ButtonStable — An Idea for a Decentralized Collateral "Stablecoin"*

Manny Rincon-Cruz
manny@thinking.farm

v.3 March 11, 2021

Abstract

ButtonWood is a family of simple DeFi smart contracts whose primitive functions can be combined into more complex financial instruments. This short paper lays out one idea for how to combine two of these protocols into a "stablecoin" which we call "buttonStable." The design borrows from financial history, and issues "derivative money notes" against a pool of "safe asset" debts, which are themselves the product of a volatility-tranching protocol called buttonTranche. The advantages of our design is that both the yield and leverage assets produced by buttonTranche do not need vaults or liquidation markets, so they can be freely traded. This makes it easy to price and manage risk. The process for minting a derivative money note is likewise simple, so buttonStable is both modular and robust.

1 Introduction

ButtonWood is a family of simple DeFi smart contracts whose primitive functions can be combined into more complex financial instruments. This short paper lays out one idea for how to combine two of these protocols into a "stablecoin" which we call "buttonStable."

The central idea, borrowing from the financial history of free banking, is that a pool of "safe asset" debt can support a stock of derivative "money notes." This contrasts with all other extant DeFi projects, which treat raw debt tokens themselves as the "money notes"—MakerDAO with Dai, Yield Protocol with yfUSD, mStable with mUSD, and so on.

---

*The idea for combining the two ButtonWood protocols discussed here was first proposed by Mark Toda in a text message dated February 2021.

Contrary to popular belief, “safe assets” are not necessarily risk free. The rate of return on these assets has historically been quite high. For example, the rate of return on European sovereign debt has declined for the last 800 years and is today often negative. However, early in the development of the European financial system, sovereign debt was still heavily collateralized with valuable commodities such as land, jewels, and metals, while nominal interest rates ranged from 12% to 15%, and as high as 25%. [1]

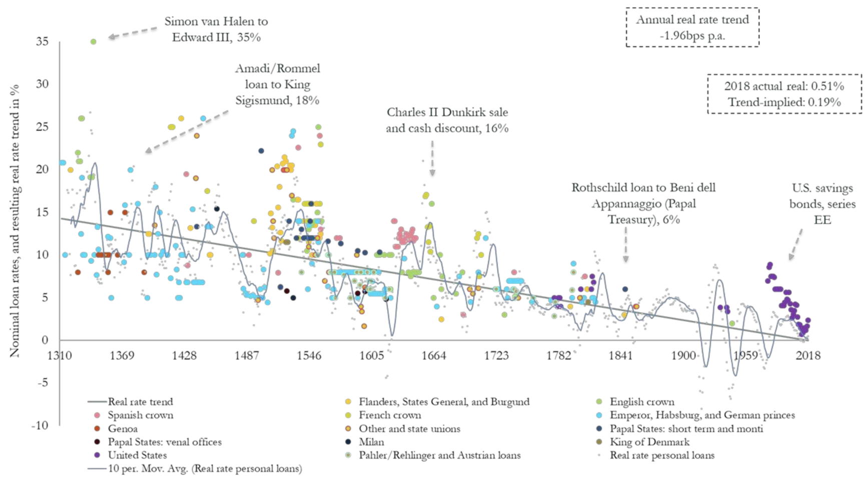


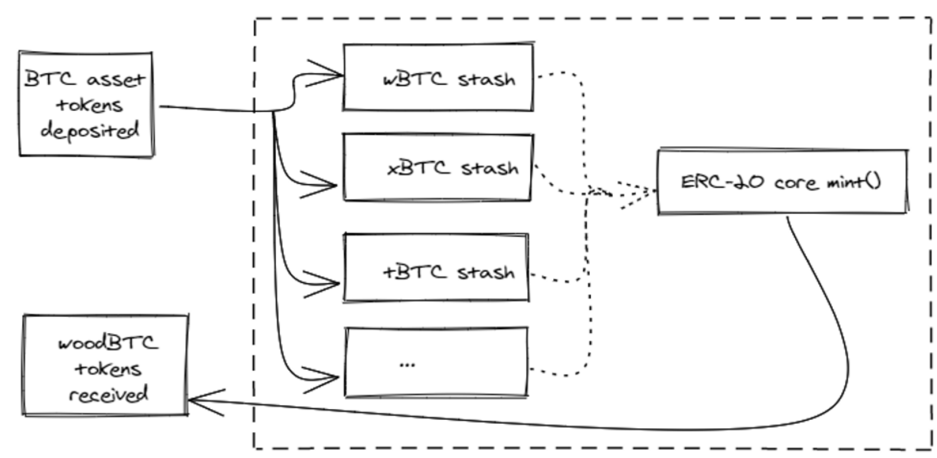
Figure 1: Paul Schmelzing, “Eight centuries of global real interest rates, R-G, and the ‘suprasecular’ decline, 1311–2018.”

DeFi is financial history in a petri dish. As a historian, I think crypto today is closer to finance in 15th century Venice than 20th century England. Thus, history suggests that the next step towards viable “money notes” is to take the volatile commodity assets that are the bulk of crypto markets today and create a system of “safe asset” debt. This debt can then be used to underwrite a derivative “stablecoin”. To this end, buttonStable uses trancheTokens as “safe asset” collateral in a woodToken contract, which mints the derivative “money note”.

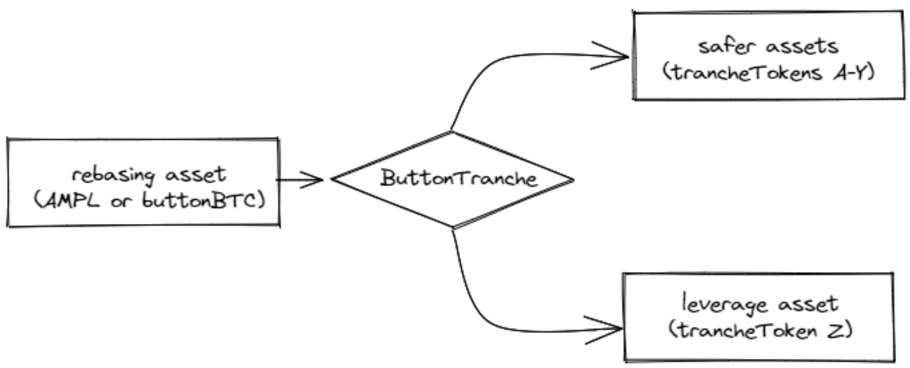
2 woodTokens and trancheTokens

The main function of woodTokens in the ButtonWood ecosystem is to aggregate liquidity for assets which have multiple representations on Ethereum—though as we will show through the buttonStable implementation, they have other uses. WoodTokens are composed of a core contract, which issues an ERC-20 when collateral is deposited, and a series of “stashes,” which store collateral. For example, woodBTC could take as collateral wBTC, renBTC, tBTC, and others. The resulting woodBTC token represents a “blend” of the underlying collateral, helping to diversity custodial risk.

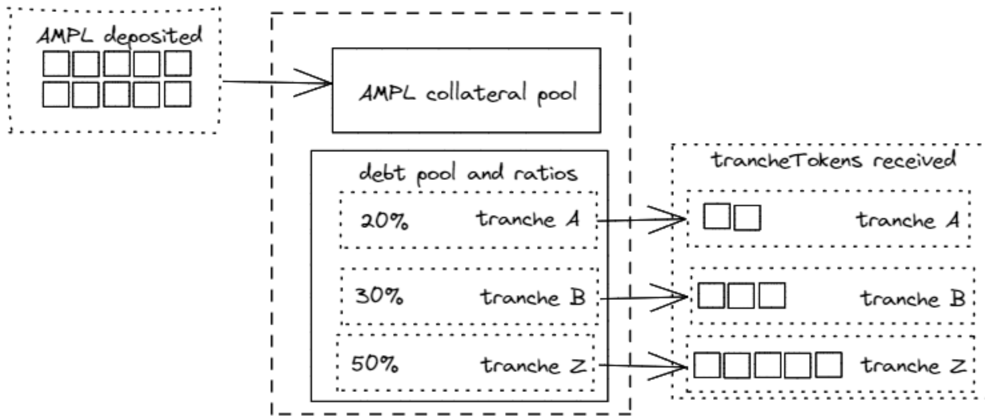
[1]Paul Schmelzing writes, ”This downward trend has persisted across monetary regimes, is visible across various asset classes, and long preceded the emergence of modern central banks.”



Each woodToken contract has a list of acceptable collateral, and this list is governed by the contract owner. By default, the owner is the ButtonWood protocol token, BUTTON. This is not necessary, as we will discuss later alongside the governance of buttonStable.



ButtonStable could be a specific woodToken contract whose collateral is a variety of trancheToken "safe assets" collateralized by AMPL, BTC, ETH, and UST. The button-Tranche protocol works by stratifying the risk of rebasing assets into "safer" tranches and "leverage" tranches. These tranches are called trTokens.



Safer tranches have a more senior claim on the underlying collateral. This represents their exposure to the volatility of the underlying collateral's value. For example, in the above diagram the ratio and priority for trAMPL tranches is A:B:Z and 20:30:50. The number of collateral tokens in an buttonTranche contract increases and decreases as the collateral changes in value and rebases. Once redemption occurs, the collateral tokens are distributed to cancel out the trAMPL "debt" in each tranche. This means the AMPL market cap would need to drop by 80% for the value of trAMPL tranche A to be impaired.

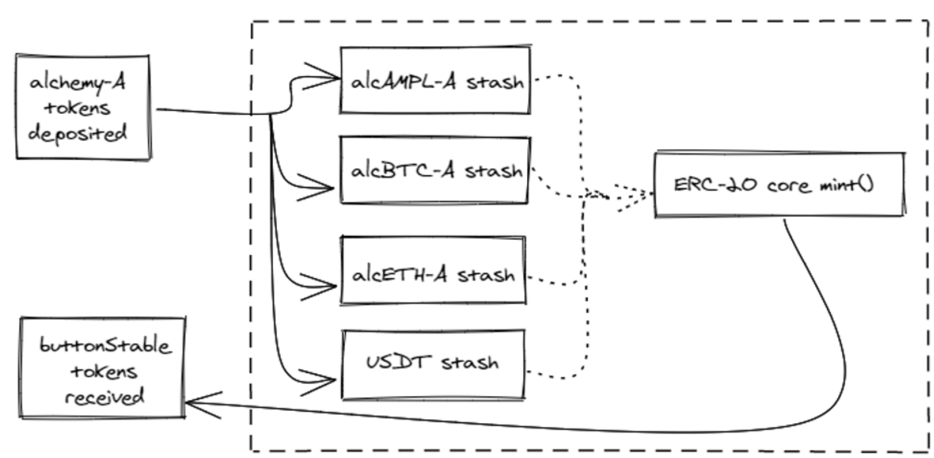
The Z-tranche, unlike other tranches, serves as an "equity" tranche. It retains upside exposure to appreciation in the AMPL collateral. Owning it is the same as owning a leveraged position on AMPL. When redemption occurs, and the claims of senior tranches have been met, tranche Z is entitled to the remainder of the collateral, no matter how much. So, using the above example, if the value of AMPL has increased by 50%, the number of AMPL tokens received by tranche Z would increase by 100%.

Various "flavors" of buttonTranche contracts can be created depending on 1) redemption conditions, i.e. does the contract have a maturity or can one freely redeem trTokens; 2) the existence of a fixed tranche minting ratio; and 3) the existence of a fixed tranche redemption ratio. Three of the eight possible combinations of these properties are discussed further in the ButtonTranche white paper, v.3. We believe fixed burn/mint ratios, with either free redemption or redemption at maturity provide the most useful variants. Free mint/burn ratios create complex game dynamics between users, which for the purposes of this papers makes these instruments unusable as safe asset collateral for a derivative money note.

	Fixed redemption (maturity)	Free redemption
Fixed mint/burn ratios	Maturing	Perpetual
Free mint/burn ratios	Game 1	Game 2

3 buttonStable

The ButtonWood ecosystem currently supports trAMPL and trBTC, since AMPL is naturally rebasing and we have created a rebasing wrapper for Bitcoin called buttonToken. This support can be expanded to ETH and other major assets. Doing so would allow us to produce enough tranche A trTokens for a diverse buttonStable asset pool. These assets, together with USDT, would diversity risk as much as is possible in a crypto portfolio, given the historically high correlations between digital assets. ButtonStable's core woodToken ERC-20 contract would then mint tokens with a blended risk profile that should be able to withstand market drops of up to 80%. Even a cataclysmic 90% drop in the market would only impair buttonStable's redemption value by 40%, assuming that USDT comprises 20% of the collateral pool.



4 Governance considerations

As mentioned earlier, the woodToken contract has two levels of governance. The first is the list of approved collateral stashes. The second is a list of whitelist of wallets approved to mint or redeem woodTokens. We outline four combinations below—though technically there are eight combinations, if the mint/redeem whitelists are not identical.

	Decentralized stash list	Centralized stash list
Decentralized mint/redeem	Anyone can mint or redeem, and the stash list controlled by BUTTON governance	Anyone can mint or redeem, but the stash list controlled by major stakeholder
Centralized mint/redeem	Wallet white list controlled by major stakeholder. Stash list controlled by BUTTON governance.	Wallet white list and stash list controlled by major stakeholder.

As can be seen, a major stakeholder could control the wallet whitelist and stash list, perhaps to jumpstart liquidity and trust in the ecosystem. It could then, over time, decentralize the stash list, minting, and redemption as appropriate.

5 Market considerations—arbitrage and impairment

There are a few market considerations worth discussing, which are impacted by the state of governance as well as other minor design features.

The most important is the nature of arbitrage. Assuming a totally decentralized system, market participants will arbitrage minor price differences between various stashes. For example, if trAMPL-A is trading at $1.10 and trBTC-A is trading at $1.00, then it makes sense to buy buttonStable on the market to redeem it for trAMPL-A, which can then be liquidated. However, this might raise the price of buttonStable to the point that this arbitrage is unprofitable. Perhaps buttonStable generally trades at the price of its highest possible redemption collateral.

A second way of arbitraging this price disparity is to buy trBTC-A on the open market, to then mint buttonStable, which is then redeemed for trAMPL-A. This activity would bring the prices of trBTC-A and trAMPL-A back in line. Of course, the size of these effects depends entirely on the relative share of trTokens in the buttonStable stashes versus the open market.

This type of arbitrage might be made less efficient if the token has a built-in minting or redemption “tax.” Even a few basis points shaved off at minting or redemption would allow the underlying collateral prices to fluctuate more freely.

The second market dynamic to consider is when a specific collateral suffers major impairment or dislocation. In other words, if the collateral underpinning one of the trToken-A stashes falls by over 80%, then the value of those tokens might be severely impaired. Two factors influence how this affects buttonStable. The first factor is the size of the stash rela-

	Summary of Considerations	
Arbitrage: divergence in market price of underlying collateral	Share of collateral asset locked in buttonStable versus general open market. Lower priced collateral and buttonStable token might trade at higher price, as can be used to redeem for more expensive collateral.	Minting or redemption fees, by default zero, add friction so that only price differences above a threshold trigger arbitrage activity.
Impairment: Some collateral permanently lower in value	Probability of a "run on the bank" closely tied to size of impaired stash relative to rest of collateral pool. Even if 20% of collateral severely impaired, run on bank risk for participants is quite low.	Share of collateral asset locked in buttonStable versus general open market. The greater the share, the greater the "monetary" premium accrued to the asset, which mitigates impairment.

tive to the overall collateral pool. If the stash is a very large portion of the collateral pool, say at the extreme 90%, then there is an incentive for buttonStable holders to redeem their tokens for the unimpaired collateral stashes. This is a classic bank run. On the other hand, if the stash is relatively small in size, say 10% of the total, there is very little incentive to run and redeem for collateral. This is because 90% of buttonStable tokens would have to be redeemed in order to run the risk of getting "stuck" with impaired collateral.

A second factor is the size of the impaired stash relative to the size of the trToken's general market. The fact that an impaired trToken could be used to mint buttonStable and redeem it for more valuable collateral (as described above), means that trTokens included in the buttonStable collateral pool will trade at a premium. The greater the share of trTokens that are locked within buttonStable, the more likely that they will trade at a higher price. Ironically, this higher price might reduce the extent to which the tokens are "impaired" in the first place.

Historically, this dynamic is not rare at all. In fact, it was a sought after property called "monetization." Metals and sovereign debt that became collateral for more stable money notes traded at higher prices than they had before (and for bonds this was reflected in lower interest rates), which also mitigated the extent to which their value was impaired during crises.

6 Big picture—comparison with other protocols

We believe that the buttonStable design is more robust than other ”money note” or ”debt” protocols given its simplicity—it does away with liquidation markets and vaults at both the woodToken and buttonTranche levels. This is the application of what Nassim Nicholas Taleb calls “via negativa,” or improvement by subtraction. The “money note” tokens themselves are basically a simple claim on blended collateral, compared to the complex mechanisms required for algorithmic stablecoins like Fei, FRAX, ESD, and Basis Cash, with their programmatic bond market-making or dynamic AMM rewards and penalties.

Compared to other debt or stablecoin protocols, this simplicity gives buttonStable five advantages: 1) it clearly separates “yield assets” from “money notes”; 2) it cleanly stratifies asset volatility, allowing for the clear pricing of risk; 3) it tokenizes the supply and demand for leverage and yield, allowing the market to efficiently price both; 4) it is fully composable and modular; 5) it has zero rent-extraction, with no unnecessary fees or tokens, i.e. we are building the protocol which we would have wanted others to build for us to use.

6.1 Separating yield and money

One major improvement over protocols like MakerDAO, mStable, Yield, and others, is the separation between yield and money. These protocols produce debt, which by definition carries a yield in its interest rate. However, these protocols also attempt to force these debts to behave like “money.” This contravenes the historical process whereby commodity assets, through collateralization, produce debts, followed by “safe” financial assets, and lastly, derivative money notes. As we have written elsewhere, money or “stablecoins” are by definition assets whose probability of gain and probability of loss are both zero—that is, p(gain)=p(loss)=0.

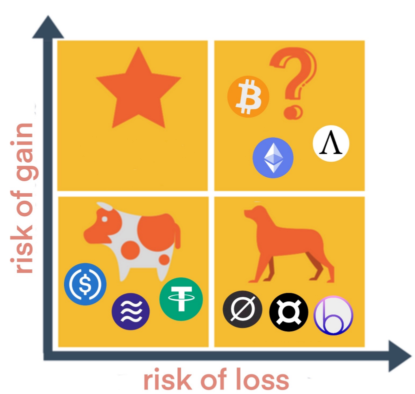


Figure 2: Manny Rincon-Cruz, “A Stablecoin by Any Other Name: Value, Risk, Loss, and Gain.”

Protocols like MakerDao and Yield try to square the monetary circle by requiring their debt tokens—DAI and yfUSD—to maintain a minimum collateralization ratio. This, of course, is the reason for having vaults and liquidation markets in the first place. Admittedly, the use of a minimum collateralization ratio allows the debt tokens to be maximally fungible. But we cannot help but be wary of a system that produces debt—a naturally yielding asset—and which attempts to pass it off as non-yielding money.

6.2 Cleanly separating volatility, cleanly pricing risk

This brings us to our second point, namely that buttonTranche cleanly separates volatility of its underlying collateral, and thus allows for clean pricing. In contrast, DAI and yfUSD introduce exotic risk. By relying on vaults and liquidations, the risk of under-collateralization depends on the volatility of the underlying assets, as well as the efficacy of liquidators, and the solvency of the protocols' borrowers. This complexity makes risk hard to assess, and thus hard to price.

6.3 Freely trading demand and supply for leverage and yield

Third, buttonTranche allows both borrowers and lenders to tokenize and trade their positions. Minting trTokens but only selling A-tranches is equivalent to borrowing—the discount offered is the effective interest rate. The Z-tranche of an trToken is functionally equivalent to the vault in the MakerDAO or Yield systems—a borrower retains their upside exposure to an asset. The A-tranches are functionally equivalent to risk-transparent versions of DAI or yfUSD.

By encapsulating both leverage and yield in freely tradeable tokens, buttonTranche lets market demand and supply determine the price for each. Perhaps during a bull market, borrowers mint and sell A-tranches at a discount. Perhaps during a bear market, it is the Z-tranches that trade at a discount, while senior tranches trade near par.

6.4 Composability

Fourth, buttonStable shows how simple protocols are robust individually and collectively. They can be easily combine and recombined, given their modularity and minimal need for governance. The contracts are easy to audit and hard to break. The primary technical risk lies in the rebasing mechanism of the collateral behind trTokens. However, we believe that this risk is much less than the risk presented by building a custom oracle for trToken redemptions, as other projects like Yield and UMA have done for their tokens.

At the same time, any other project is free to use woodTokens, buttonTranche, or any other Buttonwood protocol. This is in contrast to systems like Yield, UMA, or MakerDAO, where a user must have an open account, and which try to keep all their components and instruments "in-house".

More importantly, modularity means that different collateral stashes can be swapped in and out of buttonStable without the need to re-engineer the system. We are also designing an even more modular version of the woodToken contract, where the specific minting function can be swapped out.

6.5 Zero rent-extraction

Lastly, the simplicity of most Buttonwood protocols, including those here in buttonStable, implies no rent-extraction through fees or though a "utility token." We believe that the space will eventually converge on the most robust, lowest-rent version of any financial instrument. Projects that aim to extract rents through "utility tokens" like Bancor or UMA, we think, will suffer from slower adoption. The success of Uniswap, which is similar to Bancor but without the proprietary utility token, supports this hypothesis.

In short, we are building buttonStable and other protocols the way we wish people would build protocols for our own use. We think we would eat our own cooking! The BUTTON governance token accrues no fees and is not needed to use any ButtonWood protocol. Its sole purpose is to handle specific functional "exceptions"—such as triggering redemption on a bond or altering the buttonStable whitelist.

7 Conclusion

Fiat-collateralized stablecoins borrow the fruits of traditional finance's long history—namely credit-based money notes. History suggests that the path forward for DeFi is to march down a similar convergent evolutionary path—from the creation of markets for volatile commodities, to collateralized lending, to "safe asset" creation, to derivative money notes.

But evolutionary convergence does not mean functional equivalence—a bat is not a bird, after all. DeFi's strength is its ability to create transparent, self-contained, composable financial primitives. The ButtonWood protocol ecosystem aims to embody these principles.

Thus, buttonStable converges on the model of a derivative money note, but in a way that proves more transparent, self-contained (in terms of risk), and composable. In the long-term this makes the buttonStable system robust and easier to modify. At the same time, financial risk remains contained to individual components. This reproduces a small amount of volatility but prevents risk from building up and becoming systemic. This trade-off becomes clearer in comparison to other DeFi projects, where complexity, and interdependence, is abundant and perhaps a badge of pride for talented developers and ex-financiers. While those systems more closely mimic the functions of traditional finance, they also replicate the need for an agile, hyper-engaged team and the tendency to obfuscate systemic risk.

References

[1] **Kuo, Iles, Rincon-Cruz, et al.** (2019). “Ampleforth–A New Synthetic Commodity” *paper available at: http://www.ampleforth.org*.

[2] **The Maker Team.** (2020). “The Maker Protocol: MakerDAO’s Multi-Collateral Dai (MCD) System” *paper available at: https://makerdao.com/en/whitepaper/*.

[3] **Robinson, Niemerg.** (2020). “The Yield Protocol: On-Chain Lending With Interest Rate Discovery” *paper available at: https://research.paradigm.xyz/Yield.pdf*.

[4] **Schmelzing, Paul.** (2020). “Eight centuries of global real interest rates, R-G, and the ‘suprasecular’ decline, 1311–2018.” Staff Working Paper No. 845. Bank of England, January 2020.

[5] **Santory, Joey.** (2020). “Fei Protocol: A Decentralized, Fair, Liquid, and Scalable Stablecoin Platform.” *paper available at: https://s3-us-west-1.amazonaws.com/assets.fei.money/docs/whitepaper.pdf*.

[6] **Kazemian, Sam.** (2021). “Frax v2: Algorithmic Market Operations.” *paper available at: https://samkazemian.medium.com/frax-v2- algorithmic-market-operations-b84521ed7133*.

[7] **UMA Project.** (2021). “UMA Protocol Docs.” *documents available at: https://docs.umaproject.org/*.

[8] **Eliosoff, Jacob.** (2020). “What’s the simplest possible decentralized stablecoin?” *paper available at: https://jacob-eliosoff.medium.com/whats-the-simplest- possible-decentralized-stablecoin-4a25262cf5e8*.

[9] **Rincon-Cruz, Manny.** (2021). “A Stablecoin by Any Other Name: Value, Risk, Loss, and Gain.” *essay available at: http://thinking.farm/essays/2021-01-14-stablecoin-by-any-other-name/*.

[10] **Empty Set Squad.** (2020). “Døllar.” *paper available at: https://github.com/emptysetsquad/dollar/raw/master/d%C3%B8llar.pdf*.

[11] **Basis Cash Team.** (2021). “Basis Cash Docs.” *documents available at: https://docs.basis.cash/*.