



Sweetbridge Liquidity Protocol: Mathematical Specifications

Authors:

Dr. Michael Zargham, CEO @ BlockScience

Aleksandr Bulkin, Co-Founder @ CoinFund

Hui Huang, CTO @ Sweetbridge

J. Scott Nelson, CEO @ Sweetbridge

Publication Version 1.0

This paper provides a detailed formal specification for the *Sweetbridge Liquidity System*, an ecosystem of Ethereum smart contracts and infrastructure services that enable a digital stable-valued currency economics in which the money supply is formed from and supported by deposits of valuable collateral assets. Liquidity generated by this framework is intended to be cheaper and more accessible than traditional bank loans.

In this version of the specification, the vault mechanisms enabling UOUs are formally specified and analyzed, including the activation of Sweetcoin to generate interest-free loans of Bridgecoin. The fiat exchange mechanisms and Bridgecoin demand-generating mechanisms are described; formal analysis of these mechanisms and the overall Bridgecoin price stability conditions will be available in the next release. Monte Carlo simulations of the mathematical system model and eventually empirical analysis of the live Sweetbridge economy are part of the broader analytical roadmap aimed at instantiating and maintaining a healthy Sweetbridge economy.

Note on Presentation	4
1 Motivation	5
2 Notation	6
2.1 Currencies	6
2.2 States	6
2.3 Prices	6
2.4 Vaults	6
3 Assets, Vaults, and Loans	7
3.1 Collateral	7
3.2 Collateral Vaults	7
3.3 Locked Collateral	8
3.4 MVP Implementation: Ethereum and Sweetcoin Collateral	9
3.5 Excess Liabilities	10
3.6 Example: Balloon Payments	11
3.7 Strictly Valid Repayment Schedules	11
3.8 The Sell Line	12
3.9 MVP Implementation: Sell Line	13
3.10 Enforcing the Sell Line	16
3.11 User-Triggered Asset Sales	17
3.12 Automated Bridgecoin Repurchasing	17
3.13 Direct Collateral Purchase	17
3.14 MVP Implementation: Systemwide Borrowing Limit	17
4 Function of Sweetcoin	19
4.1 Sweetcoin Activation	19
4.2 Offsetting Liabilities with Sweetcoin	19
4.3 Global Limits on Fee Elimination	20
4.4 Interest-Free Loans	21
4.5 Operating Revenue from Loans	21
4.6 Effect of Activated Sweetcoin on the Borrowing Capacity of Asset Vaults	22
5 Sweetcoin Availability	23
5.1 Sweetcoin Supply and Crowdsale	23
5.2 Convergent Drip Crowdsale	23
6 Sweetcoin Utility	25
6.1 Time-Based Utility of Sweetcoin	25
6.2 Sweetcoin Fair Value	26

6.2.1	Time-Discounted Value of Money	26
6.2.2	Commercial Real Estate Model	27
6.2.3	Comparing the Two Approaches	27
6.3	Characterizing Sweetcoin Hype	28
6.4	Use of Crowdsale Proceeds	28
7	Additional Financial Mechanisms	30
7.1	Fiat Exchange	30
7.2	Liquidity Pool	30
7.3	Incentives to Use Bridgecoin	30
7.4	Crowdsales and the Bridgecoin Queue	31
7.5	Discounts for Bridgecoin Users	31
7.6	Discount Accounts	31
7.7	Market Makers, Trust Intermediaries, and Risk Managers	31
7.8	The Bridgecoin Community	32
7.9	Reestablishing Bridgecoin Supply	32
8	Economic System Dynamics	33
8.1	Collateral Volatility	33
8.2	Stabilizing Market Forces	33
8.3	Stabilizing via Creation and Redemption	34
8.4	Ongoing Work	34
9	APPENDIX A: Summary of System Control Parameters	37
9.1	Asset Vault Interest Rate	37
9.2	Collateralization Coefficient	37
9.3	Sell Line	37
9.4	Sell Line Shape Parameter	37
9.5	Sweetcoin Release Fraction	37
9.6	Sweetcoin Activation Sensitivity	37
9.7	Debt Ceiling	38
9.8	Maximum Redemption Rate	38
10	APPENDIX B: Proof of Theorem 1: The Sell Line Assertion	39
	Bibliography	40

Note on Presentation

This document is a work in progress and will be updated as the overall design is populated with formal details. To distinguish between intention and formal specification, those sections that have not been finalized are indicated in brown as follows: “Example of text that has not been finalized.”

Additionally, in some places, we will present informal requirements and motivations, indicated in blue as follows: “Requirement text.”

1 Motivation

The economics of supply chain today are characterized by a prevalent shortage of liquid cash. Consequently, in order to generate value, organizations must often take out loans at costs that significantly diminish their profitability. In many cases, the cost of capital becomes a determining factor in whether an organization can operate at all.

The Sweetbridge vision is best summarized as follows: *if there is value in assets held by an organization, it should be possible to represent this value in a liquid currency without resorting to expensive intermediary services.* Blockchain technology is what makes this possible. It allows one to create custom decentralized economic systems complete with their own rules for minting and transacting currency. It also removes the need for expensive and inefficient centralized lending services.

The Sweetbridge liquidity system described in formal detail in this document makes use of this approach to carry out its vision. It allows holders of blockchain-based assets (further defined in section 3.1) to maintain their ownership, while at the same time giving them access to disposable capital issued against them. This is achieved by introducing the Asset Vault smart contract (AV) and an issuance process called a *YOU* (You Owe You). This contract issues a stable cryptocurrency when collateral is deposited and locked on the blockchain. When the loan is repaid, the collateral is returned under the control of the owner.

A proof-of-concept implementation in which only cryptocurrencies are used as collateral will be delivered first. It is designed to prepare the basis for a more general liquidity system based on a broad range of assets whose ownership can be reliably managed within a blockchain-based framework. The long-term goal of the system is to enable supply chain participants to generate liquidity using collateral assets such as equipment, real estate, commodities, and even accounts receivable.

Sweetbridge economics defines two cryptographic digital tokens: Bridgecoin and Sweetcoin. Bridgecoin is intended as an asset whose value is pegged to fiat currency. Sweetcoin is issued in limited supply and assigned the role of a discount token in the ecosystem. In other words, Sweetcoin is treated by the system as a coupon that gives its holders the right to receive valuable services at a discount or free of charge.

There are two ways in which Bridgecoin enters and leaves the supply: (1) through Asset Vault loans (Section 3), and (2) through Fiat Exchanges (Section 7.1). Together, these subsystems form the complete economics of Sweetbridge and serve as mechanisms to support the stability of Bridgecoin. This document will formally describe many aspects of the first path: token design, Asset Vault contracts, and the Sweetcoin crowdsale economics. We will then outline the second path briefly, along with the overall economic structure, and a preliminary discussion of USD-pegged stability dynamics of Sweetbridge. The formal specification of those parts of the system not fully detailed here will be presented in subsequent publications, including the long-term economic model that consists of multiple stable currencies representing fiat equivalents in multiple world jurisdictions.

This paper is the first in a series of documents intended to define and demonstrate analytically the properties of Sweetbridge economic components for the purpose of both implementing them and reasoning about their macro-scale implications. Readers are encouraged to familiarize themselves with [3], which contains detail about the Sweetbridge vision and will help place this document in a larger context.

2 Notation

In this section, a convention for denoting the form and state of currency is defined for use throughout the document.

2.1 Currencies

As noted above, the economic model makes reference to three currencies: the United States Dollar (USD) fiat currency, Bridgecoin (BRC), and Sweetcoin (SWC). In some places, when referring to collateral locked inside the liquidity system, we will also refer to cryptocurrencies, such as Ethereum (ETH) (see [1]) and Bitcoin (BTC) (see [2] for more information on Bitcoin). Networkwide variables related to these currencies will be captured in equations using capital letters D for USD, B for BRC, S for SWC and E for ETH and other cryptoassets. When referring to decision variables and computed values that are specific instances of loans provided by the liquidity protocol rather than to network aggregates, the lowercase letters d , b , s and e will be used, respectively.

2.2 States

The currencies can exist in different states. For example, Sweetcoin can be “activated” to reduce the fees associated with borrowing Bridgecoin. The standard structure employed will be $\text{currency}_{\text{time}}^{(\text{state})}$. For example, the amount of Sweetcoin activated in the network at time t would be denoted $S_t^{(\text{activated})}$. Additionally, for each collateral asset in a vault, the quantity of that collateral asset in its own unit is denoted $q_t^{(\text{collateral})}$.

2.3 Prices

Another important type of variable in our economic model is the price of each cryptocurrency relative to USD fiat. The common structure for these variables will be $P_{\text{time}}^{(\text{currency})}$; for example, the market price of Ethereum at time t is denoted $P_t^{(\text{ETH})}$. While the economics are defined in continuous time, implementation requires an understanding of sampling, whereby continuous time is treated as discrete blocks of size Δt . A slight abuse of notation, $P_k^{(\text{ETH})}$ refers to the market price of Ethereum at a discretely defined time k . The time t referenced by k will be well-defined in context. We leave out the exact length of Δt as an implementation detail that doesn’t have bearing on the formal structure of the system.

2.4 Vaults

Vaults are defined as sets. Following standard mathematical notation, an arbitrary vault is denoted \mathcal{V} . Every vault \mathcal{V} is made up of individual items or elements of collateral denoted v , the properties of which are outlined in Section 3.1. Complexity arises when the Sweetbridge networkwide states are considered; these variables are defined over the set of all vaults in the network. In order to handle the set of sets, the notation $\mathcal{V} \in \mathbb{V}$ is introduced such that \mathbb{V} is the set of all vaults \mathcal{V} in the network. While the \mathbb{V} set is time-varying as a result of users creating new vaults, reference to time is suppressed, and \mathbb{V} is always used to compute system parameters defined at the current time t .

3 Assets, Vaults, and Loans

The Sweetbridge Liquidity system is loosely based on the ancient concept of banknotes, which are lightweight promissory notes evidencing deposits of valuable goods. The liquidity cryptocurrency, Bridgecoin, is issued temporarily in exchange for another valuable blockchain-based asset that gets locked until it is fully repaid. Bridgecoin is created and destroyed as the amount of collateral in the system increases and shrinks. This process is termed a UOU, denoting that the loan is not made against a counterparty, but rather against a decentralized network able to issue currency as needed. Repaying liquidity loans so created, a user returns a locked collateral into his/her own control, in effect repaying him/herself¹.

3.1 Collateral

Collateral is defined as an item of value that can be managed on blockchain. Examples include: cryptocurrencies or other transferrable cryptoassets, transferrable stakes in other smart contracts, or real-world physical assets whose ownership can be tracked on blockchain in a legally enforceable way.

In mathematical terms, define the types of collateral accepted as a set \mathcal{C} . For any $c \in \mathcal{C}$, there is a market price $P_t^{(c)}$ for all time t . In addition to having a market price determined outside of the Sweetbridge system, every $c \in \mathcal{C}$ has a collateralization coefficient α_c that is set by Sweetbridge and informed by the measured historic volatility of the market price $P_t^{(c)}$. α_c represents the limit of how much BRC may be borrowed against collateral c .

3.2 Collateral Vaults

A Collateral Vault is a smart contract that takes control of an asset and issues Bridgecoin to the vault's owner. The smart contract prevents transfer of the collateral or some portion of it as long as the issued Bridgecoin remains outstanding.

A vault \mathcal{V} is a set characterizing a portfolio of assets such that any $v \in \mathcal{V}$ must have a collateral type $c_v \in \mathcal{C}$. In the early stages of the Sweetbridge system, all forms of collateral have a quantity that can be expressed numerically, but this distinction is made to guide the implementation toward extensibility to the cases of invoices and physical assets that are not fungible. In these cases, there will be distinct items v , which share a common collateral type c . Thus the fiat value of each item $v \in \mathcal{V}$ is defined as

$$d_t^{(v)} = q_t^{(v)} \cdot P_t^{(c_v)}. \quad (3.1)$$

Further note that the value of any collateral type c is

$$d_t^{(c)} = \sum_{v:c_v=c} q_t^{(v)} \cdot P_t^{(c_v)} \quad (3.2)$$

¹Section 7.1 details additional mechanisms for issuing Bridgecoin that are not related to UOUs

and the total USD-denominated value of the vault can be expressed as either

$$d_t = \sum_{c \in \mathcal{C}} \sum_{v: c_v=c} q_t^{(v)} \cdot P_t^{(c_v)} \quad (3.3)$$

$$= \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \quad (3.4)$$

for any vault \mathcal{V} . Additionally, vault \mathcal{V} has Bridgecoin borrowing power defined analogously; each item $v \in \mathcal{V}$ confers borrowing power $\alpha_{c_v} \cdot q_t^{(v)} \cdot P_t^{(c_v)}$. However, borrowing occurs against the whole portfolio of assets, not against a single asset, so the borrowing power of the vault is given as

$$b_t^{(limit)} = \sum_{c \in \mathcal{C}} \alpha_c \sum_{v: c_v=c} q_t^{(v)} \cdot P_t^{(c_v)} \quad (3.5)$$

$$= \sum_{v \in \mathcal{V}} \alpha_{c_v} \cdot q_t^{(v)} \cdot P_t^{(c_v)}. \quad (3.6)$$

From these values it is possible to compute an effective collateralization limit for the whole vault

$$\bar{\alpha}_t = \frac{b_t^{(limit)}}{d_t} \quad (3.7)$$

which will vary in time for vaults with non-zero quantities of more than one type of collateral; in case of one collateral type $\bar{\alpha}_t = \alpha_c$ for that one type c . In general, the effective collateralization limit is a vault-specific metric equal to a fiat-value-weighted-average of the collateralization coefficients for the supported collateral types $c \in \mathcal{C}$. Well-diversified vaults will see far less volatility in this metric.

Note that $\alpha_c < 1$ for all $c \in \mathcal{C}$, thus at all times t , $\bar{\alpha}_t < 1$, which is equivalent to the fact that the borrowing power $b_t^{(limit)} < d_t$ for any vault at any time. This protects the system by limiting the probability that a member borrows the quantity $b_t^{(limit)}$ at time t only to have the vault value d_t fall to $d_\tau < b_t^{(limit)}$ at some future time τ . To further protect against loans entering this state, the Asset Vault smart contracts are equipped with a sell line. The sell line will be further discussed in section 3.8.

3.3 Locked Collateral

When borrowing loans are taken against collateral vaults, the collateral within those vaults is no longer completely accessible by the vault owner. Due to the portfolio nature of vaults, it is not the currency itself that is locked, but rather a limitation is placed on the legal withdrawal actions allowable via the vault smart contract. Specifically, legal withdrawals from vaults are defined by equation (3.6), where the current Bridgecoin liability $b_t^{(owed)}$ takes the place of the borrowing power

$$b_t^{(owed)} \leq \sum_{v \in \mathcal{V}} \alpha_{c_v} \cdot q_t^{(v)} \cdot P_t^{(c_v)}. \quad (3.8)$$

In other words, any attempt to withdraw a quantity of Δq_v for collateral v will only be allowed if

$$b_t^{(owed)} \leq b_{t+}^{(limit)} \quad (3.9)$$

$$\leq b_t^{(limit)} - \alpha_{c_v} \cdot \Delta q_v \cdot P_t^{(c_v)} \quad (3.10)$$

where $t+$ denotes the state variable at time t adjusted for the sale of Δq_v assets. At any time, the share of the assets in the vault that are *locked* is the ratio ω_t , the debt to borrowing power

$$\omega_t = \min \left(\frac{b_t^{(owed)}}{b_t^{(limit)}}, 1 \right) \quad (3.11)$$

where the min operation indicates that all assets are locked if the debt is in excess of the borrowing power. When discussing a locked share of a vault containing a portfolio of diverse collateral assets, we think of this percentage as the percentage of the dollar-denominated value of these assets.

While the vault is in the fully locked state, assets may not be removed, but they may be applied directly against the outstanding debt; see User-Triggered Asset Sales in Section 3.11. Should the value of the collateral in the vault continue to decline with respect to the liabilities, the sell line will trigger preventing the vault from falling into a default state where the debt exceeds the asset values; further detail in Section 3.8.

3.4 MVP Implementation: Ethereum and Sweetcoin Collateral

Under the initial implementation, the set $\mathcal{V} = \{ETH, SWC\}$, meaning that the only two elements in the set are collateral types Ethereum and Sweetcoin². Furthermore, since both Ethereum and Sweetcoin are fungible and divisible, it suffices to define each vault as having only having two elements e and s denoting Ethereum and Sweetcoin respectively, with $q_t^{(ETH)}$ denoting the full quantity of Ethereum deposited in the vault and $q_t^{(SWC)}$ denoting the full quantity of Sweetcoin deposited.

Under this set of simplifying assumptions, equation (3.3) simplifies to

$$d_t = q_t^{(ETH)} \cdot P_t^{(ETH)} + q_t^{(SWC)} \cdot P_t^{(SWC)} \quad (3.12)$$

and equation (3.6) simplifies to

$$b_t^{(limit)} = \alpha_{ETH} \cdot q_t^{(ETH)} \cdot P_t^{(ETH)} + \alpha_{SWC} \cdot q_t^{(SWC)} \cdot P_t^{(SWC)}. \quad (3.13)$$

The effective collateralization coefficient $\bar{\alpha}_t$ falls in the interval $[\alpha_{ETH}, \alpha_{SWC}]$, achieving the lower limit for a vault entirely comprised of Ethereum and rising to reach the upper limit as the portfolio shifts toward Sweetcoin only. This metric is included as a means of showing users the benefits they receive by collateralizing Sweetcoin. These variations in $\bar{\alpha}_t$ only occur as a result of the shifting weight of the fiat value of the vault's assets amongst collateral with different collateralization coefficients.

²Sweetbridge may add Bitcoin as collateral in short order. The reader should have no trouble extending the specification presented here to the case of three assets.

3.5 Excess Liabilities

Loans against an Asset Vault follow a standard continuously compounding interest model with the interest rate r defined as a global system parameter of the Sweetbridge economic system. Interest is charged as an increase in the vault's liabilities over time. This excess in liability constitutes the fee users pay for the services provided by Sweetbridge and can be offset through a discount model described in section 4.

A loan taken against an Asset Vault will be repaid at the option of the vault holder within constraints put forth by Sweetbridge. The general case of this loan and repayment can be stated in terms of discrete time periods Δt , and a total time period T equal to the length of the loan. By selecting a Δt , the interest rate r can be expressed as γ , the interest rate per time period Δt , which is generally more understandable to consumers: liabilities grow by the fraction γ every time a period Δt passes. The value of γ is computed from the underlying continuously compounding rate r and the time frame of choice Δt as follows:

$$\gamma = \gamma(r, \Delta t) = \exp(r\Delta t) \tag{3.14}$$

where $\exp(x)$ denotes³ the function of raising Euler's number 'e' to the power x .

Denoting the period of the loan in intervals k and total number of such intervals K , the repayment schedule can be defined as $b_k^{(payment)}$ for $k \in \{1, \dots, K\}$. The repayment plan is denoted as a vector $\vec{b} \in \mathbb{R}_+^K$. The choice of b denotes that payments are made in BRC. Using our k index for time, and t_0 as the timestamp of the initial loan, the mapping $t = t_0 + k\Delta t$ maps our discrete time view of the loan to continuous time. For a loan with principal b , the liabilities accrued accounting for payments made are given by the iteration

$$b_0^{(owed)} = b \tag{3.15}$$

$$b_1^{(owed)} = b + \gamma b - b_1^{(payment)} \tag{3.16}$$

$$b_k^{(owed)} = (1 + \gamma) \cdot b_{k-1}^{(owed)} - b_k^{(payments)}. \tag{3.17}$$

A valid payment schedule must have the following property

$$b_K^{(owed)} = b + \sum_{k=1}^K \left(\gamma b_{k-1}^{(owed)} - b_k^{(payments)} \right) = 0 \tag{3.18}$$

which allows us to express the total Bridgecoin-denominated repayment as

$$\sum_{k=1}^K b_k^{(payments)} = b + \sum_{k=1}^K \gamma b_{k-1}^{(owed)} \tag{3.19}$$

³The exponential notation is used to avoid confusion with the designation of e for quantities of Ethereum in this document.

and thus the total excess liabilities $L(b, \vec{b})$ for borrowing b Bridgecoin and repaying according to \vec{b} are

$$L(b, \vec{b}) = \sum_{k=1}^K \gamma b_{k-1}^{(owed)} \quad (3.20)$$

computed by following the interest accrual trajectory over the in period $t = t_0$ to $t = t_0 + T$ using the discrete period $k = 0$ to $k = K$. Note that this equation determines the liabilities for any given repayment schedule \vec{b} , over any number K of periods of length Δt . If fees are estimated with one repayment schedule, and a different repayment schedule is realized, the actual excess liabilities are based on the true repayment schedule, not on the planned repayment schedule.

3.6 Example: Balloon Payments

Let us consider a simple balloon payment plan supported with a single payment of all liabilities at time $T = t_0 + K\Delta t$. The total liabilities (denominated in Bridgecoin) are given by the compounding interest equation $b \cdot (1 + \gamma)^K$ and the excess liabilities resolve to

$$L(b, \vec{b}) = b \cdot (1 + \gamma)^K - b \quad (3.21)$$

derived from equation (3.20) with \vec{b} set to the balloon payment plan characterized elementwise as

$$\left[\vec{b} \right]_k = b_k^{(payment)} = \begin{cases} k = K & L(b, \vec{b}) \\ \text{else} & 0 \end{cases} \quad (3.22)$$

where the bracket notation $\left[\vec{b} \right]_k$ is used to indicate the k^{th} element of the vector \vec{b} .

The detailed specification of how compounding interest loans work in the Sweetbridge economy is critical because these liabilities can be offset through the activation of Sweetcoin. This will be addressed in Section 4.

3.7 Strictly Valid Repayment Schedules

Another case is to allow repayments over time but impose a restriction ensuring the debt cannot accumulate. Define a valid repayment schedule as any repayment schedule that does not increase the principal, i.e. at the very least, fees are repaid every period. This imposes an additional requirement on the repayment schedules \vec{b} , which is defined for each individual interval Δt ,

$$\left[\vec{b} \right]_k = b_k^{(payment)} \geq \gamma b_{k-1}^{(owed)}. \quad (3.23)$$

Enforcing this on an a per-payment-period interval ensures that the liabilities are not increasing,

$$b_k^{(owed)} \leq b_{k-1}^{(owed)}. \quad (3.24)$$

This is an alternative payment scheme from the balloon payment example, and it is, in fact, strictly inconsistent with the balloon payment scheme because the balloon payment plan violates equation (3.23) at every time interval except the final time interval K . As the project matures, a variety of payment schedules are expected to be supported, with different rules.

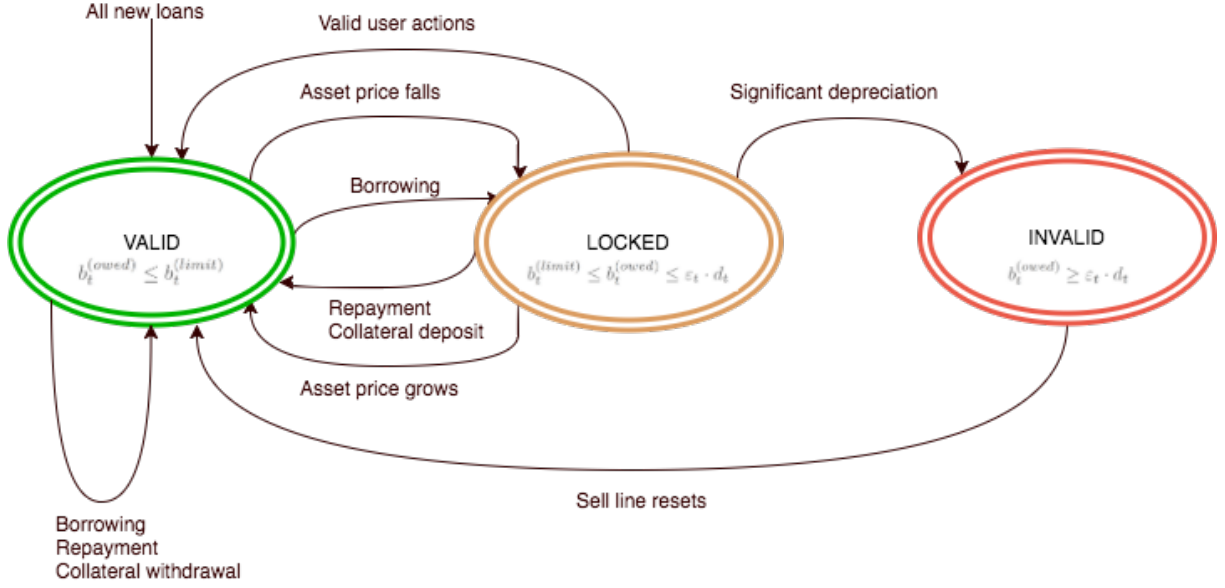


Figure 3.1: Summary of the Asset Vault states and user actions.

3.8 The Sell Line

The sell line is a system parameter in the Sweetbridge economic system, defined by a global rule computed individually for each vault. The purpose of the sell line is to protect loans from going underwater in the sense that the total outstanding liabilities (denominated in Bridgecoin) against a vault might become greater than the USD-denominated value of the vault itself, which would remove the incentives for users to repay their loans. The sell line is a provision in the Asset Vault smart contract that ensures that $b_t^{(owed)} < d_t$ for that vault at any time t . Since $b_t^{(owed)}$ can vary in time due to the accrual of liabilities, and d_t can vary due to changes in market price of the assets, it is possible that the vault might violate this invariant. Should this invariant be violated, the smart contract will sell some or all of the assets immediately, paying down the balance to achieve a valid state.

In practice, $b_t^{(owed)} < d_t$ must be enforced with slack to account for volatility in price and the fact that the price of assets may continue to move after the event is triggered, but before the liquidation of assets is completed, rectifying the state of the vault. Therefore, the sell line condition is defined

$$b_t^{(owed)} < \epsilon_t \cdot d_t \quad (3.25)$$

where $\epsilon_t = \epsilon_{t,\mathcal{V}}$ is a sell line function computed for each vault based on its riskiness. Having established the vault-specific context of this coefficient, we drop the extra subscript \mathcal{V} referencing the vault. Likewise, the riskiness $\eta_t = \eta_{t,\mathcal{V}}$ of any vault \mathcal{V} is defined

$$\eta_t = \frac{\sum_{v \in \mathcal{V}} q_t^{(c_v)} P_t^{(c_v)} \alpha_{c_v}^{-1}}{\sum_{v \in \mathcal{V}} q_t^{(c_v)} P_t^{(c_v)}} \quad (3.26)$$

noting this quantity is a weighted averaged of the riskiness of each type of asset, $\alpha_c^{-1} = 1/\alpha_c$. The greatest value η_t can take for any vault is $1/\alpha_{min}$, and the smallest value is $1/\alpha_{max}$ where $\alpha_{min} = \min_{c \in \mathcal{C}} \alpha_c$ and $\alpha_{max} = \max_{c \in \mathcal{C}} \alpha_c$.

The sell line function can then be given by

$$\epsilon_t = \left(\frac{\theta + 1}{\theta + \eta_t} \right) \quad (3.27)$$

where θ is a shape parameter. Recognizing that η_t^{-1} is analogous to a portfolio-level collateralization coefficient, it is possible to explore choices and select a shape parameter.

It is necessary to ensure that the parameter ϵ_t computed according to equation guarantees that the sell line criteria is always stronger than the loan initialization criteria. That is to say, any vault satisfying

$$b_t^{(owed)} \leq b_t^{(limit)} \quad (3.28)$$

where $b_t^{(limit)}$ is defined in equation (3.6), also satisfies

$$b_t^{(owed)} \leq \epsilon_t d_t \quad (3.29)$$

and suffices to assert equation (3.28) to enforce equation (3.29), as long as $\theta \geq 0$.

Theorem 1. *Any vault \mathcal{V} satisfying the valid new loan condition defined in equation (3.28) will always satisfy the sell line condition in equation (3.29) when the vault-dependent sell line is set using equation (3.27), where shape parameter θ is selected such that $\theta > 0$, and the risk rating of the vault is η_t as defined by equation (3.8).*

Proof is relegated to Appendix 10. Furthermore, any withdrawals of collateral that would violate (3.28) are disallowed, further guaranteeing that withdrawals cannot trigger the sell line.

3.9 MVP Implementation: Sell Line

The MVP implementation of the Asset Vaults only has Ethereum and Sweetcoin as collateral, so the sell line equations simplify to represent a weighted average of the risk levels associated with these assets. Given collateralization coefficients α_{ETH} and α_{SWC} , the risk value for any vault is given by

$$\eta_t = \frac{q_t^{(ETH)} P_t^{(ETH)} \alpha_{ETH}^{-1} + q_t^{(SWC)} P_t^{(SWC)} \alpha_{SWC}^{-1}}{q_t^{(SWC)} P_t^{(SWC)} + q_t^{(ETH)} P_t^{(ETH)}} \quad (3.30)$$

which can be interpreted as a fiat-denominated value weighted average of the risk factors. The sell line can fall anywhere in the following interval

$$\left(\frac{\theta + 1}{\theta + \alpha_{ETH}^{-1}} \right) \leq \epsilon_t \leq \left(\frac{\theta + 1}{\theta + \alpha_{SWC}^{-1}} \right) \quad (3.31)$$

achieving the lower limit when the vault only contains Ethereum and achieving the upper limit for a vault only containing Sweetcoin. This holds because equation (3.27) is monotonically decreasing in the risk factor η_t defined in equation (3.8), and the risk coefficients are set such that $\alpha_{SWC}^{-1} < \alpha_{ETH}^{-1}$. Note that the relationship proven in Theorem 1 can be checked in the MVP case by comparing the effective collateralization $\bar{\alpha}_t$ and sell line ϵ_t for any ratio of Ethereum to Sweetcoin in the vault.

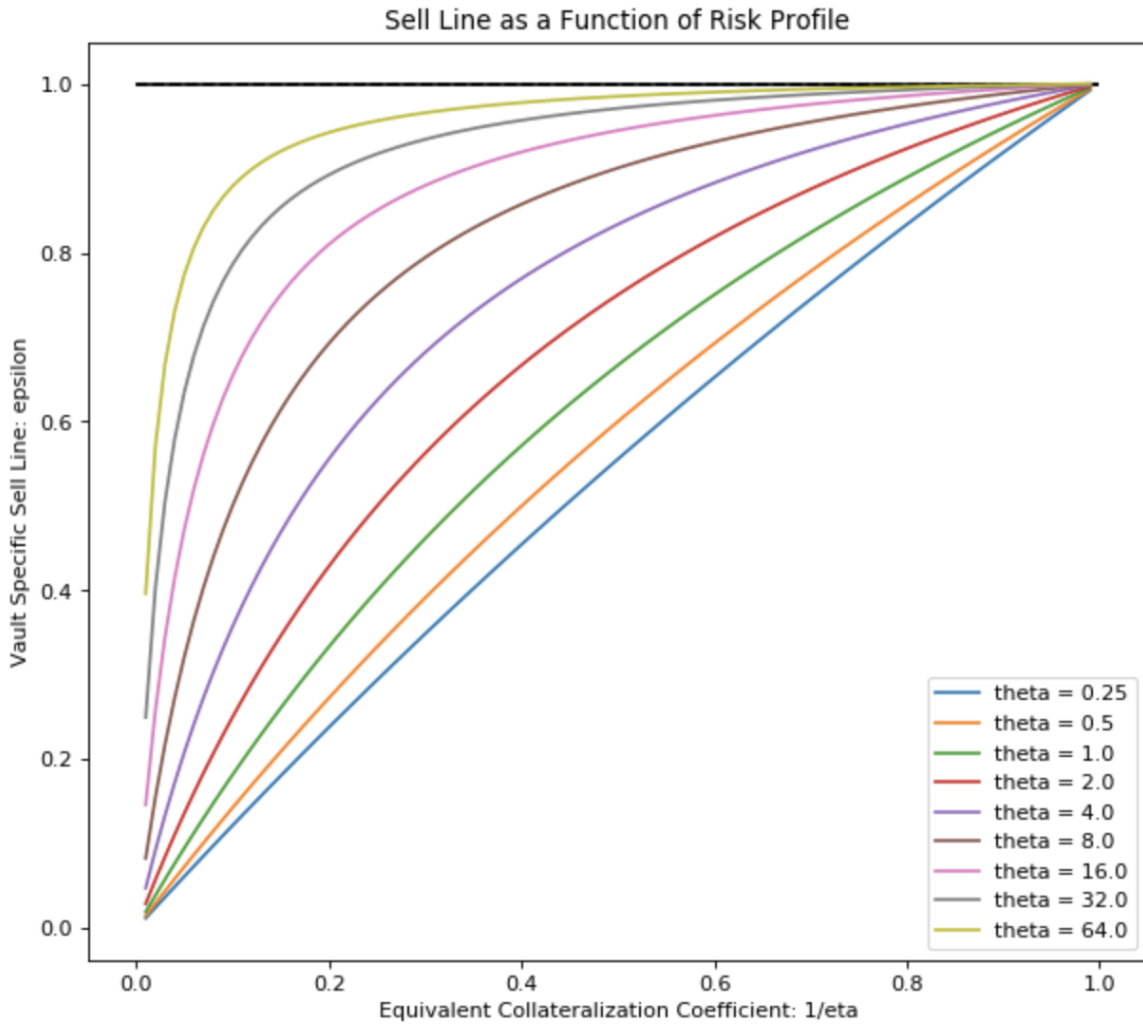


Figure 3.2: Visualization of the shape parameter θ in setting the sell line based on riskiness η computed according to equation (3.8)

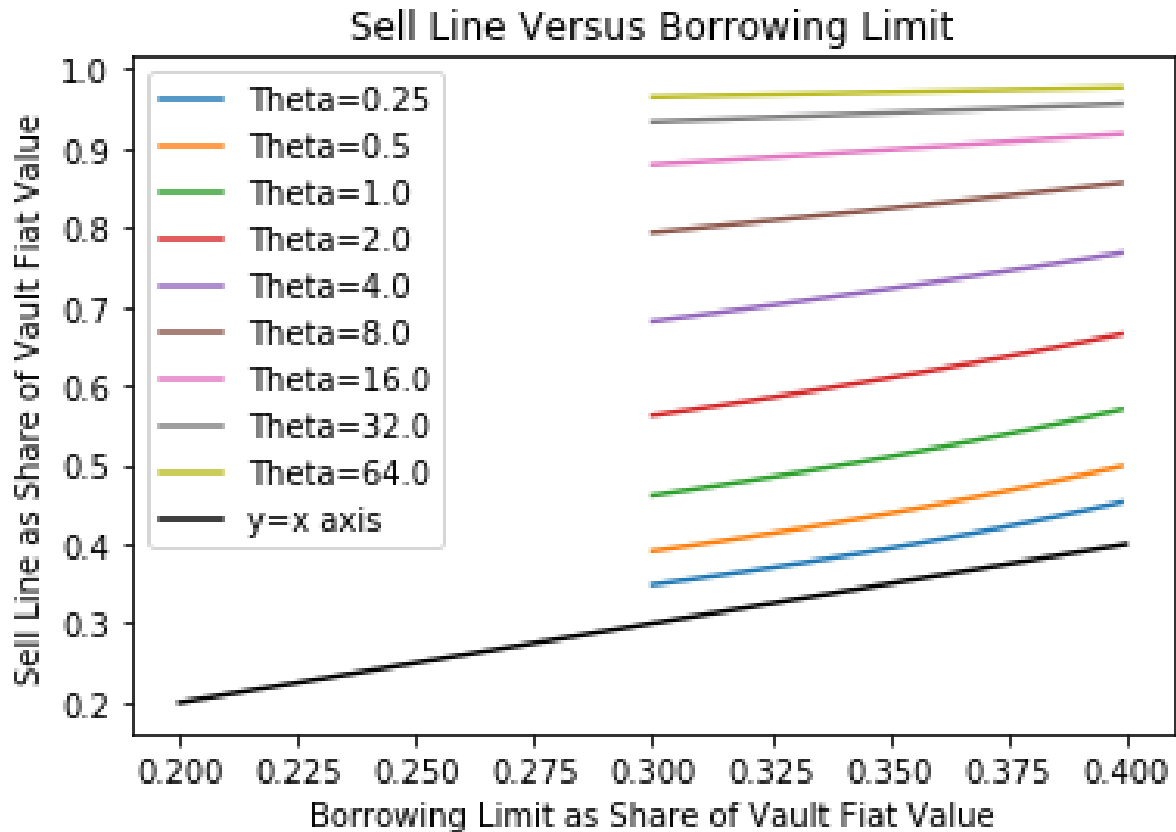


Figure 3.3: Visualization showing that the sell line remains strictly greater than the borrowing power, and how the choice of θ effects the size of the margin.

3.10 Enforcing the Sell Line

Validating the state of a vault with respect to the sell line requires that each oracle establish the current market price $P_t^{(c)}$ for each type of collateral c in the vault, and the current fiat value of the vault, d_t is computed according to equation (3.3); if $b_t^{(owed)} < \epsilon_t d_t$, the vault is in a valid state and no further action is required. However, if $b_t^{(owed)} \geq \epsilon_t d_t$, then the vault is in an invalid state, and corrective action is triggered. At this point, it is not sufficient simply to cross the sell line, because this could result in repeatedly triggering and correcting over very short periods of time. Instead, the corrective logic forces the vault into a valid state for a new loan in accordance with equation (3.28). Define the correction to be Δq_v for each v in the vault. This correction is treated as happening instantaneously at time t , so the update is accounted for using the notation t and $t+$ denoting the prior and posterior states, respectively. Specifically, the amount owed prior to the correction is $b_t^{(owed)}$ and the amount owed immediately after the correction is given by

$$b_{t+}^{(owed)} = b_t^{(owed)} - \sum_{v \in \mathcal{V}} P_t^{(c_v)} \Delta q_v \quad (3.32)$$

but in selling these assets the value of the vault d_t has been changed to

$$d_{t+} = d_t - \sum_{v \in \mathcal{V}} P_t^{(c_v)} \Delta q_v \quad (3.33)$$

and the borrowing power against that vault has become

$$b_{t+}^{(limit)} = \sum_{c \in \mathcal{C}} \alpha_c \sum_{v: c_v=c} (q_t^{(v)} - \Delta q_v) P_t^{(c_v)}. \quad (3.34)$$

Thus to return the vault to a valid state, it is required that the values Δq_v satisfy the linear inequality $b_{t+}^{(owed)} \leq b_{t+}^{(limit)}$, expanded as

$$b_t^{(owed)} - \sum_{v \in \mathcal{V}} P_t^{(c_v)} \Delta q_v \leq \sum_{c \in \mathcal{C}} \alpha_c P_t^{(c)} \sum_{v: c_v=c} (q_t^{(v)} - \Delta q_v) \quad (3.35)$$

which further simplifies to

$$b_t^{(owed)} - \sum_{v \in \mathcal{V}} P_t^{(c_v)} \alpha_{c_v} \leq \sum_{v \in \mathcal{V}} (1 - \alpha_{c_v}) P_t^{(c_v)} \Delta q_v. \quad (3.36)$$

Observing that equation (3.36) is a linear inequality constraint in the decision variable Δq_v , there is an infinite family of valid solutions. This family of solutions can be reduced to a single solution by a user-specified liquidation rule set at the time of the original loan. Candidate liquidation rules include any collateral preference scheme whereby the user ranks collateral items v in some order, and the smart contract liquidates the assets in the order specified. The quantities Δq_v required can be precomputed by setting $\Delta q_v = q_t^{(v)}$ for each v in the ranked order until (3.36) is satisfied.

Furthermore, the collateral v^* that crosses the threshold may be partially liquidated

$$\Delta q_{v^*} = \frac{b_t^{(owed)} - \sum_{v \in \mathcal{V}} P_t^{(c_v)} \alpha_{c_v} - \sum_{v < v^*} (1 - \alpha_{c_v}) P_t^{(c_v)} q_t^{(v)}}{(1 - \alpha_{c_{v^*}}) P_t^{(c_{v^*})}} \quad (3.37)$$

where $v < v^*$ indicates collateral items v ranked before v^* .

It is not practical for the user to choose a ranking at the time the sell line is triggered, so preferences must be set in advance. In addition to user preselected ranking for the preferential liquidation scheme, Sweetbridge could determine the ranking at the time of the liquidation using ranking oracles that compute rankings based on user prespecified schemes. Suggested schemes include most depreciated assets liquidated first, most appreciated assets liquidated first, or lowest growth rate assets liquidated first. This method for returning vaults to valid states does not care what ranking is used, so ranking scheme options should be determined as part of product requirements.

3.11 User-Triggered Asset Sales

While the sell line is a network protection mechanism that prevents vaults from entering a state of default, a proactive user may wish to pay down debts by directly cashing in collateral. Functionally, this user action has the same inputs, outputs and requirements as the sell line. The prior state of the vault is defined by quantities of each asset $q_t^{(v)}$ and their respective values $P_t^{(c_v)}$. At any time, a user could choose to cash in some collateral to pay down the balance by setting Δq_v for any $v \in \mathcal{V}$, as long as the criteria in equation (3.36) are met.

This enables a particularly interesting use case for appreciating assets: a user could borrow a quarter of the value of an asset, and if the market price of the asset doubles, the loan could be paid off directly by allowing a smart contract to liquidate $1/8^{th}$ of the collateral rendering the loan repaid. In the opposite case, a user who borrows against a quarter of the value of an asset and experiences the market price of that asset diminish by a half could look to liquidate half of that asset to render the loan repaid.

3.12 Automated Bridgecoin Repurchasing

To simplify the important arbitrage mechanism described in Section 8.2, a variant of user-triggered asset sales is based on the market price of Bridgecoin. Consider a situation in which intermittently $P^{(BRC)} < 1$. Users with outstanding loans are incentivized to lock in USD-denominated profits by repurchasing Bridgecoin under such market conditions and repaying their loans. Indeed, this behavior is one that would drive the price of Bridgecoin back towards par. Automated Bridgecoin repurchasing by users instructs the system to trigger a collateral sale based on the price of Bridgecoin on the market, as reported by appropriate oracles, making this dynamic automated.

3.13 Direct Collateral Purchase

Here the goal is to enable users to use Bridgecoin to directly purchase collateral into a vault as long as the state of the vault is valid after such a purchase. Example: if $\alpha_{ETH} = 0.5$, $P^{(ETH)} = 100$ and user has 500 Bridgecoin, then the user may pay this Bridgecoin to acquire a vault where $q_v^{(ETH)} = 10$, $b^{(owed)} = 500$. This is economically equivalent to a user paying 1000 BRC for 10 ETH, depositing 10 ETH into a vault and withdrawing 500 BRC from this vault. Obviously, there is a range of valid vaults that can be acquired. For example, for 700 BRC, a user may acquire a vault where $q_v^{(ETH)} = 10$ and $b^{(owed)} = 300$.

3.14 MVP Implementation: Systemwide Borrowing Limit

In order to adequately measure and control the economy implemented by the Sweetbridge liquidity protocol, a limit will be placed on the total amount of Bridgecoin borrowed networkwide. Initially, this value will be set at the discretion of a human operator taking into account the total fiat capital available in the Sweetbridge treasury's Liquidity Pool (Section 7.2), as well as redemption activity (Section 7.1). Following in-depth measurements of the

system utilization and a more empirical understanding of the borrowing and repayment incentives, the limit will be designed as a function related to the treasury, collateral and other measurable system variables, so that the Sweetbridge economic system stability is reliably maintained.

4 Function of Sweetcoin

The Sweetbridge liquidity system will issue and utilize a second currency, Sweetcoin, as a coupon that gives their holders rights to certain discounts and services. From the outset, Sweetcoin will be created in limited supply and released indefinitely via a convergent drip mechanism reviewed in Section 5.2.

4.1 Sweetcoin Activation

Section 3.5 describes the fees Sweetbridge will charge for UOU loans represented as excess repayment liabilities. At user's option, Sweetcoin may be used to offset these fees. To do this, a user may activate Sweetcoin in her vault. The Sweetcoin associated with a vault at any time is denoted s_t . That Sweetcoin may be used as a collateral type, in which case $q_t^{(SWC)}$ denotes the quantity locked as collateral within the vault at the current time t . This usage is distinct from activating the Sweetcoin, denoted $s_t^{(activated)}$. The values s_t , $q_t^{(SWC)}$ and $s_t^{(activated)}$ are well-defined within the context of a specific vault \mathcal{V} . The total Sweetcoin associated with the vault $s_t = s_{t,\mathcal{V}}$, the quantity of Sweetcoin collateralized in the vault $q_t^{(SWC)} = q_{t,\mathcal{V}}^{(SWC)}$, and Sweetcoin activated $s_t^{(activated)} = s_{t,\mathcal{V}}^{(activated)}$ are used when a direct reference to the vault is required. Suppressing the \mathcal{V} , for any such vault, the following must hold

$$s_t \geq s_t^{(activated)} + q_t^{(SWC)}. \quad (4.1)$$

That is to say, activating Sweetcoin in a vault is distinct from using it as collateral alongside other valuable assets. When used as collateral, Sweetcoin acts as any other asset, with its own collateralization coefficient and a real-time price oracle. This equation enforces the fact that a quantity of Sweetcoin may be activated, locked as collateral, or neither, but the same Sweetcoin cannot be used in both capacities simultaneously.

4.2 Offsetting Liabilities with Sweetcoin

The effect of activating Sweetcoin to offset liabilities is defined with respect to the liabilities outlined in section 3.5. The fraction of liabilities that are offset per Sweetcoin activated is determined by the total Sweetcoin locked in the network and the total fiat-denominated value of all locked collateral. The total collateral value in the network is

$$D_t = \sum_{\mathcal{V} \in \mathbb{V}} \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \quad (4.2)$$

$$= \sum_{\mathcal{V} \in \mathbb{V}} d_{t,\mathcal{V}}. \quad (4.3)$$

The total quantity of liabilities in the system can be computed as

$$B_t^{(owed)} = \sum_{\mathcal{V} \in \mathbb{V}} b_{t,\mathcal{V}}^{(owed)}. \quad (4.4)$$

The vault smart contracts are equipped with sell lines ensuring that the individual vaults have the property

$$b_{t,\mathcal{V}}^{(owed)} \leq d_{t,\mathcal{V}} \quad \forall \mathcal{V} \in \mathbb{V} \implies B_t^{(owed)} \leq D_t \quad (4.5)$$

ensuring that the total Bridgecoin owed within the economy remains well supported. The total Sweetcoin activated in the network is given by

$$S_t^{(activated)} = \sum_{\mathcal{V} \in \mathbb{V}} s_{t,\mathcal{V}}^{(activated)} \quad (4.6)$$

where \mathbb{V} is the set of all vaults \mathcal{V} current in the Sweetbridge network. The amount of Sweetcoin that would eliminate all liabilities for a given vault is given by

$$s_t^{(free)} = R_t b_t^{(owed)} \quad (4.7)$$

where R_t is network state dependent *activation rate* defined

$$R_t = \beta \cdot \frac{S_t^{(activated)}}{B_t^{(owed)}} \quad (4.8)$$

with system parameter β set by Sweetbridge to control the sensitivity of the system to fluctuations in the network state. The defining property of β is the fact that reducing β proportionally reduces the amount of Sweetcoin that needs to be activated to achieve an interest-free loan.

4.3 Global Limits on Fee Elimination

As a consequence of its role in activation rate defined in equation (4.8), the parameter β directly defines the share of the excess liabilities networkwide that can be offset through Sweetcoin activation. The global share of liabilities offset Φ_t is defined as

$$\Phi_t = \frac{S_t^{(activated)}}{S_t^{(free)}} \quad (4.9)$$

where the global variable $S_t^{(free)}$ is the total Sweetcoin that would need to be activated to offset all liabilities currently in the network defined

$$S_t^{(free)} = R_t B_t^{(owed)} \quad (4.10)$$

in accordance with equation (4.8), and the observation that the discount is a linear function. Substituting the definition of R_t yields

$$\Phi_t = \frac{S_t^{(activated)}}{S_t^{(free)}} \quad (4.11)$$

$$= \frac{S_t^{(activated)}}{\beta \cdot B_t^{(owed)} \cdot \frac{S_t^{(activated)}}{B_t^{(owed)}}} = \frac{1}{\beta}. \quad (4.12)$$

By design the quantity Φ_t is not a free variable at all but rather a time invariant control of the system tied directly back to the choice of β . Given a preferred value for the global share of the liabilities being offset at any time, Φ , the parameter β must be set as

$$\beta = \frac{1}{\Phi} \quad (4.13)$$

and in the live system, the actual value of $\Phi_t = \frac{S_t^{(activated)}}{S_t^{(free)}}$ can be computed for every block and comparing Φ_t to the value $1/\beta$ will provide a critical measure of health of the network.

4.4 Interest-Free Loans

Consider a vault \mathcal{V} with borrowing power $b_t^{(limit)}$, assuming the user borrows the limit, denote $b_t = b_t^{(owed)} = b_t^{(limit)}$. The user also has sufficient Sweetcoin available to set $s_t^{(activated)} = s_t^{(free)}$, in accordance with equation (4.7). This alters the liabilities accrued as long as the Sweetcoin remains activated; the liabilities update equation found in (3.17) becomes simply

$$b_k^{(owed)} = b_{k-1}^{(owed)} - b_k^{(payments)}. \quad (4.14)$$

It is assumed that t for the purpose of determining the $s_t^{(free)}$ for a specific vault \mathcal{V} is set to the last time the vault state was modified via user interaction. $t = t_0$. In doing so, the discrete time index k with $k = 0$ for time $t = t_0$ may be used for considering the more general case where $s_t^{(activated)} < s_{t_0}^{(free)}$, equivalently expressed as $s_k^{(activated)} < s_0^{(free)}$. Sufficient Sweetcoin has not been activated to completely offset the excess liabilities, but a linear discount is still applied resulting in the liability update equation

$$b_k^{(owed)} = \left(1 + \gamma - \gamma \frac{s_{k-1}^{(activated)}}{s_0^{(free)}} \right) b_{k-1}^{(owed)} - b_k^{(payments)}. \quad (4.15)$$

One can think of this as defining a discounted interest rate

$$\hat{\gamma}_k = \gamma \left(1 - \frac{s_k^{(activated)}}{s_0^{(free)}} \right) \quad (4.16)$$

allowing us to re-express the fees with the context dependent interest rate $\hat{\gamma}_k$

$$b_k^{(owed)} = (1 + \hat{\gamma}_{k-1}) b_{k-1}^{(owed)} - b_k^{(payments)}. \quad (4.17)$$

Both equations (4.15) and (4.17) resolve to equation (3.17) when no Sweetcoin is activated to offset the fees, and to equation (4.14) when Sweetcoin is activated as determined by equation (4.7) computed at time $t = t_0$, representing the last time the vault state was modified. So for any particular repayment plan \vec{b} , the excess liabilities are given by

$$L(b, \vec{b}) = \sum_{k=1}^K \gamma \left(1 - \frac{s_{k-1}^{(activated)}}{s_0^{(free)}} \right) b_{k-1}^{(owed)} \quad (4.18)$$

where $b_k^{(owed)}$ for each k in the sequence is given by equation (4.15).

4.5 Operating Revenue from Loans

As a consequence of the global limits in fee elimination, it is possible to directly derive the revenue generation of the Sweetbridge liquidity protocol in Bridgecoin per payment period from t to $t + \Delta t$ as a function of the total outstanding liabilities $B_t^{(owed)}$. Since every loan has the same base interest rate γ for each period Δt , and the discount function for activating Sweetcoin is linear, it follows that the total new liabilities generated are

$$L_t = \gamma \left(1 - \frac{S_t^{(activated)}}{S_t^{(free)}} \right) B_t^{(owed)} \quad (4.19)$$

and applying equation (4.12), this is simply

$$L_t = \gamma \left(1 - \frac{1}{\beta}\right) B_t^{(owed)}, \quad (4.20)$$

allowing the definition of a new quantity, the global revenue generator coefficient

$$\Gamma = \gamma \left(1 - \frac{1}{\beta}\right) \quad (4.21)$$

which determines the expected Bridgecoin revenue from vaults per Bridgecoin of liabilities outstanding. Knowledge of this relation allows Sweetbridge to reason practically about the required interest rate γ and the fraction of liabilities that can be offset $\frac{1}{\beta}$, necessary to support the operation of the network and stability incentives described in Section 7.3.

This relation justifies the expectation that efficiencies of scale will emerge, allowing the interest γ to be reduced as the scale of the network grows, measured in total concurrent outstanding debt.

4.6 Effect of Activated Sweetcoin on the Borrowing Capacity of Asset Vaults

Sweetbridge will further enhance the users' ability to access Sweetcoin in order to receive discounts. Purchasing Sweetcoin on the open market may be expensive. This expense can potentially be mitigated by increasing a vault's borrowing capacity based on how much Sweetcoin is activated in it. Enabling such a feature would create an additional incentive to purchase and activate Sweetcoin, as it would allow users to access additional funds unlocked from their collateral.

The formal specification of the collateral vaults presented above assumed that α_c is a constant chosen based on the type and market behavior of a collateral asset type. The feature of providing excess collateralization capacity described in this section requires additional analysis with respect to its effect on the rest of the economic specification, because it makes α_c into a function of $s^{(activated)}$. It is clear, however, that in a system in which an upper bound is imposed on α_c , all equations provided in this paper will serve as boundary conditions when $\alpha_c(s^{(activated)})$ is substituted with its upper bound. In this sense, an existence of an upper bound on α can serve as an ad hoc reason for why the system that enables bounded excess collateralization will behave no worse than the system we have described so far.

5 Sweetcoin Availability

5.1 Sweetcoin Supply and Crowdsale

Sweetcoin is a limited supply currency issued by Sweetbridge as a discount coupon toward future services. There will be a total amount of $S^{(total)}$ Sweetcoin created initially. A total public float of SWC $S^{(float)}$ is defined as all SWC allocated by Sweetbridge for public sale.

Sweetbridge will hold a continuous crowdsale in order to provide initial funding for development of the network and the technology, as well as to create a supply of liquid fiat currency to provide free Bridgecoin exchange services to users. Additionally, the crowdsale is designed to incentivize users to generate early deposits of collateral into the Asset Vaults.

In order to purchase Sweetcoin from Sweetbridge: (1) users will acquire Bridgecoin either through a direct at-par purchase from Sweetbridge (see Section 7.1), or by depositing collateral into their Asset Vault, (2) users will deposit Bridgecoin into a Purchase Queue smart contract, (3) Sweetbridge will initiate periodic releases of Sweetcoin that will execute against the orders in the Purchase Queue. Execution priority will be given to the orders at the front of the queue. The price of the Sweetcoin sold against the Purchase Queue will always be beneficial to users in the queue as compared to purchasing Sweetcoin on the open market. This process is intended to support Bridgecoin value early in the system's lifetime at a time when other uses of Bridgecoin are limited.

5.2 Convergent Drip Crowdsale

The crowdsale will proceed over an indefinite period of time in small tranches. Since the total Sweetcoin to be released for public sale is $S^{(float)}$, the remaining quantity for sale can be related to the total sold as

$$S_t^{(remaining)} = S^{(float)} - S_t^{(sold)} \quad (5.1)$$

where $S_t^{(sold)} = \sum_{\tau \in \mathcal{T}_t} S_\tau^{(tranche)}$ and \mathcal{T}_t is the set containing all the distinct times τ earlier than t for which tranches were released.

The size of each release tranche $S^{(tranche)}$ will be determined as a percentage $\rho \ll 1$ of the remaining float

$$S_t^{(tranche)} = \rho S_t^{(remaining)} \quad (5.2)$$

$$= \rho \cdot \left(S^{(float)} - S_t^{(sold)} \right) \quad (5.3)$$

$$= \rho \cdot \left(S^{(float)} - \sum_{\tau \in \mathcal{T}: \tau < t} S_\tau^{(tranche)} \right) \quad (5.4)$$

By defining ρ as a share of the remaining public float, it is ensured that Sweetcoin tranche releases can continue indefinitely without fully depleting $S_t^{(float)}$. At any time t , the remaining float is given by

$$S_t^{(remaining)} = (1 - \rho)^m S^{(float)} \quad (5.5)$$

where $m = |\mathcal{T}_t|$, the cardinality of the set \mathcal{T}_t which also is the number of tranches passed. It is further evident that

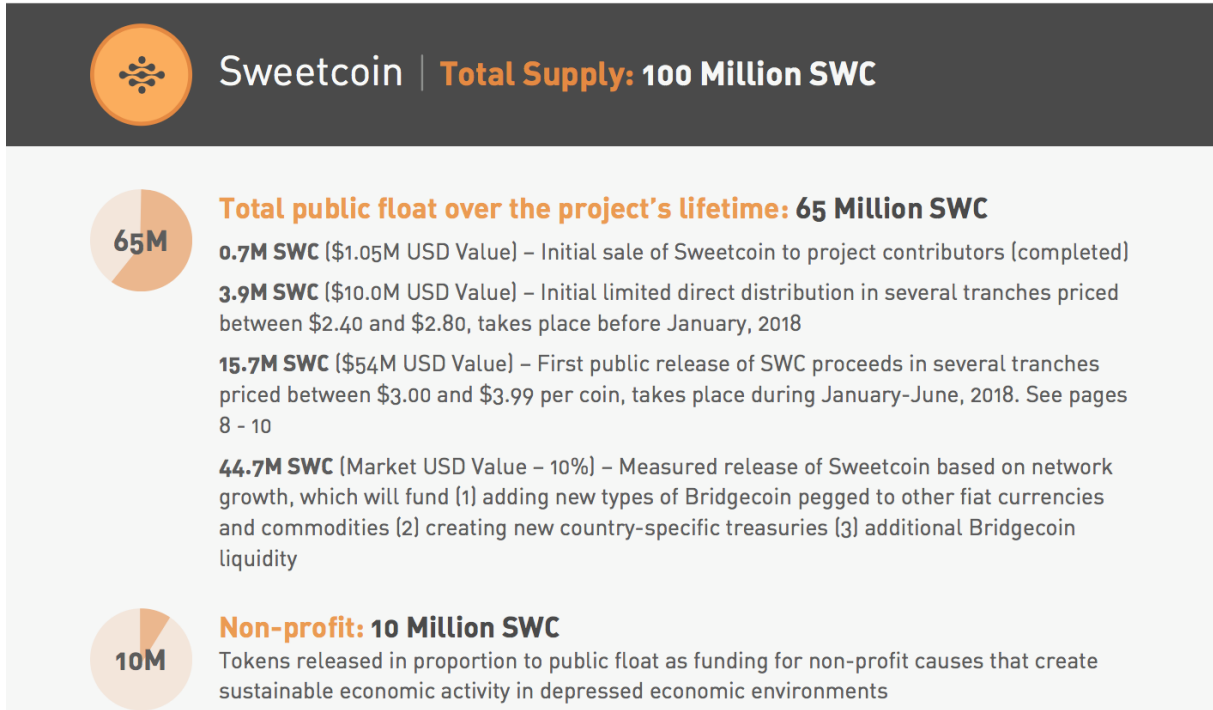


Figure 5.1: An excerpt from the Sweetbridge Crowdsale Announcement document, breaking down the total supply of Sweetcoin

as $t \rightarrow \infty$, and if the tranche releases continue, then $m = |\mathcal{T}_t| \rightarrow \infty$ and

$$\lim_{t \rightarrow \infty} S_t^{(sold)} = \lim_{t \rightarrow \infty} S^{(float)} - S_t^{(remaining)} \quad (5.6)$$

$$= S^{(float)} - \lim_{t \rightarrow \infty} S_t^{(remaining)} \quad (5.7)$$

$$= S^{(float)} - \lim_{m \rightarrow \infty} (1 - \rho)^m S^{(float)} \quad (5.8)$$

$$= S^{(float)}(1 - 0) = S^{(float)} \quad (5.9)$$

thus converging to the intended total of Sweetcoin sold at the exponential rate $(1 - \rho)$. Sweetbridge has selected $\rho = 0.01$; the exponential decay rate is 0.99, which is equivalent to selling half of the remaining public float supply every 69 tranches.

6 Sweetcoin Utility

6.1 Time-Based Utility of Sweetcoin

The utility value of Sweetcoin $U_t^{(SWC)}$ is defined as price type variable derived from the Bridgecoin liabilities offset per one Sweetcoin activated per period of time. $U_t^{(SWC)}$ has the units of dollars per Sweetcoin per period of time and directly accounts for savings realized by a user of the Sweetbridge liquidity protocol borrowing via the vault contract.

Previously (Section 4), the liabilities associated with Bridgecoin loans were analyzed with respect to the amount being borrowed, $b_t^{(owed)}$. In this section, attention is returned to the liabilities equations, but the focus is shifted to the role of Sweetcoin. First, the liabilities per repayment period Δt are restated in terms of the decision to activate a quantity of Sweetcoin s and a quantity being borrowed b ,

$$L(s, b) = \gamma \cdot \left(1 - \frac{s}{b \cdot R_{t+}}\right) b \quad (6.1)$$

where R_{t+} denotes the activation rate defined in equation (4.8) accounting for the additional Sweetcoin being activated

$$R_{t+} = \beta \frac{s + S_t^{(activated)}}{b + B_t^{(owed)}}. \quad (6.2)$$

Substituting and simplifying,

$$L(s, b) = \gamma \cdot \left(b - \frac{s(b + B_t^{(owed)})}{\beta(s + S_t^{(activated)})}\right) \quad (6.3)$$

represents the liabilities incurred in the period Δt for a loan with decision variable b for Bridgecoin borrowed and s for Sweetcoin activated. The financial utility per Sweetcoin activated is the negative of the rate of liability reduction per unit Sweetcoin activated

$$-\frac{\partial L(s, b)}{\partial s} = \gamma \beta \frac{(s + S_t^{(activated)}) \cdot (b + B_t^{(owed)}) - s(b + B_t^{(owed)})}{\beta^2 (s + S_t^{(activated)})^2} \quad (6.4)$$

$$= \gamma \frac{S_t^{(activated)} (b + B_t^{(owed)})}{\beta (s + S_t^{(activated)})^2}. \quad (6.5)$$

Equation (6.5) indicates that in the general case the utility of activating Sweetcoin is a function of the amount of Sweetcoin being activated and the amount of Bridgecoin being borrowed. Let us consider the case in which these quantities are defined as fractions of the network totals, $s = \delta_s S_t^{(activated)}$ and $b = \delta_b B_t^{(owed)}$,

$$-\frac{\partial L(s, b)}{\partial s} = \gamma \frac{B_t^{(owed)}}{\beta S_t^{(activated)}} \cdot \frac{1 + \delta_b}{(1 + \delta_s)^2} \quad (6.6)$$

$$= \frac{\gamma}{R_t} \cdot \frac{1 + \delta_b}{(1 + \delta_s)^2}, \quad (6.7)$$

indicating that the borrower's share of the Sweetcoin activated is a more powerful factor than the share of the

total Bridgecoin borrowed. This may be meaningful for early adopters taking on non-trivial shares of $B_t^{(owed)}$ and $S_t^{(activated)}$. However, in practice $b \ll B_t^{(owed)}$ and $s \ll S_t^{(activated)}$ as soon as the network gains traction. In this case δ_s and δ_b rapidly approach zero. Therefore, define $U_t^{(SWC)}$ in terms of the steady state,

$$U_t^{(SWC)} = - \lim_{\delta_s, \delta_b \rightarrow 0} \frac{\partial L \left(\delta_s S_t^{(activated)}, \delta_b B_t^{(owed)} \right)}{\partial s} \quad (6.8)$$

$$= \frac{\gamma}{R_t} \quad (6.9)$$

indicating the utility of Sweetcoin is directly proportional to the interest rate and inversely proportional to the Sweetcoin activation rate R_t , defined in Section 4.2. The utility of Sweetcoin can be directly represented in terms of the Sweetcoin activated and Bridgecoin debt,

$$U_t^{(SWC)} = \frac{\gamma}{\beta} \cdot \frac{B_t^{(owed)}}{S_t^{(activated)}}. \quad (6.10)$$

6.2 Sweetcoin Fair Value

The utility value of Sweetcoin $U_t^{(SWC)}$ is defined per time interval. In this section we seek an appropriate estimation of Sweetcoin fair value at the present moment. What would it be worth for a user today to own the right of receiving the future discounts enabled by owning a unit of Sweetcoin?

6.2.1 Time-Discounted Value of Money

The first approach we take addresses the time-discounted value of money. The basis for this approach is to calculate the present value one will need to own so as to receive the equivalent amount of services from Sweetbridge in the future. In other words, we answer the question of how much Bridgecoin one needs to have today to pay the Sweetbridge UOU fees equivalent to the amount cancelled by Sweetcoin over the infinite time horizon.

The discount rate used in this calculation is the rate of return on a reference investment. In our case (see Section 7.6), the reference investment is given by the Sweetbridge discount accounts that will have the rate of return equal on average to the Asset Vault interest rate. We will term this value the “discount value” of Sweetcoin, defined as $\bar{U}_t^{(SWC)}$, an expectation of savings over time discounted to the present moment based on the interest rate γ

$$\bar{U}_t^{(SWC)} = \sum_{k=0}^{\infty} U_k^{(SWC)} \frac{1}{(1 + \gamma)^k} \quad (6.11)$$

$$= \sum_{k=0}^{\infty} \frac{\gamma}{\beta} \frac{B_k^{(owed)}}{(1 + \gamma)^k \cdot S_t^{(activated)}} \quad (6.12)$$

where k is the discrete time index; $k = 0$ at time $t = t_0$ and each successive k corresponds to $t = t_0 + k\Delta t$. In the general case, the sum remains open form because the value $U_k^{(SWC)}$ is expected to vary in time. Breaking the $U_t^{(SWC)}$ out using the definition, it is clear that $U_t^{(SWC)}$ can be expected to grow with traction in the network as

defined by increasing $B_t^{(owed)}$ in time, because the $S_t^{(activated)}$ is bounded by the Sweetcoin supply.

To create a practical $\bar{U}_t^{(SWC)}$, a conservative simplifying assumption is applied: $U_k^{(SWC)} = U_t^{(SWC)}$ for all k , effectively treating the current $U_t^{(SWC)}$ as if it were constant. Accounting for this assumption, the infinite sum converges, resolving to

$$\bar{U}_t^{(SWC)} = \frac{\gamma + 1}{\gamma} U_t^{(SWC)} \tag{6.13}$$

$$= \frac{\gamma + 1}{\beta} \frac{B_k^{(owed)}}{S_t^{(activated)}}. \tag{6.14}$$

Observing that the total Sweetcoin available in the network is bounded in accordance with the Convergent Drip Crowdsale, it can be inferred that the value $U_t^{(SWC)}$ and consequently $\bar{U}_t^{(SWC)}$ will rise as the network gains traction. In this case, the total amount $B_t^{(owed)}$ is our notion of traction, and success of the platform will see the total debt $B_t^{(owed)}$, the total dollar value of all collateral D_t and the fiat reserves of Sweetbridge (to be discussed in Section 7.1), all rising in appropriate proportions but without explicit bounds.

6.2.2 Commercial Real Estate Model

An alternative model for calculating fair value is given by comparing the discounts provided by Sweetcoin to rent charged on commercial real estate. A real estate investment is an active investment with two income-generating components: (1) rent value charged to tenants over time, and (2) resale value. Commercial real estate prices are typically calculated as a sum of rent over a finite period of time without applying time discounts and under reasonable rent increase assumptions. The time horizon and the growth assumptions differ by market. In many cases, the time horizon ranges between five and ten years.

Applying this methodology to Sweetcoin, we get the “rent value” of Sweetcoin over time horizon T

$$\hat{U}^{(SWC)} = \sum_{k=0}^T U_k^{(SWC)} \tag{6.15}$$

6.2.3 Comparing the Two Approaches

Without historical data on Sweetbridge utilization, it is hard to create a good estimate on $\hat{U}^{(SWC)}$. However, the two models converge to the same value, assuming a 5% interest rate to calculate $\bar{U}^{(SWC)}$ and taking the time horizon of five years and growth assumptions on $U^{(SWC)}$ of 77% year over year or seven years with growth assumption of 36%. This tells us that the time-discounted valuation model lands us well in the vicinity of the values given by the real estate model under reasonable growth assumptions in an early stage network.

We will be using $\bar{U}^{(SWC)}$ to designate the fair value, with the understanding that as new data emerges, the model

will have to be corrected to reflect actual market behavior. The fair value of Sweetcoin with respect to real utilization of the services provided will be based on the economic analysis of live network data.

Fundamentally, the fair value is a mathematical interpretation of the discounts provided through the activation of Sweetcoin for use of the Sweetbridge platform; the novel contribution of this work is to show that this discount mechanism aligns incentives between Sweetcoin holders and Sweetbridge by showing mathematical alignment between demand on the platform and the implied value of the utility the platform provides.

6.3 Characterizing Sweetcoin Hype

As with all token economies, it can be expected that the market price of Sweetcoin will be influenced by speculation. To a moderate extent, speculation can be good because it indicates a belief by the market that the token will appreciate. Nonetheless, hype often becomes a feedforward process that can ultimately undermine the perception of legitimate utility. Sweetbridge will be leveraging its utility metric $S_t^{(SWC)}$ in order to track meaningful measure of hype within its economy, H_t . The hype metric is defined

$$H_t = \frac{P_t^{(SWC)} - \bar{U}_t^{(SWC)}}{\bar{U}_t^{(SWC)}} \tag{6.16}$$

precisely the percent error of the market price against the fair value. Due to the arbitrage opportunity discussed in the previous section, it is expected that H_t will remain positive on average and recover from negative values rather quickly, limited only by liquidity in the market. An exception to this would be an expectation on part of the users that the utilization of Sweetbridge will diminish over time, which would only be expected in response to material indications of serious problems in the technological implementation or platform governance.

The expected and more interesting case will be to examine H_t empirically to determine whether the hype itself is increasing or decreasing over time. Decreasing hype, converging to $H_t = 1$ on average would represent the market coming to consensus around the belief that the time-weighted expected utility value $\bar{U}_t^{(SWC)}$ represents the fair value of Sweetcoin. A consistent average hype value greater than one would indicate that the market expects Sweetcoin to appreciate at a more or less continuous rate, and it would be interesting to use analytics to observe how long a time interval $t - \tau$ should be expected before $\bar{U}_\tau^{(SWC)} = P_t^{(SWC)}$ where $\tau > t$. Finally and arguably the dangerous case, the hype may be growing in time; should H_t be observed to grow steadily in t , there is a risk that speculation will so completely swamp utility as to undermine the functional value of the token. Should this be observed, Sweetbridge would take action to counteract the destabilizing effects of irrational hype.

6.4 Use of Crowdsale Proceeds

Recognizing the potential for hype around Sweetcoin value, Sweetbridge pledges that operational funds will only draw from the portion of the crowdsale proceeds directly attributed to the network's growth and success.

Preliminarily, the design for the Sweetcoin tranche sales includes discounts against the market price as a function of a number of variables, including but not limited to the size of the Bridgecoin queue commitment and the period for which that commitment has been queued. Further discussion of these concepts is addressed in Section 7.3, and the structural details of the Bridgecoin queue process will be available in a forthcoming publication. At this stage, the primary mathematical requirements are limited to the sale price remaining greater than the fair value of Sweetcoin

$\bar{U}_t^{(SWC)}$, yet being offered at a discount from the market price $P_t^{(SWC)}$. Revenue from every sale of Sweetcoin will be distributed between the development and operations team, Liquidity Pool (Section 7.1), and Stability Pool (Section 7.3). The exact breakdown between these buckets will be determined as part of the ongoing analysis of stability, but certainly will fulfill the requirement that only the revenue attributed to the fair value of Sweetcoin is used to fund the Sweetbridge development and operations, whereas the rest of the revenue will be dedicated to fund mechanisms driving system stability and liquidity.

7 Additional Financial Mechanisms

To this point in the document, most of the financial mechanisms expressed have been subjected to formal analysis. From this point forward, the design and requirements for the other financial forces in the Sweetbridge economic system are presented as they are envisioned by the team. Formal specification of these mechanisms with consistent notation is under active development and will be published incrementally, building toward a complete assertion of the stability criteria for the Sweetbridge economic system.

7.1 Fiat Exchange

Sweetbridge will provide a service termed the Fiat Exchange. It will offer users a limited ability to exchange Bridgecoin for USD and back at par. For clarity, we will use terms *creation* and *redemption* to distinguish Fiat Exchange transactions from the situation where users buy and sell Bridgecoin on open markets at price $P^{(BRC)}$.

Sweetbridge intends to give priority to Sweetcoin holders when using the Fiat Exchange. For example, the redemption mechanism that allows exchanging Bridgecoin for USD directly and at par will be most available and least expensive for users that activate Sweetcoin. This mechanism will contribute a component to the fair value of Sweetcoin in addition to that described in Section 6.2.

7.2 Liquidity Pool

The Liquidity Pool is a fund of USD that is used to offer free redemption services to Sweetcoin holders. The requirement is to specify a mechanism that fairly and responsibly determines how much Bridgecoin can be redeemed by a user who has activated Sweetcoin in her account, accounting for the networkwide effects of this action on stability and solvency.

The ability of a user to purchase Bridgecoin for USD from Sweetbridge at par (creation) is unlimited but may carry a small (infrastructure) fee. Given a period of time appropriate for the real world settlement cycle, all Bridgecoin that a user creates in that period can also be redeemed by the same user for free (infrastructure fees only). USD received but not used for redemption during that period will go into the Liquidity Pool.

The ability to redeem BRC at par for USD is limited by a maximum redemption rate λ , which will be determined precisely in later versions of this specification. The limiting mechanism will account directly for the size of the liquidity pool and the debt outstanding in the Bridgecoin liabilities within the network. In other words, in order to enable redemptions, Sweetbridge maintains a liquid supply of USD in the liquidity pool generated from Bridgecoin that was created but not redeemed in the prior periods; this pool is augmented by revenue generated during tranche sales where the market price $P^{(SWC)}$ exceeds the fair value $\bar{U}^{(SWC)}$, as described in Section 6.4. Due to the importance of retaining the Liquidity Pool, there will be limits imposed on its utilization; those limits are encoded by the parameter λ , the maximum redemption rate.

7.3 Incentives to Use Bridgecoin

This section outlines a set of mechanisms that provide incentives for the direct use of Bridgecoin; that is, transactions that proceed entirely inside Sweetbridge. Sweetbridge will directly use Bridgecoin to create incentives for users for keeping Bridgecoin inside the Sweetbridge ecosystem rather than exchanging it for USD, either through redemption or on the open market. The Bridgecoin being used to fund these incentives comes from the Stability Pool and consists

of Bridgecoin paid into Asset Vault contracts in excess of the original borrowed quantity. Unlike the Liquidity Pool described in Section 7.1, which is a pool of fiat, the Stability Pool is a pool of Bridgecoin. These funds will be reintroduced into the economy through a variety of mechanisms, including subsidies for parties doing business in Bridgecoin and as payment to market makers and other parties whose activity is deemed beneficial to stabilizing the value of Bridgecoin.

7.4 Crowdsales and the Bridgecoin Queue

Sweetbridge will provide a system for public crowdsales of both Sweetcoin and other tokens issued and sold by Sweetbridge's partner projects. This system will use Bridgecoin as a crowdsale currency and will create a fair pipeline where users who deposit Bridgecoin early receive priority to participate in these offerings. Additionally, Sweetbridge will use Asset Vault fees to offer purchase discounts to users who keep Bridgecoin in the crowdsale queue for a long time.

The details of this mechanism will be provided in a separate document.

7.5 Discounts for Bridgecoin Users

In cases where users use Bridgecoin to transact with others, Sweetbridge is in position to offer discounts and rewards for such transactions using the Stability Pool.

Suppose a merchant or service provider chooses to charge 10 BRC for their product or service. Using the Stability Pool, Sweetbridge could support a 10% discount, whereby the user pays 9 BRC and Sweetbridge provides 1 BRC from the Stability Pool, allowing the service provider to receive full payment of 10 BRC.

7.6 Discount Accounts

Sweetbridge will provide incentives for merchants and users who trade in Bridgecoin to keep their value in Bridgecoin rather than convert it to USD. Asset Vault fees will be used to pay users who keep Bridgecoin locked for a given time period. The specific structure of these payments will be defined in subsequent revisions to this document.

7.7 Market Makers, Trust Intermediaries, and Risk Managers

Any healthy economic ecosystem is predicated upon the existence of specialized entities that are professionally qualified and economically able to manage various types of risk within the system, most often earning profit in the process. Sweetbridge is no different. It will present opportunities and offer incentives for entities to provide these types of services to the network. The following are some of the activities that are attractive to third-party professionals in this context:

1. Market Makers are active traders that stabilize the market by purchasing and reselling an asset on the open market while taking short-term financial risks. Anyone can act as a market maker, but some entities will receive a portion of the Sweetbridge Stability Pool if they adhere to some general guidelines, such as consistent presence on the market and a code of ethical conduct.
2. Trust Intermediaries are entities that Sweetbridge allows to use its loan mechanisms on a wholesale basis. In turn, Sweetbridge requires the entity to fund a portion of their loans by buying BRC and placing it in a vault

as security. For example, a large corporation might anchor their supply chain with liquidity, using excess cash to increase the ability of Sweetbridge to loan money to its suppliers in a supply chain finance program. To enable the program, the corporation is required to keep a percentage of its total liability in a vault as Bridgecoin. Trust intermediaries make money through fees that come from the Stability Pool, while retaining a cash equivalent asset on their balance sheet.

3. Risk Managers. In Sweetbridge, a possibility of collateral depreciation is one of the biggest risks to stability. Risk managers are counterparts willing to receive a fee for serving as a buffer against collateral price drops. For example, an entity may want to lock up a large amount of Bridgecoin for a period of time, in the expectation that the system will use this Bridgecoin to repurchase collateral if it starts falling in price and crossing the sell lines. The risk manager will then be responsible for selling or risk-managing these assets, and will be taking risks in the process. In order to incentivize these services, Sweetbridge will pay such entities from the Stability Pool.

These and other similar roles will be evaluated and instituted by Sweetbridge over time.

7.8 The Bridgecoin Community

It is important to note that these types of stability incentives are designed to create a community of Bridgecoin users who can do business with one another. Merely adding these subsidies in a system where service providers immediately cash out Bridgecoin does not provide the intended benefit. These subsidies should be provided to partners who are participating directly in the Bridgecoin economy by *both* spending and accepting Bridgecoin. The goal of these mechanisms is to encourage whole supply networks to adopt Bridgecoin from the first mile to the last mile in a supply chain.

While the Sweetbridge network is in the early stages, incentives to participate directly in the Bridgecoin economy come from subsidies. Later, the second protocol, the Sweetbridge Settlement Bus, will add to this community creation process by introducing tools for efficient settlement of transactions in Bridgecoin and without using fiat currency. The Settlement Bus fees will be lower than credit card and bank settlement fees while providing an additional revenue to fund the Stability Pool. The Sweetbridge Settlement protocol is outlined in [3].

7.9 Reestablishing Bridgecoin Supply

The incentive mechanisms described in this section serve an additional purpose of reestablishing the supply of Bridgecoin that is being slowly diminished by the presence of the liquidity fees described in Section 3.5. The Bridgecoin received by users as a result of discounts or interest payments goes back into the liquid supply and covers the shortfall.

An important control parameter that will be considered in detail in subsequent publications is the question of how quickly this supply shortage should be eliminated. The question to consider here is the negative effect of supply shortage on the overall economics as balanced against the need to maintain the size and liquidity of the Stability Pool of Bridgecoin.

8 Economic System Dynamics

Bridgecoin stability is the property that the expected value of $P_t^{(BRC)}$ is always within a small margin of par; the mathematical characterization of this property is the necessary condition under which the economic system model has a stable equilibrium at $P_t^{(BRC)} = 1$. This stability analysis guides the decisions regarding the power and nature of mechanisms introduced into the Sweetbridge economy. In addition to stability analysis from control theoretic perspective, the Sweetbridge team uses Monte Carlo simulations to validate the stability properties of the system described in the document. Furthermore, both the analysis and the simulations provide insights that determine the logging and reporting requirements for the live Sweetbridge network required to monitor and maintain the stability of the economic network.

8.1 Collateral Volatility

In order to understand this dynamic, it is important to introduce the concept of the “fair value” of Bridgecoin, to be defined in terms of the relationship between the fiat value of the assets locked within vaults and the Bridgecoin repayments required to unlock these assets. As long as the fair value of Bridgecoin is higher than its par value, there exist incentives to repurchase and repay Bridgecoin to unlock collateral.

Under this formulation, the volatility in the price of the assets in the vaults is the primary driver of instability. When collateral prices drop, this creates a disincentive for users to repay their outstanding debts, because the assets unlocked through repayment are worth less. This causes the implied value of Bridgecoin to become less than par, which is a serious malfunction of the Sweetbridge economics.

While there generally exists a symmetrical upward pressure on the Bridgecoin price when collateral price increases, the creation mechanism described in Section 7.1 provides an absolute upper bound on Bridgecoin market price. Accounting for this mechanism, we note that the upward price pressure on Bridgecoin will result in the increase of the size of the Bridgecoin Liquidity Pool and, consequently, create a cushion that will dampen subsequent downward moves.

Recall from Section 3.2 that the vaults were defined as portfolios with incentives for users to diversify their holdings; well-diversified portfolios, particularly among uncorrelated assets, dampen the effect of changes in the value of any particular asset. This means that as Sweetbridge supports more asset types, the user experience will improve, and the system will become more stable.

8.2 Stabilizing Market Forces

Dampening these forces is helpful but not sufficient to drive the economy toward a stable equilibrium at $P_t^{(BRC)} = 1$, and additional stabilizing forces are required. By design, this instability is further suppressed by prominent feedback mechanisms caused by financial incentives:

- **Purchase and Repay:** When $P_t^{(BRC)} < 1$, there is an explicit incentive to buy Bridgecoin on any exchange listing it below par and to use it at an effective discount to repay loans created and spent at a time when $P_t^{(BRC)} \geq 1$. This realizes the gains from the price shift while applying upward pressure on the price of Bridgecoin. The feature that enables users to automatically repurchase BRC, making this dynamic frictionless,

is briefly described in section 3.12.

- **Borrow and Exchange:** When $P_t^{(BRC)} > 1$, there is an explicit incentive to borrow Bridgecoin from a vault contract and sell it on exchanges for an effective gain with respect to the fiat value of the collateral locked, which applies downward pressure as sellers compete to have their sell orders filled.

In addition to these implicit stabilizing market forces, the Fiat Exchange mechanisms described in Section 7.1 further mute the effects of market price volatility by providing an anchor.

8.3 Stabilizing via Creation and Redemption

The Fiat Exchange provides anchoring on the market price by absorbing demand that would otherwise flow into exchanges. Consider the following cases,

- **Absorbing Demand:** When $P_t^{(BRC)} > 1$, buy orders will flow to the creation mechanism, which offers Bridgecoin for sale at par, strictly less than the market price. Pulling demand off the exchanges creates a downward pressure on the market price.
- **Absorbing Supply:** When $P_t^{(BRC)} < 1$, sell orders will flow to the redemption mechanism, which offers USD at a conversion rate of 1 Bridgecoin, strictly greater than the market price. Pulling supply off the exchanges creates an upward pressure on the market price.

This additional stabilizing force is powerful, and while the creation mechanism is unlimited – as Bridgecoin generated by this mechanism is actually created by the smart contract – the redemption mechanism is fundamentally limited by the size of the Sweetbridge Liquidity Pool. The accumulation or depletion of the Liquidity Pool is a second-order effect. Strictly speaking, the stability of the market price of Bridgecoin around par is not a sufficient condition for the stability of the Sweetbridge Treasury. The processes that add to the Liquidity Pool constitute the primary mechanism for ensuring the network remains solvent. Sections 6.3 and 6.4 describe how the release of new tranches of Sweetcoin will add fiat to the Liquidity Pool. Ultimately, this source of fiat not only expands the Sweetbridge Fiat Exchange’s ability to fulfill redemption orders, but more importantly, it supports the confidence that the Sweetbridge Fiat Exchange will be able to meet all future demand, thus materially reducing the risk of a fear-driven run on its funds. The ability to release tranches of Sweetcoin, whether determined by human judgement or an algorithmic stabilizing function, is a powerful mechanism controlling the state of the economy. Since Sweetbridge aims to minimize interest rates to the greatest extent possible, it is the management of this mechanism that bears the most direct analogy to the role of the US Federal Reserve in guiding the United States economy.

8.4 Ongoing Work

The ongoing goal of this analysis is to sufficiently formalize the mathematics of the price-stabilizing mechanism described above and to extend our analysis to determine the steady state pressure on the Sweetbridge treasury as well as to ensure that the Liquidity Pool mechanisms are sufficient to ensure solvency. The term *solvency* is used

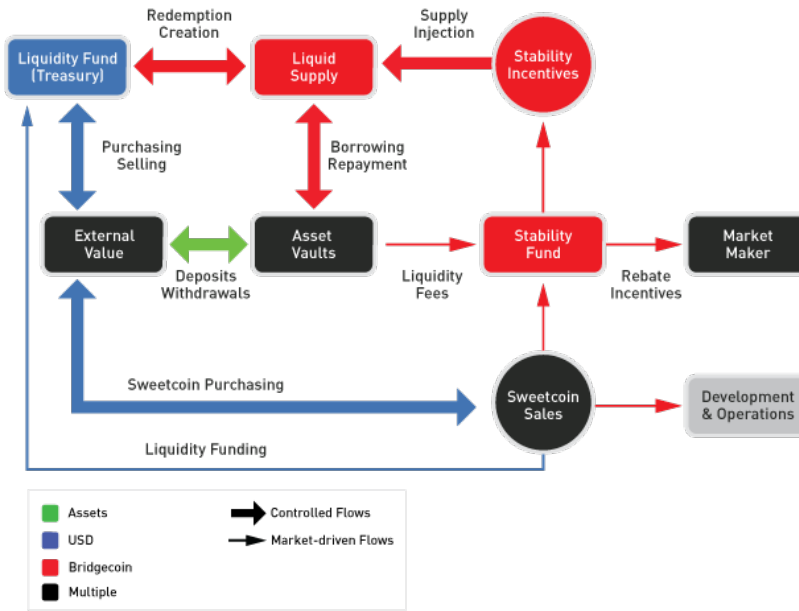


Figure 8.1: Summary of financial flows inside Sweetbridge

to indicate the property that there is zero probability¹ of the Sweetbridge treasury having insufficient USD to meet the demand on the redemption mechanism. This equates to arguing that the Liquidity Pool is sufficiently large as to absorb any temporary imbalances in the utilization of the creation and redemption mechanisms. Such analysis will be based on the theory of stochastic processes, and its results will be further validated by Monte Carlo simulations. Furthermore, both the analysis and the simulations provide an advanced understanding of the Sweetbridge economic system required to determine initial settings of the Sweetbridge-controlled system parameters, as well as to define the logging and analytics requirements for monitoring and maintaining the health of the economy long term.

Figures 8.1 and 8.2 provide a high-level summary of the Sweetbridge financial flows and feedback relationships.

¹Measuring theoretic arguments for stochastic processes will allow us to prove claims of this form should all the necessary mechanism be engineered appropriately.

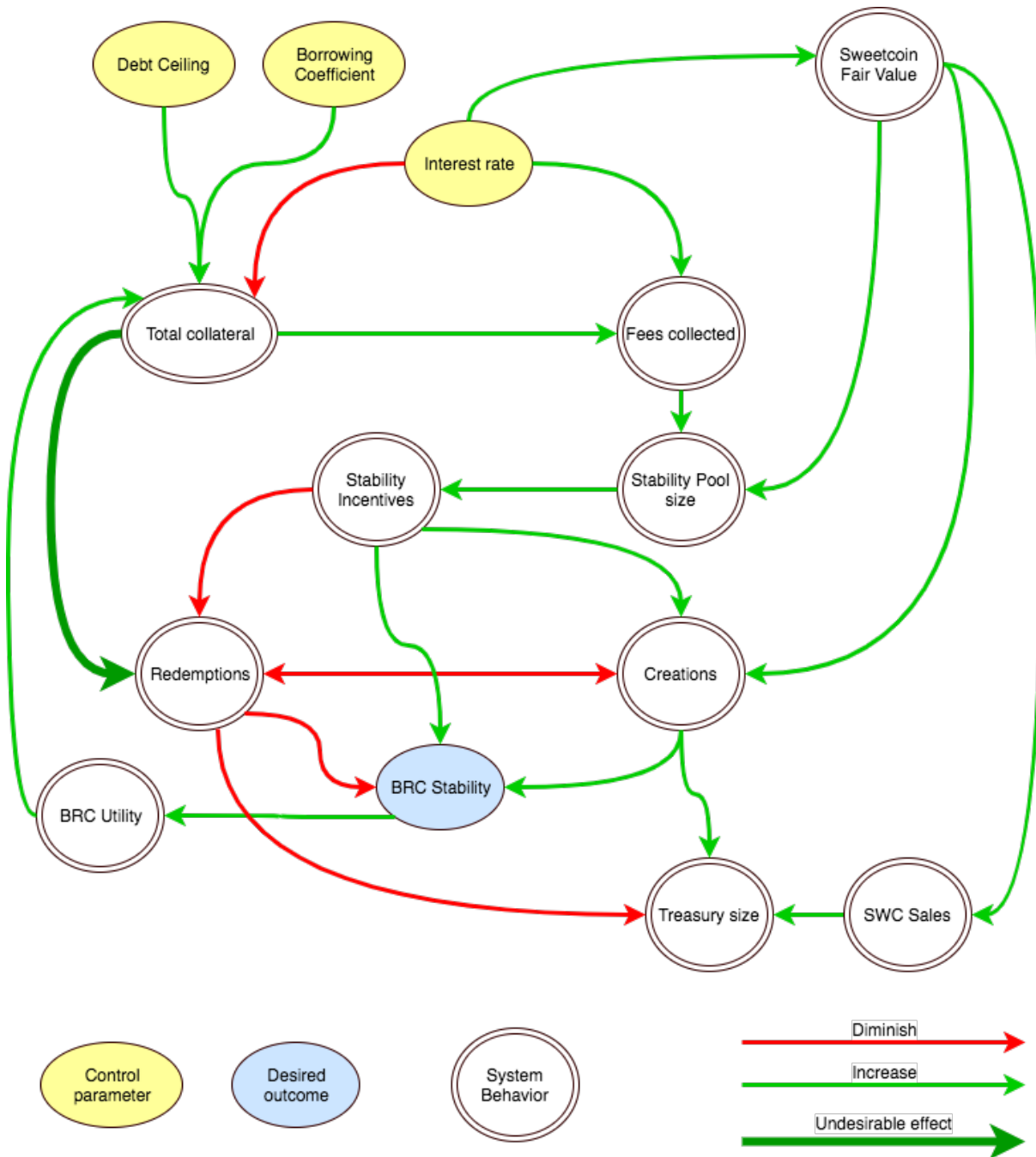


Figure 8.2: Partial feedback loop diagram of the Sweetbridge economics

9 APPENDIX A: Summary of System Control Parameters

In this section, we summarize control parameters within Sweetbridge economics.

9.1 Asset Vault Interest Rate

Interest rate r determines the liquidity fees paid by users for borrowing Bridgecoin from an Asset Vault. Note that practical implementation of r is done by selecting period Δt and computing the discrete time interest rate γ according to equation (3.14). The choice of Δt also has direct impact on user experience as it defines the minimum loan period. Supposing the desired continuously compounding interest rate is $r = 5\%$ per month and the minimum period is to $\Delta t = 1$ day, the continuous time representation of the interest can be avoided and the conversion can be computed:

$$5\frac{\%}{month} = 5\frac{\%}{month} \times \frac{12\text{ months}}{year} \times \frac{year}{365\text{ days}} = 0.164\frac{\%}{day} \quad (9.1)$$

which resolves to $\gamma = 0.164\%$ per day. Due to discrete time sampling the user incurs the full liability of each day, regardless of the time of payment within that day.

9.2 Collateralization Coefficient

Coefficient $\alpha_{c \in \mathcal{C}}$ defined for all collateral types c is the percentage of their USD value that contributes to a vault's borrowing power as defined in equation (3.6). In the first version, only collateral supported will be Ethereum and Sweetcoin with suggested $\alpha_{ETH} = 0.25$ and $\alpha_{SWC} = 0.5$.

9.3 Sell Line

Sell Line coefficient ϵ defines the threshold such that if the value of collateral in the vault drops below the threshold, a sale of collateral will be initiated. It is suggested that $\epsilon = .75$ is a reasonable starting point. Lower would be safer, but keep $\epsilon > \max_{c \in \mathcal{C}} \alpha_c$ where \mathcal{C} is the set of all collateral types currently supported.

9.4 Sell Line Shape Parameter

The sell line shape parameter θ determines the slack in the sell line with respect to the value of the assets in the vault and the riskiness of any particular vaults portfolio, η . Referring to figure 5.1, $\theta = 8$ is a conservative starting point for the initialization of the network.

9.5 Sweetcoin Release Fraction

This parameter has been set to $\rho = .01$. Along with the release schedule, the value ρ controls the speed of Sweetcoin crowdsale distribution.

9.6 Sweetcoin Activation Sensitivity

Value β controls the size of the effect Sweetcoin activation has on reducing liquidity fees. The share of liabilities realized is $\Phi = 1/\beta$, therefore the recommended value is $\beta = 2$ associated with a bound of 50% of systemwide interest fees being offset via Sweetcoin activation.

9.7 Debt Ceiling

The value of the debt ceiling $B^{(ceiling)}$ is denominated in Bridgecoin and places a cap on the total amount of Bridgecoin $B^{(owed)}$ that may be borrowed, always requiring that $B^{(owed)} \leq B^{(limit)}$. This is the networkwide equivalent of the vault-level borrowing limit $b^{(limit)}$. In the early days of the network, this limitation is explicitly and continuously monitored and controlled by a human operator, informed by the total sum of fiat in the Sweetbridge treasury. Following early network edge cases, this value will take algorithmic form, protecting the system from the possibility of insolvency.

9.8 Maximum Redemption Rate

Value λ controls the amount of Bridgecoin that can be redeemed at par for the USD held in the Sweetbridge Liquidity Pool per unit of time and per amount of Sweetcoin activated. This parameter is a critical control mechanism that meters the use of the redemption mechanism to ensure solvency.

10 APPENDIX B: Proof of Theorem 1: The Sell Line Assertion

Any vault \mathcal{V} satisfying the valid new loan condition defined in equation (3.28) will always satisfy the sell line condition in equation (3.29), when the vault-dependent sell line is set using equation (3.27), where shape parameter θ is selected such that $\theta > 0$, and the risk rating of the vault is η_t as defined by equation (3.8).

Proof. The statement of the theorem is equivalent to the claim that

$$\sum_{v \in \mathcal{V}} \alpha_{c_v} \cdot q_t^{(v)} \cdot P_t^{(c_v)} \leq \left(\frac{\theta + 1}{\theta + \eta_t} \right) \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}. \quad (10.1)$$

The following argument demonstrates this claim working backwards from the righthand side expression. The critical point driving this argument is an application of Jensen's inequality applied to the definition of ϵ_t , which holds because function $h(z) = \frac{\theta+1}{\theta+z}$ is concave for $z > 0$ by design; the application of Jensen's inequality yields

$$h \left(\frac{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \alpha_{c_v}^{-1}}{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}} \right) \geq \frac{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} h(\alpha_{c_v}^{-1})}{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}}. \quad (10.2)$$

Putting this inequality in place and simplifying

$$\left(\frac{\theta + 1}{\theta + \eta_t} \right) \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} = h \left(\frac{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \alpha_{c_v}^{-1}}{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}} \right) \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \quad (10.3)$$

$$\geq \frac{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} h(\alpha_{c_v}^{-1})}{\sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}} \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \quad (10.4)$$

$$= \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} h(\alpha_{c_v}^{-1}) \quad (10.5)$$

$$= \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)} \left(\frac{\theta + 1}{\theta + \alpha_{c_v}^{-1}} \right), \quad (10.6)$$

beyond the application of Jensen's inequality, all of the above is substitution of the aforementioned definitions. To proceed now, it is necessary to show that

$$\alpha_{c_v} \leq \left(\frac{\theta + 1}{\theta + \alpha_{c_v}^{-1}} \right) \quad (10.7)$$

which is proven through contradiction. Suppose the oppose claim:

$$\alpha_{c_v} > \left(\frac{\theta + 1}{\theta + \alpha_{c_v}^{-1}} \right) \quad (10.8)$$

then it follows through simple algebra that

$$(\alpha_{c_v} - 1) \cdot \theta > 0 \quad (10.9)$$

which contradicts the definitions as all $\alpha_{c_v} < 1$ and $\theta > 0$. Now inequality (10.7) is applied to (10.6) to yield

$$\sum_{v \in \mathcal{V}} \alpha_{c_v} \cdot q_t^{(v)} \cdot P_t^{(c_v)} \leq \left(\frac{\theta + 1}{\theta + \eta_t} \right) \sum_{v \in \mathcal{V}} q_t^{(v)} \cdot P_t^{(c_v)}. \quad (10.10)$$

□

Bibliography

- [1] Vitalik Buterin. A next generation smart contract and decentralized application platform. Ethereum White Paper, 2015. [LINK](#).
- [2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Bitcoin White Paper, 2008. [LINK](#).
- [3] J. Scott Nelson, David Henderson, Glenn Jones, Micha Roon, Michael Zargham, Aleksandr Bulkin, Jake Brukhman, and Kenny Rowe. A blockchain-based protocol stack for global commerce and supply chains. Sweetbridge White Paper, September 2017. [LINK](#).