$.

12.11.3 ϱ

1. **Pool Address:** $\varrho^{@}$
2. $\Delta b_{\Game}$ **Calculation:** The protocol calculates $\Delta b_{\Game}^{\varrho}$ using $\Delta b_{\Game}^{\mathfrak{w}}$.

12.11.4 ϕ

1. **Pool Address:** $\phi^{@}$
2. $\Delta b_{\Game}$ **Calculation:** The protocol calculates $\Delta b_{\Game}^{\phi}$ using $\Delta b_{\Game}^{\mathfrak{w}}$.

12.11.5 ϖ

1. **Pool Address:** $\varpi^{@}$
2. $\Delta b_{\Game}$ **Calculation:** The protocol calculates $\Delta b_{\Game}^{\varpi}$ using $\Delta b_{\Game}^{\mathfrak{w}}$.

12.12 Pod Market

The *Pod Market* is a peer-to-peer marketplace that allows *Pods* to be bought and sold in a trustless fashion without trading fees.

12.12.1 Pod Orders

Anyone with Pinto not in the *Silo* can *Order Pods*.

A *Pod Order* has four inputs:

1. The maximum number of *Pods* to be purchased;
2. The maximum place in the *Pod Line* (*i.e.*, the number of *Pods* that will become *Harvestable* before a given *Pod*) to purchase from;
3. The minimum number of Pods that can *Fill* the *Pod Order*; and
4. A constant that represents the maximum price per *Pod*, denominated in Pinto.

A *Pod Order* can be *Cancelled* at any time until it is entirely *Filled*. To facilitate instant clearance, Pinto are locked in a *Pod Order* until it is entirely *Filled* or *Cancelled*. Pinto can only be locked in a single *Pod Order* at a time.

12.12.2 Pod Listings

Pods that *Yield* from Pinto that were *Sown* from a single call of the `sow` function form a *Plot*. Anyone with a *Plot* can *List* a whole or partial *Plot* to be sold for Pinto.

A *Pod Listing* has six inputs:

1. The *Plot* being *Listed*;
2. The difference between the front of the portion of the *Plot* included in the *Pod Listing* from the front of the whole *Plot*, denominated in *Pods*, where a null input *Lists* from the back of the *Plot*;
3. The number of *Pods* in the *Plot* for sale, where a null input *Lists* the whole *Plot*;
4. The minimum number of Pods that can *Fill* the *Pod Listing*;
5. The maximum number of total *Pods* that can become *Harvestable* over all *Seasons* before the *Pod Listing* expires; and
6. A constant that represents the minimum price per *Pod*, denominated in Pinto.

A *Pod Listing* can be *Cancelled* at any time until it is entirely *Filled*. *Plots* can only be *Listed* in a single *Pod Listing* at a time. *Pod Listings* are automatically *Cancelled* if the owner of the *Plot* transfers, or simultaneously includes in another *Listing*, any *Pods* in the *Plot*.

12.12.3 Clearance

An outstanding *Pod Order* can be entirely or partially *Filled* at any time by a *Pod* seller. If the *Pod Order* is partially *Filled*, the rest of the *Pod Order* remains. Similarly, an outstanding *Pod Listing* can be entirely or partially *Filled* at any time by a *Pod* buyer. If the *Pod Listing* is partially *Filled*, the rest of the *Pod Listing* remains.

In instances where $0 < \Delta D_t$ causes a *Pod Order* and *Pod Listing* that previously were not overlapping to overlap, either the buyer or seller can *Fill* their *Order* or *Listing*, respectively, at their preferred price.

12.13 Depot

The following *Pipeline* to the *Depot* currently exists.

12.13.1 Pipeline

The Pipeline *Pipeline* allows anyone to perform an arbitrary series of actions on Base in a single transaction by using 0xb1bE0001f5a373b69b1E132b420e6D9687155e80[32] as a sandbox for execution.

The following functions to interact with Pipeline can be called through the Pipeline *Pipeline*.

- `pipe(...)`
- `multiPipe(...)`
- `advancedPipe(...)`

[32] basescan.org/address/0xb1bE0001f5a373b69b1E132b420e6D9687155e80

12.14 Glossary

The following conventions are used throughout this whitepaper:

- Lower case Latin letters are unique values;
- Upper case Latin letters are totals or rates;
- Subscripts are time, where t is the current *Season*, q is the current interval of t, Ξ is the end of the current block, and $\Game$ is the current transaction; and
- Superscripts are modifiers.

Δ, ∂ and f are ignored for the purposes of categorization and ordering in this glossary.

The following variables and terms are used throughout this whitepaper:

12.14.1 Terms

AMM - Automated market maker;

Burnt - Sent to the null address;

Cancel - Revoke an offer to buy or sell *Pods*;

Conversion - Exchange one *Deposited* λ for another within the *Silo*;

Conversion Whitelist - The whitelist that permissions *Conversions* within the *Silo*;

Crop Ratio - Determines the ratio between (1) the *Seeds* per *Deposited* in the *Silo* that are not *Germinating* at the beginning of t and (2) the *Seeds* per PDV of the λ with the highest *LP Seed Gauge Points* per PDV of λ *Deposited* in the *Silo* that are not *Germinating* at the beginning of t;

Crop Scalar - The value adjusted by the peg maintenance mechanism in order to change the *Crop Ratio*;

DeFi - Decentralized finance;

Deposit - An asset in the *Silo*;

Deposit Whitelist - The whitelist that permissions *Deposits* into the *Silo*;

Depositors - Holders of *Deposited* assets in the *Silo*;

Depot - Facilitates complex, gas-efficient interactions with other Base-native protocols in a single transaction;

Earned - Pinto paid to a *Stalkholder* after the last *Season* the *Stalkholder* called the `plant` function;

ETH - Ether;

Farm - Where Pinto peg maintenance and use of Pinto take place;

Farm Balances - Assets stored on the *Farm* that are not *Deposited* in the *Silo* or locked in *Pod Orders*;

Field - The protocol-native credit facility;

FIFO - First in, first out;

Fill - Match an outstanding offer to buy or sell *Pods*;

Flood - When the protocol mints extra Pinto and sells them directly in liquidity pools on the *Flood Whitelist*;

Flood Whitelist - The whitelist of pools that the protocol sells Pinto directly in when it mints extra Pinto during a *Flood*;

Fork Migration System - A protocol-native system that allows users to migrate assets from one or more Pinto deployments to another;

Germination - The period during which *Deposits* (1) cannot be *Converted*, (2) are not eligible for mints and (3) are not included in *Seed Gauge System* PDV calculations;

Grow - *Stalk* being created by *Seeds*;

Grown Stalk - *Stalk* that has been created by *Seeds* and not yet *Mown*;

Harvest - Redeem;

Harvestable - Redeemable;

Harvested - Redeemed;

LP Seed Gauge Points - Points that determine *Grown Stalk* issuance to each non-Pinto λ per the *Seed Gauge System*;

Liquidity Rate - The protocol liquidity level relative to the Pinto supply;

List - Create an offer to sell *Pods*;

LP tokens - Liquidity pool tokens;

Maximum Soil - The maximum *Soil* supply at a given interval during a *Season*;

Maximum Temperature - The maximum *Temperature* the protocol is willing to offer during a *Season*;

Minting Whitelist - The whitelist of liquidity pools included in the calculation of $\Delta B_{\overline{t-1}}$;

MoE - Medium of exchange;

Morning - The first $\mathfrak{Q}$ intervals of each *Season*;

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

Mown Stalk - *Stalk* that has been created by *Seeds* and has been *Mown*;

Minimum Soil - The minimum *Soil* supply at a given interval during a *Season*;

Order - Create an offer to buy *Pods*;

Pause - Stop accepting `gm` function calls;

Paused - The protocol has stopped accepting `gm` function calls;

Pinto Contract Multisig - The owner of the Pinto contract;

Pipeline - A connection between the protocol and another Base-native protocol via the *Depot*;

PCM - The *Pinto Contract Multisig*;

PDV - Pinto-Denominated-Value;

Plant - Turn *Seeds* associated with *Earned* into *Seeds* by *Depositing* the *Earned* in the current *Season*;

Plantable Seeds - *Seeds* that can be *Planted*;

Plot - Pinto *Sown* from a single call of the `sow` function;

Pod Line - The order of *Pods* that will become *Harvestable*;

Pod Listing - An offer to sell *Pods*;

Pod Market - A protocol-native marketplace for buying and selling *Pods*;

Pod Order - An offer to buy *Pods*;

Pod Rate - The protocol debt level relative to the Pinto supply;

Pods - The protocol-native debt asset, redeemable for 1 each once they *Ripen*;

Raining - $V < P_{\overline{t-1}}$ and $R^D_{t-1} < R^{D^{\text{lower}}}$;

Ripen - Become *Harvestable*;

Season - Protocol-native discrete time;

Seed - An illiquid token that *Grows* $\frac{1}{10000}$ *Stalk* each *Season*;

Seed Gauge Points - Points that determine *Grown Stalk* issuance to each λ per the *Seed Gauge System*;

Seed Gauge System - The protocol-native mechanism for adjusting *Seeds*;

Silo - The protocol-native *Deposit* facility;

Soil - The number of Pinto the protocol is willing to borrow;

SoV - Store of value;

Sow - Lend Pinto to the protocol;

Sower - A Pinto creditor;

Sown - Lent;

Stalk - An illiquid token that entitles the holder to a pro rata share of future Pinto mints.

Stalkholder - Holders of the *Stalk* token;

Stalk System - The protocol-native mechanism for *Stalk*;

Sun - The protocol-native timekeeping and code execution mechanism;

Temperature - The interest rate on Pinto loans;

Toolshed - A suite of tools that decrease friction to use Pinto and participate in peg maintenance;

Tractor - A protocol-native marketplace that allows third parties to perform pre-authorized actions through the protocol on behalf of a user;

Transfer - Send a *Deposit*;

TWA - Time weighted average;

Unharvestable Pods - *Pods* that are not yet redeemable;

Unpause - Resume accepting `gm` function calls;

Unpaused - The protocol has resumed accepting `gm` function calls;

UoA - Unit of account;

USD - US Dollar;

Wallet Balances - Assets stored in users' wallets;

Withdraw - Remove from the *Silo*; and

Yield - *Pods* being created from *Sown* .

12.14.2 Latin Alphabet Variables

a_t - The reward for successfully calling the `gm` function for t;

B - The total Pinto supply;

B_{t-1} - The Pinto supply at the end of the previous *Season*;

$\Delta B_{\overline{t-1}}$ - The sum of time weighted average shortages or excess of Pinto across liquidity pools on the *Minting Whitelist* over the previous *Season*;

$\Delta B^{\mathrm{EMA}}_{\Xi-1}$ - The sum of shortages and excesses of Pinto across liquidity pools on the *Minting Whitelist* calculated using the inter-block MEV manipulation resistant instantaneous reserves in Multi Flow *Pump*;

$\Delta B_{\text{Ↄ}}$ - The sum of differences between the optimal number of Pinto and the number of Pinto in each liquidity pool on the *Flood Whitelist* in the current transaction;

$\Delta B^{\mathrm{MAX^b}}$ - The total absolute minting cap;

$\Delta B^{\mathrm{MAX^m}}$ - The total relative minting cap;

$\Delta b^{\mathfrak{w}}_{\overline{t-1}}$ - The time weighted average shortage or excess of Pinto in a *Well* over the previous *Season*;

$\Delta b^{\lambda}_{\overline{t-1}}$ - The time weighted average shortage or excess of Pinto in λ over the previous *Season*;

$\Delta b^{\mathfrak{w}}_{\text{Ↄ}}$ - The excess or shortage of Pinto in a *Well* at the end of the current transaction;

$\Delta b^{\lambda}_{\text{Ↄ}}$ - The excess or shortage of Pinto in λ at the end of the current transaction;

$\Delta b^{\mathfrak{w}}_{<\text{Ↄ}}$ - The excess or shortage of Pinto in a *Well* at the beginning of the current transaction;

$\Delta b^{\mathfrak{w}^{\mathrm{LAST}}}_{\Xi-1}$ - The excess or shortage of Pinto in a *Well* at the end of the last block;

$\Delta b^{\mathfrak{w}^{\mathrm{EMA}}}_{\Xi-1}$ - The excess or shortage of Pinto in a *Well* based on the inter-block MEV manipulation resistant instantaneous reserves in the Multi Flow *Pump* of the *Well*;

$\Delta b^{\lambda^{\mathrm{EMA}}}_{\Xi-1}$ - The shortage or excess of Pinto in λ calculated using the inter-block MEV manipulation resistant instantaneous reserves in Multi Flow *Pump*;

$\Delta b^{\mathrm{MAX^b}}$ - The absolute per *Well* minting cap

$\Delta b^{\mathrm{MAX^m}}$ - The relative per *Well* minting cap;

$f^{\Delta b^*}$ - The function to determine the magnitude of the movement away from $\Delta b = 0$ given the Δb before and after a transaction;

$\mathbf{C}_t$ - The total *Seeds* during t;

$\overline{\mathbf{C}}^{\circ}$ - The initial average *Seeds* per PDV issued in the *Silo*;

$\overline{\mathbf{C}}^{\min}$ - The minimum average *Seeds* per PDV issued in the *Silo*;

C_t - A *Stalkholder's* total *Seeds* during t;

$C_{t,i}^{\lambda}$ - The number of *Seeds* during t for a given λ *Deposit Deposited* during i;

$c_{\circ}^{\lambda}$ - The initial number of *Seeds* per PDV of λ *Deposited*;

c_t^{λ} - The *Seeds* during t for a given *Deposit*;

ΔD_t - The number of *Pods* that *Ripen* and become *Harvestable* at the beginning of each *Season*;

D - The total number of *Unharvestable Pods*;

D_{γ} - The number of *Unharvestable Pods* that grew prior to the *Flood*;

D_{γ}^{F} - The maximum percent of the Pinto supply worth of *Pods* that grew from Pinto *Sown* before it began to *Rain* that become *Harvestable* during a *Flood*;

d - The number of *Pods* that *Yield* from a given number of *Sown* ;

E - Base block timestamps;

E_1 - The timestamp in the Base block containing the Pinto deployment;

$E_t^{\min}$ - The minimum timestamp Pinto accepts a `gm` function call for a given t;

E_{Ξ} - The timestamp of the current block;

E_{Ψ} - The timestamp in which the protocol last *Unpaused*;

$E_t^{u^{\text{first}}}$ - The Base timestamp of the first *Sow* in t such that there is at most one *Soil*;

ΔE_t^{u} - The difference in time it took for the Pinto to be *Sown* in all *Soil* over the previous two *Seasons*;

$\Delta E_t^{u^{\text{first}}}$ - The time of the first *Sow* such that Pinto are *Sown* in all *Soil* in each *Season*;

$\Delta E_t^{u^{\text{lower}}}$ - The level below which demand for *Soil* is considered decreasing when the difference in time it took for the Pinto to be *Sown* in all *Soil* over the previous two *Seasons* is used to calculate demand for *Soil*;

$\Delta E_t^{u^{\text{upper}}}$ - The level above or equal to which demand for *Soil* is considered increasing when the difference in time it took for the Pinto to be *Sown* in all *Soil* over the previous two *Seasons* is used to calculate demand for *Soil*;

$\Delta E_{t-1}^{u^{\text{first}^{\max}}}$ - The level below which if Pinto were *Sown* in all *Soil* of the previous *Season*, demand for *Soil* is considered increasing;

F - The number of consecutive full *Seasons* it must *Rain* in order to *Flood* at the beginning of the next *Season*;

G_t - A *Stalkholder's* total *Grown Stalk* that can be *Mown* during t;

$g_{t,i,\varkappa^{\lambda}}^{\lambda}$ - The *Grown Stalk* from *Seeds* from λ *Deposits* that can be *Mown* during t to start earning Pinto seigniorage for a given *Deposit* of a *Stalkholder* that last *Mowed* their *Grown Stalk* from λ *Deposits* in $\varkappa^{\lambda}$;

$g^{\lambda'^{*}}$ - The *Mown Stalk* penalty in Generalized *Convert*;

H_t - The *Maximum Temperature* during t;

H^b - Absolute change to the *Maximum Temperature*;

H^m - Relative change to the *Maximum Temperature*;

h - The *Temperature*;

h_{t_q} - The *Temperature* in interval q of t;

$\mathbf{K}_t$ - The total *Stalk* in the *Silo* during t;

$\widehat{\mathbf{K}_t}$ - The total *Mown Stalk* in the *Silo* during t;

K_t - A *Stalkholder's* total *Stalk* during t;

k^{λ} - The number of *Stalk* per PDV of λ *Deposited*;

$k^{\lambda}_{t,i,\varkappa^{\lambda}}$ - The *Stalk* during t for a given *Deposit* of a *Stalkholder* that last *Mowed* their *Grown Stalk* from λ *Deposits* in $\varkappa^{\lambda}$;

$\mathbf{L}_t$ - The total *Deposited* in the *Silo* that are not *Germinating* at the beginning of t;

$\mathbf{L}^{\lambda}_t$ - The total PDV of λ *Deposited* in the *Silo* that are not *Germinating* at the beginning of t;

$\mathbf{L}^{\lambda^{\max}}_t$ - The total PDV of the λ with the highest *LP Seed Gauge Points* per PDV of λ *Deposited* in the *Silo* that are not *Germinating* at the beginning of t;

L^{λ}_i - The total PDV of Z^{λ}_i when *Deposited*;

$f^{L^{\lambda}}(z^{\lambda})$ - The function to calculate the flash-loan-resistant PDV of a given number of *Deposited* λ;

$f^{L^{\mathfrak{w}}}(z^{\mathfrak{w}})$ - The function to calculate the flash-loan-resistant PDV of a given number of *Deposited* $\mathfrak{w}$;

M - The total Pinto minted over all *Seasons*;

m_t - The number of Pinto that the protocol mints at the beginning of each *Season*;

N - The total *Burnt* over all *Seasons*;

P - The price;

$P_{\overline{t-1}}$ - The inferred liquidity and time weighted average price of 1 compared to V over the previous *Season*;

P^{*} - The optimal price;

P^{upper} - A level above which price is considered excessively high;

Q - The length of each interval of the *Morning* in seconds;

q - The current interval of t;

R^D - The *Pod Rate*;

R^D_{t-1} - The *Pod Rate* at the end of the previous *Season*;

R^{D^*} - An optimal level of debt;

$R^{D^{\text{lower}}}$ - A level below which debt is considered excessively low;

$R^{D^{\text{upper}}}$ - A level above or equal to which debt is considered excessively high;

R^W - The *Liquidity Rate*;

R^W_{t-1} - The *Liquidity Rate* at the end of the previous *Season*;

R^{W^*} - An optimal level of liquidity;

$R^{W^{\text{lower}}}$ - A level below which liquidity is considered excessively low;

$R^{W^{\text{upper}}}$ - A level above or equal to which liquidity is considered excessively high;

S - *Soil*;

S_t^{end} - The *Soil* supply at the end of the *Season*;

S^{all} - The amount of remaining *Soil* below or equal to which Pinto are considered to be *Sown* in all *Soil* in a *Season*;

$S_{t_q}^{\max}$ - The *Maximum Soil* in interval q of t;

$S_{t_q}^{\min}$ - The *Minimum Soil* in interval q of t;

t - The current *Season*;

U - The total *Sown* over all *Seasons*;

u - The number of *Sown*;

u_t - The number of *Sown* during t;

$\frac{\partial u_t}{\partial t}$ - The rate of change of u_t over the previous two *Seasons*;

$\frac{\partial u_t}{\partial t}^{\text{lower}}$ - The level below which demand for *Soil* is considered decreasing;

$\frac{\partial u_t}{\partial t}^{\text{upper}}$ - The level above or equal to which demand for *Soil* is considered increasing;

V - The value target for 1;

W - The total non-Pinto USD-denominated value of liquidity in the *Silo*;

Z_i^{λ} - The total number of λ *Deposited* during *Season* i;

z^{λ} - A given number of *Deposited* λ;

12.14.3 Mathfrak Style Latin Alphabet Variables

$\Delta\mathfrak{B}^{\max}$ - The maximum change in current Δb of (a) λ, (b) λ' and (c) the sum of changes across all $\lambda \in \Lambda$, adjusted for the changes in current supplies of λ, λ' and all $\lambda \in \Lambda$, respectively;

$\Delta\mathfrak{B}^{\max^{\text{LAST}}}$ - The maximum cumulative change in Δb of (a) λ, (b) λ' and (c) the sum of cumulative changes across all $\lambda \in \Lambda$, adjusted for the changes in supplies of λ, λ' and all $\lambda \in \Lambda$, respectively, based on and relative to the inter-block MEV manipulation resistant last values in the Multi-Flow *Pump* for each λ at the end of the previous block;

$\mathfrak{K}$ - The lookback period in seconds of $f^{\$^{\lambda'}}(\mathfrak{K})$;

$\mathfrak{Q}$ - The number of intervals in the *Morning*;

$\mathfrak{w}$ - A *Well*;

$\mathfrak{w}_{\text{Ͽ}}^{*}$ - The optimal number of Pinto in $\mathfrak{w}$ at the end of the current transaction;

$\mathfrak{w}_{\text{Ͽ}}$ - The number of Pinto in $\mathfrak{w}$ at the end of the current transaction;

$\mathfrak{w}_{\text{Ͽ}}$ - The number of non-Pinto in $\mathfrak{w}$ at the end of the current transaction;

$\mathfrak{w}_{<\text{Ͽ}}^{*}$ - The optimal number of Pinto in $\mathfrak{w}$ at the beginning of the current transaction;

$\mathfrak{w}_{<\text{Ͽ}}$ - The number of Pinto in $\mathfrak{w}$ at the beginning of the current transaction;

$\mathfrak{w}_{<\text{Ͽ}}$ - The number of non-Pinto in $\mathfrak{w}$ at the beginning of the current transaction;

$\mathfrak{w}_{,\text{Ͽ}}^{\text{LAST}*}$ - The optimal number of Pinto in $\mathfrak{w}$ at the end of the last block;

$\mathfrak{w}_{,\text{Ͽ}}^{\text{LAST}}$ - The number of Pinto in $\mathfrak{w}$ at the end of the last block;

$\mathfrak{w}^{\text{LAST}}_{\blacklozenge,\text{Ↄ}}$ - The number of non-Pinto in $\mathfrak{w}$ at the end of the last block;

$\mathfrak{w}^{\text{EMA}^*}_{\circledast,\text{Ↄ}}$ - The optimal number of Pinto in a *Well* based on the instantaneous reserves in the Multi Flow *Pump*;

$\mathfrak{w}^{\text{EMA}}_{\circledast,\text{Ↄ}}$ - The number of Pinto in a *Well* based on the instantaneous reserves in the Multi Flow *Pump*;

$\mathfrak{w}^{\text{EMA}}_{\blacklozenge,\text{Ↄ}}$ - The number of non-Pinto in a *Well* based on the instantaneous reserves in the Multi Flow *Pump*;

$\mathfrak{w}^{\circledast^*}_{\overline{t-1}}$ - The optimal time weighted average number of Pinto in $\mathfrak{w}$ over the previous *Season*;

$\mathfrak{w}^{\text{SMA}}_{\circledast,t-1_0,\text{Ↄ}}$ - The inter-block MEV manipulation resistant TWA Pinto reserves in the Multi Flow *Pump* of $\mathfrak{w}$ from the beginning of the previous *Season* to the current transaction;

$\mathfrak{w}^{\text{SMA}}_{\blacklozenge,t-1_0,\text{Ↄ}}$ - The inter-block MEV manipulation resistant TWA non-Pinto reserves in the Multi Flow *Pump* of $\mathfrak{w}$ from the beginning of the previous *Season* to the current transaction;

$\mathfrak{w}^{\text{LAST}^*}_{\Xi-1}$ - The maximum remaining *Conversions* of $\mathfrak{w}$ permitted before a penalty is accrued;

$\mathfrak{w}^{@}$ - The *Well* token address;

$\mathfrak{w}^*$ - The data associated with the *Well Function* of $\mathfrak{w}$;

$\mathfrak{w}^{\min}$ - The minimum number of $\mathfrak{w}$ received from a *Conversion*;

$\mathfrak{w}^{\min(\circledast)}$ - The minimum number of Pinto required in a *Well* in order to calculate PDV or Δb;

$\frac{\partial \mathfrak{w}}{\partial \circledast}$ - The inter-block MEV manipulation resistant derivative of the $\mathfrak{w}$ LP token supply with respect to Pinto;

12.14.4 Mathscr Style Latin Alphabet Variables

$\mathscr{B}^{\lambda^*}$ - The optimal percent of *Deposited* PDV of λ;

$\mathscr{B}^{\lambda'^*}$ - The optimal percent of *Deposited* PDV of λ';

$\mathscr{B}^{\lambda'}$ - The percent of *Deposited* PDV of λ';

$\mathscr{C}_t$ - The *Crop Ratio* during t;

$\mathscr{C}^{\max}$ - The maximum *Crop Ratio*;

$\mathscr{C}^{\min}$ - The minimum *Crop Ratio*;

$\mathscr{C}^{\text{rain}}$ - The *Crop Ratio* when it is *Raining*;

$\mathscr{L}^{\lambda}_t$ - The *LP Seed Gauge Points* of λ during t;

$\mathscr{L}^{\lambda}_{\circ}$ - The initial *LP Seed Gauge Points* of λ;

$\mathscr{L}^{\lambda'^{\max}}$ - The maximum *LP Seed Gauge Points* of λ';

$\mathscr{L}^{\lambda'^{\text{close}}}$ - The percent level below which the *LP Seed Gauge Points* of λ' is considered close to optimal;

$\mathscr{L}^{\lambda'^{\text{far}}}$ - The percent level above $\mathscr{L}^{\lambda^{\text{close}}}$ and below which the *LP Seed Gauge Points* of λ' is considered far from optimal;

$\mathscr{L}^{\lambda'^{\text{vfar}}}$ - The percent level above which the *LP Seed Gauge Points* of λ' is considered very far from optimal;

$f^{\mathscr{L}^{\lambda}}$ - The function to calculate the *LP Seed Gauge Points* of λ;

$f^{\mathscr{L}^{\lambda'}}$ - The function to calculate the *LP Seed Gauge Points* of λ';

$f^{\mathscr{W}^{\lambda}}$ - The function to calculate the *Liquidity Weight* of λ;

$f^{\mathscr{W}^{\lambda'}}$ - The function to calculate the *Liquidity Weight* of λ';

$\mathscr{S}_t$ - The total *Seed Gauge Points* during t;

$\mathscr{S}_t^{\lambda}$ - The *Seed Gauge Points* of λ during t;

$\mathscr{T}$ - The number of *Seasons* for a new *Deposit* with an average number of *Seeds* per PDV, averaged across all *Deposits* in the *Silo* that are not *Germinating*, to catch up to the average *Mown Stalk* per PDV across all *Deposits* in the *Silo*;

$\mathscr{V}_t^{\lambda}$ - The *LP Seed Gauge Points* per PDV of λ during t;

$\mathscr{X}_t$ - The *Crop Scalar* during t;

$\mathscr{X}^b$ - Absolute change to the *Crop Scalar*;

$\mathscr{X}^m$ - Relative change to the *Crop Scalar*;

12.14.5 Greek Alphabet Variables

η - *Earned* ;

η_t^c - The *Plantable Seeds* associated with a *Stalkholder's* η that can be *Planted* to start earning *Grown Stalk* during t;

Λ - The *Deposit Whitelist*;

λ - and other assets on the *Deposit Whitelist*;

$\lambda^{\text{SMA}}_{,t-1_0,\eth}$ - The inter-block MEV manipulation resistant TWA non-Pinto reserves in the Multi Flow *Pump* of λ from the beginning of the previous *Season* to the current transaction;

$\lambda^{\text{SMA}}_{\lambda,t-1_0,\eth}$ - The inter-block MEV manipulation resistant TWA number of λ LP tokens from the beginning of the previous *Season* to the current transaction;

$\lambda^{\text{LAST}^*}_{\Xi-1}$ - The maximum remaining *Conversions* of λ permitted before a penalty is accrued;

λ^{ρ} - The LP token supply change cap per block set in Multi Flow;

$f^{\lambda\to\lambda'}(z^{\lambda})$ - The function to determine the number of λ' received for *Converting* a given number of λ;

$f^{\lambda\to\lambda'}(z^{\lambda}\to z^{\lambda'})$ - The function to determine the number of *Mown Stalk* burned for *Converting* a given number of λ to λ';

σ - A control variable used to calculate the *Temperature* during the *Morning*;

ρ - The exchange rate cap per block set in Multi Flow; and

ϕ - PINTO:USDC LP tokens.

12.14.6 Glyph Variant Greek Alphabet Variables

$\varkappa^{\lambda}$ - The last *Season* a *Stalkholder Mowed* their *Grown Stalk* from λ *Deposits*;

ϑ - PINTO:WETH LP tokens;

φ - PINTO:cbETH LP tokens;

ϱ - PINTO:cbBTC LP tokens; and

ϖ - PINTO:WSOL LP tokens.

12.14.7 Symbol Variables

❀ - Pinto;

❀$^{@}$ - The Pinto contract address;

❀$^{@}$ - The Pinto token address;

❀$^{\text{min}}$ - The minimum number of ❀ received from a *Conversion*;

$f^{\$^{\lambda}}$ - The function to calculate the USD price of the non-Pinto asset in λ;

$f^{\$^{\mathfrak{w}}}$ - The function to calculate the USD price of the non-Pinto asset in $\mathfrak{w}$;

$f^{\$^{\lambda'}}(\mathfrak{K})$ - The function to calculate the USD price of the non-Pinto asset in λ' given a lookback period $\mathfrak{K}$;

$f^{\$}(z^{\lambda})$ - The function to calculate the time weighted average non-Pinto USD-denominated value in a given number of *Deposited* λ; and

☼ - The minimum Pinto reward for successfully calling the `gm` function.