BitBay

DECEN-TRALIZED PEG

Forcing any price through freezing and unfreezing of coins in cryptocurrency networks

David Zimbeck
www.bitbay.market

1.27.2017

Abstract

This paper proposes a profoundly simple, yet unstoppable method for controlling the price of any cryptocurrency. It is a fair technique that serves to eliminate volatility concerns in cryptocurrency once and for all. ***This dynamic peg will allow prices to increase and decrease, but in a controlled manner that is fully decentralized and not reliant on third parties, governments, backers, trading tricks or trusted entities.*** This method will show that a currency's price can be determined by any algorithm, or even the users themselves - who all maintain a vested interest. Users can always make the price match demand, fall above or below it based on their collective wishes. This method seeks to eradicate the manipulation, speculation, and consumerism that drives price within cryptocurrency markets. It allows even the smallest local economy to thrive and prosper in any market condition, no matter how hostile. It is the perfect system for failing governments, low volume penny stocks and cryptocurrencies. We will propose a good foundation for healthy cryptographic currencies, taking into consideration their speculative and risky nature. We seek to bring Bitcoin[1] and other currencies to their rightful position as permanent members of society, harboring freedom of choice for individuals to participate in any currency without the fear of financial loss.

The method is simple: freezing and unfreezing coins. A global supply rate is set and may decrease or increase by a certain amount daily, or in intervals. The daily positive or negative percent in change can be determined through voting. A new rule is added that enforces users to freeze their own funds based on the times sent and received, relative to the supply rate. This rule is enforced each time funds are spent. If funds are to be frozen, they are sent back to the origin address as change. In addition, bylaws can be made for sending frozen coins at a slower pace (such as 1 month time locks) to allow for two asset classes within the same network. The end effect will be fair to all users of any currency that employs this method. It will not favor an elite few. Both rich and poor are affected evenly through the effect of freezing and unfreezing funds (unlike centralized political systems which only favor those who control them). Various strategies and effects of this idea will be revealed after implementation, and we expect this to only be the beginning of a long awaited financial revolution. Furthermore, multiple techniques beyond the primary theory will be discussed to inspire more work within the subject. This technology is set to debut as open source code in a coin called "BitBay"[2] as a proof of concept developed in Python by David Zimbeck.

Table of Contents

Introduction & history

Supply and demand is the fundamental rule when it comes to the value of any commodity. In the case of currencies, centuries of experience has proved this to be true. Throughout history, currency prices have been controlled with a vast array of methodology. Backing with gold, inflation, taxation, perception, network effect, political methods, price pegs, and even obscure secondary aspects - such as military force, price fixing, and market manipulation - all serve as examples.

> *They all possess one thing in common: a centralized method of value derivation.*

The net result is a delegation of power to the hands of a select few who subsequently use that power to oppress the less-fortunate. In the most extreme cases, this power has transformed money into a tool of economic slavery, as opposed to a basic tool of free trade. Until now, the majority of price control methods have been implied through compounding effects of economic force, including human psychology, convenience, and the force of law (a threat of violence or incarceration). Various political and economic methods of price stabilization have been tried with some success. Typically, the origin of price lacks transparency and is relative (open to subjective debate). Only those controlling the currency truly know the reason for fluctuations. Entire books have been written on any one of these subjects. However, this is not a paper in political and economic philosophy. It is one of utility and reality. Simply put, one rule always rings completely true. Supply and demand is the most simple method of price control. In the case of gold, something which was formerly abundant is now highly scarce. This scarcity directly contributes to its high value. For some, food can appear to be very trivial to acquire. However in times of famine, it is more valuable than anything in the world. You can't eat a Bitcoin, you cannot drink gold. It is the tools of trade that allow us to compensate others, who in turn, help us acquire the things we cherish most: food, shelter, and freedom. We take for granted the tool of money which is secondary to primal needs. The foundation of the value of money is therefore vital to its utility. ***Money based on trust dissolves and dies the moment that trust is eroded.*** When that system of trade fails, famine and death can quickly result. To prevent this, a trustless, decentralized, safe, and unbreakable system must be found. We believe the ideas presented here will be an important piece in the development of such a system. There have been examples of dynamic pegs in the past. This type of peg can be called a "crawling peg". Until now, this

archaic system has been based on both market conditions and a country buying its own currency to politically control the price. It is a centralized method that relies on a select few, rendering it somewhat unsustainable. Additionally, when managing physical fiat (not electronic cash), it is not possible to control the supply in a transparent way - which is being proposed here.

Bitcoin is a newer technology that has obtained its value through network effect, low supply, low inflation, and its appeal to multiple unique niche target markets. Many currencies around the world inflate faster than Bitcoin, driving the demand for citizens of countries (such as Venezuela) to purchase it in favor of their own fiat currency. People in China buy Bitcoin to avoid capital controls and move money out of the country. People in Cyprus bought Bitcoin because banks were doing unethical runs on their money. Some even buy Bitcoin to subvert and break laws due to it's untraceable nature. However, what makes Bitcoin more valuable than other cryptographic currencies? Are there not, after all, many currencies that exceed Bitcoin in technology and capacity? The network effect, branding, popularity, demand and distribution are all significant factors. To many, it seems like a dream currency that only goes up. Unfortunately, what goes up, must come down. It has repeatedly happened in the past. At one point Bitcoin reached 30 dollars and then crashed to almost a few dollars. Then later on, Bitcoin exchanges performed theft, many coins were held by a small number of people, fear and uncertainty swept the market and quickly cut the price in quarters.

Political forces have revealed themselves and are not without their own motives. Political maneuvers have caused Bitcoin to drop suddenly in value overnight. So what can a consumer do? Will this be a repeat of history again? Can we prevent centralization of Bitcoin to protect consumers? Does might always make right? What happens when Bitcoin crashes, and people lose confidence in the stability of its price? Are the buyers of Bitcoin not forced to "trust" the distribution, price and lack of political pressure for it to survive? Can independent currencies compete? ***Why has a fail safe method of price control not been implemented?*** As you will see below, a far superior and efficient method has been found. One that will inspire smaller and newer projects, and greatly accelerate the entrance of new ideas into the industry without fear.

"Pump and Dump", Propaganda vs. Technology:

The Bitcoin "boom" has driven independent developers from all around the world to compete and design innovative tools to help cure some of Bitcoin's problems. Although these "Altcoins" add utility value, and some are truly revolutionary works of art. Unfortunately, those advancements easily get lost in the sea of coins. Many investors have a hard time knowing which currency to buy because they look at price before utility. This has naturally given rise to "pump and dump" markets - similar to the penny stock industry. Unethical projects

tend to win these battles, because it is easy to trap naive investors with a fancy website. Many have been known to fabricate team members, photos, and development deadlines - incurring little (if any) expenses for doing so. This can "dazzle and amaze" investors, but is nothing short of a lie. This behavior sabotages honest developers who maintain a genuine interest in advancing the technology. In the end, you can kill the messenger, but you cannot kill the message! We want to see the "best ideas win". The only way to defeat corrupt price manipulation and propaganda is through controlling price in a positive, decentralized manner - more effectively than ever done before. Using technology to our advantage, we can remove the "market maker" and allow the coin to recover from savage losses. For example, suppose a currency has lost half its value. In a normal market condition this can be difficult to recover from without the introduction of new money. Investors who should be rewarded for their loyalty may find themselves suffering extreme losses. This is a daily occurrence in today's Altcoin markets. It happens at such a blinding pace that it becomes impossible for developers to keep up. Their projects take years to code and build, while investors gain and lose fortunes within days. In BitBay's example of decentralized supply control, we can freeze coins progressively until the price fully recovers. The supply can be dropped below demand for a deflation effect, forcing a price increase. Alternatively, it can be set above demand for an inflation effect, forcing a price decrease. The net result is a resilient currency that will naturally survive any market condition. Its price will not be a sole reflection of its popularity. Instead its price will be a reflection of both supply and the people backing it - no matter how few.

Other attempts at pegs so far, and their drawbacks:

BitShares[3], NuBits[4] and Tether[5] are some examples of different attempts at price control. This paper will discuss the issues surrounding these methods and explain why I believe them to be centralized and inflexible. BitShares issues assets called "Smart coins" which, in theory, is a brilliant idea. It leverages both short and long positions, and issues collateralized assets. These assets can be redeemed in exchange for BitShares. However, there are a couple major drawbacks. One is price manipulation and various "Black Swans", as outlined in their own words on their website. It is centralized, because the collateralized asset can never be worth more than BitShares. If the price of BitShares collapses, then so does the asset. In the end, it is a useful tool. However it is not sufficient for a long term currency. The price and speculative nature of BitShares is its ultimate weak point. This structure is also a voluntary system which relies on traders to be available to take these positions. With this process, offers need to give enough incentive for a trader to profit. It can be improved on nonetheless.

Another example of price control is found within Nubits. Unlike BitShares, Nubits uses "custodial wallets" and is closed source, which ultimately centralizes the entire model.

These custodians place large buy and sell walls on the markets to force the price into a certain range. This, in turn, gives them the unfair power to pump or dump the market at will. In theory, this system is similar to BitBay's, as they do attempt to control the number of coins in circulation. However the Nubits method is not decentralized and is subject to inflation. Essentially, Nubit's coins are added into existence by voting when the price is too high (above 1 USD). Then, extremely tempting interest rates are offered for users to "Park" or freeze their coins when the price falls too low. That is not sustainable since an investor, at some point, would want to reintroduce their gains to the market (which is claimed to reach as high as 400%). Even if a solution was found to mitigate this, it still adds an unnecessary layer of complexity, and continues to remain centralized.

Lastly there is Tether. ***To date, Tether has proven to be the most popular peg despite being the riskiest.*** It is a system based on trust. The ones issuing it offer to buy back Tether for dollars. The amount of Tether offered then increases based on the fees gained from those trades. Again in theory, this is a great idea. However, it is still centralized. If the founders fail to honor this system, investors would suffer catastrophic losses. Due to its centralized structure, Tether is also vulnerable to national government regulation. If deemed illegal by certain countries, every coin could become worthless. BitBay prevents this problem through pure decentralization and voting.

It is worth mentioning that some currencies have offered to back their coins with commodities ranging from hemp to stocks and gold. Most of these are still subject to speculation, dishonesty, and in some cases, offer more valuable commodities to encourage buying of coins (front running). The buying power can massively exceed the requested amount of commodity. Then, when the price is equivalent to the commodities value, the founders bail out and leave investors holding worthless coins. These above mentioned systems do not offer any flexibility in the price and currently only track the dollar/commodity.

There have been many new currencies attempting to create a "stablecoin" with low volatility, similar to BitBay. Although the concept for BitBay's peg was proposed in 2014, it was not until later that other coins would attempt to implement a dynamic peg themselves. These proposals have fallen into various categories such as: backing with other cryptocurrency assets, controlling supply, and various trading techniques. If one volatile asset is used to back another volatile asset, then the coin may not have long term stability. Recently, a technique for controlling supply similar to BitBay's peg was described by a company called Basis[6]. The idea of Basis is fairly interesting, however their system is voluntary. Users are offered incentives to burn coins for a different asset, which is later used to issue new coins. This method is very similar to Nubits, but more decentralized. Unfortunately, voluntary systems usually need to offer extremely lucrative incentives. This shifts money into the hands to a

select few, and is also not ideal for a strong foundation.

Interestingly, all of the aforementioned techniques are actually great secondary methods of price control. Lucrative trading offers, backers, crypto commodities, real world commodities, and even dividend payouts are all great ways to help support an economy. However, if any are used as a primary method, they will not be able to force price when a crisis hits.

Enter BitBay: the dynamic, decentralized, and non-collateralized peg

a. GLOBAL SUPPLY RATE

The BitBay technique will now be described in the simplest terms, and it will be technical enough that anyone can code their own version of it. First, there needs to be a way to fairly determine the number of coins in existence. However, custodial wallets, external systems, and commodities must not be relied upon. Only those invested in the currency (the users) should have a say in the future value of their commodity, no matter how many their number is. Therefore, only those staking the currency can vote for the supply to increase, decrease, or stay the same. These votes are then counted in intervals, which in turn, effects users when they attempt to spend. In addition, there are alternative systems to voting, and the decision to freeze or unfreeze can be purely based on an algorithm. However, BitBay will use the voting system because it allows users to change their algorithms and adjust to market conditions without forking. As of today, staking and mining in Bitcoin have been fair methods to determine who will "verify" transactions in each block on a blockchain (the sequential history of all transactions). The mining and staking systems in all coins run as a "random" competition. Otherwise known as consensus mechanisms, these systems are continually being further improved and decentralized as new ideas for security arise. Mining and staking systems are provably fair - as long as they are well distributed and random. Newly minted coins are given to Bitcoin miners who win a block. The competition itself is determined through a math puzzle which varies in difficulty as competition increases or decreases. Staking is similar in that regard, however instead of a math puzzle, new coins are given based on random probability. The probability of receiving a reward is directly proportional to a staker's pre-existing balance. Staking systems are arguably superior as they don't waste billions of dollars in electricity. However up until this innovation, they had nothing backing up their price. This is not to say that funds spent on an electric bill forces the price to be a certain value, but it should not be considered irrelevant either. The technology arms race has certainly impacted the price of Bitcoin.

Blackcoin's "Proof of Stake 3" rewards active nodes in the network for verifying transactions. It also has a custom staking command which spends the winning stake in a transaction. This is useful, because it allows BitBay to integrate voting when a user wins a block. BitBay is currently using this code base within its design. Since mining and staking systems are

fair and random, everyone gets a fair chance to cast a vote on the total supply rate - even if there are thousands of users. Nevertheless, staking also has the ability to centralize (large players can buy up the coin and control the stake). To combat this, BitBay's system will actually increase the competition for voting power and interest, thus naturally balancing the amount of people holding frozen coins. ***Frozen coins, by definition, cannot be moved. However, if those coins are used to stake, an exception is granted.*** An exception can also be granted if a user locks those coins for one month to their new destination. A user may wish to acquire frozen coins at a discount to get more staking power, voting power and interest. The frozen asset can be considered a feature. Additionally, the frozen assets have advantages as a bond and as a lending collateral which will be discussed later. This is beneficial for distribution, as a person who holds more stake in the currency is likely to vote in the favor of its health. They could have the option to vote in favor of liquidity at the highest realistically attainable value. This value would require a balance of both high volume (to sell coins) and dynamic supply (to maintain increased price). However this does not mean liquid assets are always favored over frozen ones. The frozen assets could potentially be higher in supply and lower in price, giving users the benefit of receiving additional yearly interest (higher probability of stake reward).

Blocks are staked approximately every minute in BitBay. In Blackcoin's POS 3 protocol, a "voting address" can be paid by burning coins to the blockchain using a tiny amount of the stake reward. Only the votes coming from a valid staking block are recognized, making it impossible to game the system. For example, votes can be in favor of a +1% increase in supply, a -1% decrease in supply or no change in supply. Automatically, each vote is based on an algorithm. This algorithm favors high volume and a healthy price. At any time stakers are allowed to diverge and cast a vote personally every time they win a block (to influence the coin in the appropriate direction). Please remember, if needed, the vote can be forced to follow an algorithm and remove the human element entirely. This feature is not a rule that will make or break this system, and the end result remains the same. ***The supply will be controlled by a set of rules based on voting or an algorithm.***

To summarize this first section:

Total supply increases and decreases daily, based on votes or an algorithm and dependent on stakers' preference. No individual or entity controls the coin, however the algorithm will always favor healthy volumes and prices. In BitBay, users have the option to design a custom algorithm and submit it in the API. This allows the community to make changes when necessary. Votes are counted long after the block confirms to avoid issues during a reorganization.

b. ENFORCING THE FREEZING OF COINS IN A FAIR WAY

Bitcoin and Altcoins are structured as a series of inputs and outputs. A payment of 125,624 coins is seen as an input of 125,624 coins. Breaking that input into two parts of 100,000 and 25,624 would require the user to spend a total amount of 125,624 and pay themselves the two parts in change. Consequently, scripting a Bitcoin transaction works similar to cash, in that a user can request change on a payment. The Bitcoin scripting system is therefore a set of rules determining how coins can be spent. With BitBay, the first step users must complete is to count the votes. When they see a transaction (for example: Alice wants to spend 1,000 coins), they check to see what the total supply rate was when she received them. Alice can "tag" her output (the destination) by saving some data explaining when fragments of coins are set to freeze and unfreeze. If a coin is not tagged, it is assumed that it was received at 100%. For the sake of brevity in this paper, assume 100% means "100% of the total coins in circulation are liquid" and 0% means "all of the total coins in circulation are frozen". So 60% means "60% of the coins are liquid and 40% are frozen". To illustrate this, let us say Alice is spending 1,000 coins that were first received at 100%. She is sending them when the currency is 60% liquid. The network will then prevent her from spending all 1,000 coins. Instead, the network will only allow 6/10ths of the coins to be spent. This means she must pay herself 4/10ths (400 coins) as change. Those coins are now frozen at 100-60. So this means if the currency continues to deflate she will not be able to normally trade those coins. Now the next phase of understanding this system is to realize that everything works in terms of coin fragments and proportions. For example, one payment was sent to herself frozen at 100-60, however the other payment was sent liquid at 60-0. ***So both the frozen asset and the liquid asset freeze and thaw at completely different rates!***

c. EVERY COIN IS UNIQUE; THE NETWORK REMEMBERS EVERYTHING

In traditional cash systems (like the US Dollar), money always appears to have the same liquid spending power. However, this is not actually true. The dollar deflates at an alarming rate every year, as goods and services become more expensive in relation to the dollar. The difference between the dollar and BitBay is that the dollar does not display when and how its value has changed. It is only realized after the fact, when prices of goods/services rise and the dollar falls. In the end, a dollar is always a dollar.

In BitBay, no two coins are alike, unless their rate of freezing or unfreezing is the same.

However one thing is obviously true about the dollar: its value is not transparent. People don't see it marked on each bill. If the peso goes up or down, its trade value in dollars goes up or down. This dramatically affects Americans' spending power abroad. However, 200 pesos still appears to be 200 pesos to the average consumer, even when there are other forces at play.

This now brings us to BitBay, where every coin is transparent. Each coin displays both its exact liquid or frozen value, and its rate of inflation or deflation. In the previous example, Alice was only allowed to spend 60%. Let us say for this new example, she sends that 60% of 1,000 (600 coins) to Bill. Now Bill has 600 coins that are liquid from 60-0. Alice has 400 coins that are frozen from 100-60. Now let us say the network votes for inflation or "unfreezing/thawing". Over the next few weeks, the coin inflates 20%. Now Alice sees that she has the ability to unfreeze some of her locked coins. She decides to send a second payment to Bill. Now from her frozen asset, the global supply rate is now 80%. Since she last froze this at 60%, she now has 1/2 of the frozen coins that are completely liquid. So if she wants to spend 1/2 of the 400 as a payment to Bill, she will send him 200 coins that are liquid from 80-60, and she will receive change in an output of frozen coins that are frozen from 100-80. This creates a really unique paradigm. Bill now has two liquid assets that are of a different value! He has premium and sub-premium liquidity. Why? Because the coin he just received from Alice is set to deflate faster than the previous one he received. The 200 coins deflate at a rate of 1/20th (since 80-60 is 20) and we acknowledge that the rates might deflate as high as 1% a day in our example. However, the actual change in supply rate will not be linear, it will compound. Also the change won't necessarily be daily, it depends on how frequently votes are counted. On the other hand, Bill's other 600 liquid coins will not begin to deflate until the network hits 60%.

*****Please note: Bill did not see his 600 coins increase in size, as that would be unfair to Alice, who holds a frozen share from her investment prior to Bills.**

While a system which directly increases and decreases supply through burning and creating coins is interesting, (as opposed to freezing) it does not change a users percentage of what they hold from the total market cap. Therefore, it has no true economic impact. The only acceptable method of implementing this system is to have the network remember each individual coin fragment that is sent and received. It must inflate and deflate each fragment based upon its arrival and departure. Bill also gains an advantage when the coin inflates.

His 600 coins become more valuable in the sense that they have increased liquidity. This protects them from deflation for a much longer period of time. However, his other 200 coins are in immediate danger of being deflated, so he is compelled to spend those first.

> *This idea of premium and sub-premium liquidity and premium and sub-premium frozen assets is completely new in cryptocurrency.*

Nevertheless, it can be seen today in the stock market where holders own shares with different liquidity characteristics. Some shares offer holders different powers, such as voting power. Additionally, there are both premium and sub-premium shares tied to a company based on the contract. For a new example, let us say that Jane asks Bill for 20 coins in payment for a service. What is to prevent Bill from subdividing his 200 coins and selfishly keeping the premium coins for himself? He could easily pay her 20 coins that are liquid from 80-78, while keeping 180 coins that are liquid from 78-60. This means her 10 coins can freeze 50% by tomorrow. She clearly is being paid a less valuable asset and she might not be familiar with economic theory to understand this new system. How can a system like this work for day to day transactions and payments? To answer this, we will refer you to the next section.

d. FORCING FAIRNESS, SUB-PREMIUM LIQUIDITY REQUIREMENTS

In the case of the 20 coin problem above, Bill can be forced to send Jane his premium liquidity first. There are two ways to accomplish this. The first method (and simplest) is allowing the software to handle everything automatically. For standard users of these markets, only the most premium liquidity possible can be accepted. The users are alerted when someone attempts to send a payment of less-desirable coins, and the software will discourage them from accepting it (if it is in exchange for goods and services). The users are also allowed to either increase or decrease their tolerance of liquidity for the payment of goods and services. ***Inherently, users should not have to think about it.*** The other method of enforcement is to have the miners enforce a maximum speed of deflation when sending as a hard rule. For example, let us say a 5% daily maximum deflation rate is chosen. With that in effect, Bill cannot send his 20 coins (set to deflate completely in two days), as that entire output could potentially deflate at 50% if the stakers move the rate by 1% the next day. Instead, the 5% maximum means he must send a full premium portion and range of

his 200 coins. Please note: Bill cannot be selfless either and spend (for example) 10 coins from 62-60, keeping 180 coins (from 80-62), as that would also be violating the terms. His 180 coins would be deflating in 18ths, which is faster than 5%. Instead, he subdivides the output and makes two new outputs. One is 180 coins (divisible in 20ths, from 80-60) and the second is 20 coins (divisible in 20ths, also from 80-60). This is completely fair. The probability of the coin supply deflating for 20 consecutive days is extremely low. In the rare event of this occurring, it would only encourage Jane to spend the sub-premium coins first as Bill did. She would also need to be holding some liquidity if her 20 coins began to deflate. This is not a negative event, as it only increases demand for liquid coins, and encourages the trading and buying of coins on the markets. If the coin deflates 1% tomorrow, and Jane's 20 coins comprises of 1 frozen and 19 liquid (divisible in 19ths), then she needs only one coin in the range of 59 or better to spend it. The daily maximum can be flexibly customized to support the utility of the network. The impact of a maximum deflation spending requirement also affects frozen coins that have recently thawed. For example, if the coin began inflating more (lets say 10%), and Alice wants to spend another 100 coins, she must send enough coins from her premium 600 to match a smooth deflation rate of 5%. With these two methods of enforcement in mind, the question then becomes: Should this rule be hard coded through the stakers in the network, or should it be automated on the software level? The answer is that the software can handle the issue on it's own. Why would Jane accept or send sub-premium coins in the first place? The entire "maximum deflation" rule can be enforced on the merchant level, while the software simultaneously structures itself to display how much merchants will accept within a users balance! For example, the software warns the user not to accept coins that are less liquid than 5% per day in exchange for services. This allows merchants to also compete for the sub-premium liquid coins (creating a healthy secondary market). To make it all seamless, the merchants and buyers are automatically filtered and the user does not have to think about it.

> *You can even write your software to contain 3 balances; frozen, sub-premium, and liquid.*

The liquid coins would be the balance they always see as available, and the sub-premium and frozen coins can still be spent, but under certain conditions. You might have noticed that inputs will eventually be fragmented. This fragmentation can bloat the blockchain and cause tiny little coin shards in everyone's account. Even though very large transactions cost more in transaction fees, that in itself is not a deterrent and will not prevent fragmentation. Which brings us to the next section.

e. DEFRAGMENTATION & VARIOUS METHODS OF COMBINING INPUTS

Is there a way to perform system maintenance and recombine inputs so that we are not left with accounts having various inputs of various ranges? What are the rules for recombining inputs? Let us take an easy example first. Let's say that Bill has 10 coins from 100-99 and 990 coins from 99-0. Well, it is clear that he can recombine those coins into a single output of 1,000 coins at 100-0. But what happens when Bill has 10 frozen coins from 100-98 and 995 liquid coins from 99-0? Well, unless he wants to create more shards, there is no obvious way to recombine these. He can take 5 from the 10 to make one input of 1,000 (from 100-0), and then make 5 coins that freeze at 99-98. However, that doesn't completely solve the problem. He still has 2 inputs, although they are slightly "prettier". The 10 coins deflate at 50% per day maximum, and the 995 coins deflate at 1/99th (starting at 99%). However, there appears to be a brief period of time where the 995 coins do not deflate. Consider that these proportions are not always pretty fractions. They can be 12,405 coins (liquid from 99-63) and 1,012 coins (liquid from 100-87). They can overlap or be separated by gaps. So can the input of 12,405 coins be made slightly worse, while making the 1,012 coins slightly better? What method would have the same net result? Could we add the coins together and smooth out the rate of deflation by combining the fractions? For this example, let us assume that we are 100% liquid. We will add 12,405 + 1,012 to get 13,417, but now what? How do we count the day from 100-99 where nothing happens to the 12,405 coins? Do we simply use the fraction of 1/37th as opposed to 1/36th? Do we perform more complex math? Also, how do we want to combine them? Let's suppose we combined 1/37 to 1/13 to find the lowest common denominator: 25/481. So now the 12,405 coins which previously froze at 2.7% become 13,417 coins that freeze at 5.19%? Is that fair? Probably not, as we are significantly diluting the 12,405 coins for the much smaller amount of 1,012. If constructed this way, combinations can still be made to "game" the mathematics. Even if an equation for recombination of uneven fractions was found along with a protocol that was "fair", it may not be correct. It might create a normal distribution in the center of the coin, subsequently creating lower amounts of premium and sub-premium liquidity on the bookends. Also, we are assuming a linear 1% per day maximum or minimum deflation. Once the coin is 50% deflated, 1% based on the total supply now deflates the remaining liquid coins at 2% a day. This increases the difficulty of meeting the maximum deflation daily requirements. ***Thus, we must either create a method of compound deflation, or one that readjusts itself at certain intervals.*** Additionally, if our supply was 1 billion coins, we would probably be working in billionths and not hundredths. We could work in "steps" instead of days, and work with a complex function instead of fractions. Our deflation rate calculations would also be in billionths. A limit could be placed on rate of supply decrease, as that will

have an effect on the average amount of input shards in each account - especially after the system gets extremely deflated. So what is the solution?

Realistically, there are a maximum number of fragments that can possibly exist. Here is a more extreme and simple example to prove this point. Assume that the daily global deflation rate was a linear 25% and there are no “maximum daily deflation requirement for sending coins”. This allows only 4 possible total supply percentages! They are: 100, 75, 50 and 25. That is 24 potential inputs of completely different ranges. For example, 10 coins (100-75), 10 coins (100-50), 10 coins (100-25), 10 coins (100-0), 10 coins (75-50), and so forth. This technique of defragmentation is to fragment and recombine. We look at intersections and combine them where they meet, creating brand new inputs. Let us say that Bill has 10 coins (from 100-50), 20 coins (from 100-0), 30 coins (from 75-0), 75 coins (from 100-25), 100 coins (from 100-0), and 200 coins (from 50-25). If we break all those coins up into the smallest possible shards, we can recombine them all into 4 shards as opposed to 6.
Here is how the accounting magic is done: 5(100), 5(75), 5(100), 5(75), 5(50), 5(25), 10(75), 10(50), 10(25), 25(100), 25(75), 25(50), 25(100), 25(75), 25(50), 25(25), 200(50). Which is now extremely easy to reduce to 4 parts. 60(100), 70(75), 265(50), 40(25). All that was needed was to find out how many coins were frozen during each change!

*****Please note: the inputs were abbreviated for simple reading, by listing when they were liquid.**

We fragment first and then simply recombine. With this method, a complex problem becomes extremely simple.

Another example: There are 435 coins in total. Bill will have 60 coins (frozen at 75%), 70 coins (frozen at 50%), 265 coins (frozen at 25%), and the remaining 40 coins (frozen at 0%). This also means that coins can be made into new inputs by finding out where they intersect. In the end, if there are 5,000 possible permutations of global supply rates, there would be a maximum of 5,000 inputs per account when defragmented. However, it is reasonable to assume that those inputs can be combined into much larger inputs as well. For example, if Bill wanted to spend his 60(100-75) with his 70(75-50) he could make 120(100-50) and give the remainder of 10(75-50) to the miners as a burn. Of course he would not burn that many coins, this is just an example. However, there is still a problem. We don’t want 5,000 inputs, as that is still far too many. Eventually, each fragment that a user holds will not be similar in size. If a user wants to send their whole balance, they would end up with thousands of signatures. Therefore, the next concept is letting users combine uneven inputs into a single input. We expressed earlier how this was not ideal. However, it can be achieved if the inputs

are tagged with a description of the coins involved. We can tag with ranges and individual fragments if we so choose. Since this will be a large amount of data, it is possible to either increase the capacity of metadata our coin is allowed to send with each transaction, or have the network make the assumption and store the information in a separate database. An example of a tag might look like this: 100 coins {100: 12, 99: 4, 98: 3, 47-40: 12, 34: 69}. However, instead of percentages they might be step numbers. “Step 1” would be from 100%-99% , “Step 2” would be from 99% to 98%, and so forth. Miners need to check the tags to make sure they are accurate. They will do so by checking the previous inputs. This brings us to the next section about mining fees, their liquidity, and if users should burn the remainder.

f. HOW MINING FEES ARE HANDLED

Every transaction pays the miner a certain amount in fees to process it. Currently BitBay is set to inflate at 1% annually. However, that part does not mention the many fees a miner will collect in each transaction. Fees are currently calculated as pay per kilobyte. Due to the fact that fees are paid by omission, a miner would have to look at which inputs pay their respective outputs, less than the amount put in. This calculation can be tedious. There are a few ways to handle this system. The first is very literal. A miner can simply receive the exact amount in shards that were omitted, or the fees can be forcibly burned by the miner. Burning the fee is not sustainable long term though, because it means some liquidity ranges will have fewer coins than other ranges. Rewards in mining should come in at a full range, as if they were brand new. Thus, it’s best to know what is going to be collected as a fee by simply looking at all of the unspent shards.

g. THE RATE OF DEFLATION, VOTING FREQUENCY

Previously in this paper, we showed a linear 1% change in daily supply rate to make the system easier to learn and understand. This means, votes would also be counted daily. However, it is possible to count votes hourly (or every few hours) to overcompensate for the wild fluctuations of crypto. ***The frequency of vote counting dramatically changes the temperament of this system.*** A linear 1% (based on the total number of coins) would compound very quickly. In order to make it 1% relative to current liquidity, we could readjust at 50% to .5% and yet again at 25% to .25%. Unfortunately, this is not a smooth compound interest rate. If we don’t implement a linear adjustment of 1% and instead do a compound 1% for each change based on the total liquid supply, this creates a parabolic curve. It then

begs the question, are we working with fractions or not? It is possible to still notate the coin ranges with shorthand notation, even when it is using complex functions and is no longer a clean fraction. There are a few ways to handle the problem of how to express these objects. The first is to prevent the curve from being perfectly parabolic. We instead make it linear, and correct it when it reaches a certain percent. For example, adjust from 100-50, 1% at a time (so it becomes a maximum of 2% total change in supply), and then correct back to 1%. This makes the new global deflation at .5%, which is relatively 1%, then adjust this from 50-25% and so forth. The advantage created from this system is the ability to work with nice, whole fractions and create easy to calculate ranges (that are not entirely based on functions). We can force coins to split when they become non-linear for more comprehensible inputs. It would create easy and simple ranges for the user of BitBay to calculate.

However, this also makes it difficult for merchants to enforce their maximum daily deflation for sub-premium liquidity. It is also slightly jerky for investors, and creates significantly more aggressive rates. Essentially, it is a debate of economic theory and taste. If a compound rate is chosen, then we are forced to work with functions instead of fractions. For example, if we had 1 billion coins (which is ten million * billion satoshis) then we are working in fractions of 10 quadrillionths. If we make the deflation rate 1% daily in either direction (and we make it compound), then we can know that the daily rate will be predictably smooth. However, this makes calculating the frozen/unfrozen supply and labeling inputs (by ranges) defined by their numerators coming in and going out - and always derived from functions. Coins can still be recombined this way, but each “step” or “day” is the new representation of a range. The amount of coins that are frozen (per 10 quadrillionths) changes with each step. We can also still keep our fractions in the billionths if we round the numbers. For example, we start at 100% and allow a maximum of 1,000 steps of deflation. Then, our numerators for the last 10 steps (990-1000) would be [47,735], [47,258], [46,785], [46,317], [45,854], [45,396], [44,942], [44,492], [44,047], [43,607]. Meaning if a user wanted to spend an input of a billion coins and the coin was deflated to the absolute maximum of 1,000 steps, then they would have to freeze all but 43,607 coins. Of course, the numbers will never be perfectly even, so the users will burn any remainder.

> *There are several ways to code this, and if this decentralized supply control system catches on, many others will attempt to use their own approach.*

This is a new frontier where the best economic model will be found through experimentation. Many mathematicians will love to experiment by copying this coin and making their own rates. Unfortunately, this does mean that many currencies will also abuse this system for

artificial “pump and dumps” - on a level far beyond anything Bitcoin could have imagined. Ultimately, it will be immeasurably cheaper to pump a coin. A single person who living on the poverty line could do it themselves. However this is not too much different from premined coins that are already very rampant in the market. If this system becomes abused, a cryptocurrency exchange might consider listing the market capitalization by the amount of coins in circulation, instead of basing it on the total number of coins that could theoretically be in circulation.

h. THE SOLUTION TO THE TAGGING PROBLEM, HOW BITBAY DOES IT

Up until this point, various techniques for shorthand notation of coins were described. In reality, shorthand notation, combining of fragments, and having users software combine fragments does not solve the much larger problem of bloat. Although, it will inspire future research on expressing very large objects in a more compact way. They are really just practices of compression. In the end, it is inefficient to have users publish the tags to the blockchain.

Alternatively, users who synchronize the chain and check the blocks always make the assumption of the liquidity being spent. This saves a tremendous amount of data, because each output does not need to explicitly state what it is doing. Then, users tag transactions in the output (up to a certain amount of blocks). Once enough blocks pass, they prune them and move all of the liquidity into account driven liquidity pools. ***As a blockchain grows, the number of inputs is exponentially larger than the number of user accounts.*** It is not practical to store this data from each input. If there is a reorganization beyond the pruning point, then the users simply resynchronize the peg database. Outputs are tagged in steps and the interest rate is compounded. Thus, if 1% per change is chosen, the supply then goes to 99%, then to 98.01%, and so forth. In this example, 99% is step 1. Each output simply divides the coins, immediately showing how many coins are set to freeze at each step. So it may say {1: 10, 2: 42, 3: 19, 4: 12.}, and so forth. There is no shorthand notation used. Instead, pruning by account and having the network assume is the best technique for scaling the system. The peg database stores files referenced by the beginning of the txid hash or the account number. Therefore, there are roughly 45,000 file combinations in our implementation. If we assume that every 200 blocks are pruned, then even with a million unique accounts (containing hundreds of thousands of transactions within the 200 blocks), the database would hardly exceed 100 gigabytes. Data can be compressed in “bson” (also known as binary “json”) averaging around 10kb of tagging data per output, if 1,200 possible

steps are used. This means to deflate X% 1,200 times maximum. In addition to the hash files, if we prune every 200 blocks, we may keep the previous 300 blocks and store data about what happened in a file. The files will contain what was spent and what was taken from each account's liquidity pool, if applicable. When a user spends a transaction, the network will check account liquidity pools and choose what to take. It will always take an even range and decide which parts to take the remainder from sequentially. Also, since transactions will take from pools, we must know the sequence of events if we see multiple transactions in a block taking from the same pool. In this scenario, a user must set nlocktime, and the memory pool must organize transactions to be processed in the order they were received. If a user wants to spend their liquidity to a specific output, they can do so through burning a message. This message must be burned in the same position as the inputs index, and can tell which output to pay to. A user will use the same burning technique when deciding to send frozen funds which are bound by a one-month lock. This will also happen when they want to voluntarily freeze liquid coins, which will be described later. Lastly, a file is also stored for the basic peg data and voting data. During testing, this solution has shown to keep very small files which do not overload the RAM, give good processing times, and reduce the total amount stored to disk. Also, when the network evenly spends liquidity, it becomes increasingly difficult for a user to hoard premium liquidity. For further research, shorthand notation can be explored to improve compression and perhaps allow users more interaction with their liquidity. Block files are reviewed during a reorganization, and because it was known exactly what was taken from a liquidity pool, the fragments can be added back to the pool.

i. MOVING "FROZEN" COINS WITH TIME LOCKS: BONDS, FUTURES, AND LOANS/ TRADES

In this section we will propose a bylaw which allows users to move frozen funds. This is for many reasons. First, having a completely immovable and frozen asset class would create problems if an exchange was to become insolvent. However, that is not the main reason for this unique feature.

> *Users will be incentivized to obtain frozen coins, because they provide more voting power and offer a higher probability of earning rewards.*

Although, if users could obtain them immediately, then they would not be frozen! Instead, the coins can be transferred slowly over a longer period of time.

For this example, we will use a 1-month time lock as the minimum. A time lock (also known as checklocktimeverify) is an output that can change its ownership at certain time intervals. It will appear on the blockchain, but the person who owns it may change at a later day, if not spent within a certain time frame. Instead of making users send transactions with checklocktimeverify, we will simply have them state their intention to move reserve funds, and the network will remember this event. So to make a "slow" coin you send it to the new address, and only after a month's time passes will the recipient be able to spend it. The liquidity doesn't change for what is being sent. The new owner must follow the same rules as the previous owner, if he wishes to transfer it after the month passes and its status is still considered frozen. This idea of coins moving at different speeds within the same network creates a beautifully decentralized way of having different asset classes, and each possessing a self-imposed difference in value. These assets may potentially trade at different market prices, since waiting months for an asset to arrive is significantly different from one that arrives immediately. To further incentivize these transfers, we also give a slightly higher interest rate to these coins. A user's reserve (frozen) funds will have much less voting power than their liquid funds. The function for determining this states: ***As liquidity of the total supply decreases, the voting power of a liquid output proportionately increases.*** Regardless, frozen funds still have voting power and can make a big difference.

The increased probability of earning rewards gives frozen coins a unique characteristic that is very similar to a "bond" or a "future". Keep in mind, this is a peer-to-peer future that needs no middleman or central bank. It is simply a future by its own virtue, since one day it may become liquid. This wonderful system also allows for a secondary trading market, and gives users an asset class that is both speculative in value, but very reliable for generating increased rewards. It will favor the wise investor who plays this market properly. Users can easily overbuy and oversell their frozen assets, not realizing their value. With that being said, the frozen coins also can serve as "trustless loans". Although I am personally not a fan of lending, ***a user could extend a "loan" without interest by simply swapping liquid coins for frozen ones.*** The lender benefits from having the extra staking and voting power, and does not even need to trust the person seeking liquid coins (assuming the price is right). For example, Alice has absolutely no liquidity, but she needs money to buy food. She is sitting on 10,000 frozen coins that she is unable to move. However, she decides to sell those coins to Bill. Bill is a long term investor. He has an abundance of liquidity and knows the future value of Alice's frozen asset. He will offer to buy her frozen coins at a discount. She can't use them for the purchase of goods and services, so she sells them to Bill at 1/10th their value. She receives 1,000 liquid coins, and she knows that there is a good chance she can

buy more frozen coins at a later date.

This system is self-correcting, trustless, and purely based on technology and knowledge. Both Alice and Bill are taking advantage of their resources. The trade is purely a product of supply and demand.

> *With this system, a society no long has to endure the corruption of parasitic lenders.*

Alternatively, peoples' creditworthiness could be solely based on the amount of frozen coins they can put up. This is not to say that Bill might not enter a contract with Alice to buy back the coins at a premium. However, that agreement is between the two parties as it should be.

j. VOLUNTARY FREEZING, CHANGES TO ESCROW, REORGANIZATIONS AND OTHER DETAILS

The peg itself can be combined with other traditional methods. As described earlier, there are some viable secondary methods of controlling price, aside from a dynamic supply. We include voluntary freezing for slightly higher interest. ***If a user takes their liquid funds and freezes them, they will gain double the interest of a regular user.*** This is also useful if they want to stake those coins and vote. The peg itself has various effects on the economy, the user experience, and other minor details. First, the BitBay software runs trustless 2 party escrow in the application. The coins in escrow could, in theory, become frozen while they are in escrow. In order to ensure mobility, the two parties must avoid sending coins that could freeze in the deposit transaction. This might mean setting aside extra change or properly timing the broadcasting of transactions. Transactions which have checklocktimeverify should ensure that the two different timing mechanisms do not conflict. One concern is the uncertainty of the exact rate change time due to reorganizations. A user could attempt to send coins that are (by chance) partially or totally frozen at that moment of the rate change. A user does not want to find his transaction in a later block, or have it invalidated because of a reorganization. At first, this might seem like a real problem. In reality the same type of issue could surely arise with Bitcoin's checklocktimeverify. A reorganization could delay a transaction and make it retroactively invalid if a double spend happens. In fact, Bitcoin must deal with double spend anyways. This is why on all cryptocurrency networks it

is recommend to wait for a certain number of confirmations before spending funds. While the Bitcoin scripting language has no concept of an invalid transaction after a certain date, it does not mean it cannot incorporate this. There are a few ways to handle this situation for our users. The first (and simplest) is to have the software recommend setting aside a small amount of change during transactions that might be affected by a rate change. Having extra change ensures that network will have enough funds to store when making the calculations. When setting aside funds is not an option, the simplest is to wait more confirmations. This could be a hard coded rule, which lets transactions near a rate change mature a little longer. It is also possible to have the users or memory pools make this decision. All software will want to show users their liquid and reserve balances, as well as any coins that were frozen in a time lock. Merchants may want to be aware of what level of liquidity they will accept. They may agree to accept sub premium coins or they may decide they only want the finest liquidity. At first glance, it seems like the loss of liquidity would be inconvenient for merchants and contractors. However, it's not really an issue because anyone can buy superior liquidity for use in their daily lives. Just like any free market, users will learn what is best for them. In the beginning, the software will aim to be as user friendly as possible, and not overwhelm the users with everything that is under the hood. It may just let users know that sub premium coins will not be used for sending to exchanges or merchants. Additionally, the memory pool must ensure prioritization for transactions with earlier lock times (to prevent taking from a liquidity pool out of order). A separate, small database will need to be used for transactions in the memory pool.

k. THE WEAKEST LINK, DEALING WITH EXCHANGES:

In this section, we will propose multiple methods of dealing with exchanges. Some will be more effective than others. In the end, the decentralized peg will force the price to go up and down no matter which method is used. Of course, the best exchange is a decentralized one, and we do feel that people will eventually prefer the safety and savings of a decentralized exchange.

1. Exchange API

The first way to deal with central exchanges is to have them query a "trading simulator" via an API. This means they will store exactly which coins are on the order book, and which coins are in everyone's account. The API will know the specific inputs of all accounts and simulate the balances. It will seek to obviate the process, not telling the user about the quality of coins they are buying, which will be completely random. A savvy trader will

deposit and sell off their sub-premium coins in the hopes of receiving a higher Bitcoin value in exchange than compared to somewhere else. ***The exchange would be required to query the API every time an order is posted or completed.*** The exchange may also want to keep track of who owes who. As long as the API is told about the inputs in each account, it can quickly calculate who owns which input (even though it's a bit more work). At this point, the blockchain is not involved. Only a balance in each account and the type of liquidity owned is necessary. The API can then calculate the balance of every account in both Frozen and Liquid coins. Users of an exchange would have a "BitBay Liquid" and "BitBay Frozen" or "BitBay Reserve" balance.

It is possible an exchange would want to trade the frozen coins in a secondary market. However this may not be supported by the API in the beginning, in order to reduce the liquidity of frozen coins and increase the challenge of trading them. We do believe that exchanges will be able to query our API on each trade, as they have that data on their end. However, in some cases they might not want to go through the trouble of querying the trading simulator for each trade. With the right data supplied to the API, this can work out fine. If for example, they have their own built in IOU system, then the problem of calculating balance is possible. We only need to know who owes whom, what the inputs of each account are, and each user's entire trade history. Even with just the users personal trade history, the software can figure out a users balance.

There is a second thing needed in addition to balance inquiries. Coins that are liquid may become frozen while they are live on the order book, and this may create the appearance of a "paper tiger" or an artificial wall. A purchase of that order would only have access to the liquid coins. It is possible to allow the user to buy the frozen coins along with the liquid coins. However, this may disappoint a trader who expected to buy liquid coins, only to find out many of them are frozen upon receipt. To solve the problem of "paper tigers" we can ask the exchange to clear all live orders every time a supply change happens however that would discourage trading because it forces traders to post again. The other solution is to force frozen coins to return to the rightful owner upon purchase. This only works if the exchange notifies the software on each trade. For example, a trader might pierce a wall of 1 Bitcoin with only .1 Bitcoin, assuming 90% of it was frozen. They would only receive what was liquid. Either solution is acceptable and works for the trading experience. However instead of clearing the order book, the exchange may also decide to simply adjust the amounts when a change happens on the live orderbook. ***That is of course the best solution. The exchange needs to only track when rate changes happen.***

The more data supplied to BitBay the better. Because accounts prune to big pools, it will be important to know if a deposit address is shared. There are a few ways to handle liquidity. It

is likely best for the exchange to not share deposit accounts, however, they can as long as deposit transactions are known for each user. The advantage to not sharing deposit accounts is that liquidity for each user is clearly known. You have more control over what liquidity gets paid out, and useful data is not pruned into a pool. The advantage to sharing deposit accounts is having fewer inputs during withdraws to achieve a good mix of liquidity. Any compromise of the exchanges database could mean loss of liquidity for a user. Withdraws may have to pull from many different inputs to achieve the proper liquidity. When a user posts sub-premium coins to the market, is it really fair to the buyer if they can't see the liquidity? Of course the exchange could have a minimum of acceptable liquidity for deposit.

Another option is to have all coins within the order book share liquidity, either in groups or in total. This means posting an order might improve liquidity slightly. The software will have to remember what each user deposited to the "giant pool", so when the order is withdrawn, they maintain the same coins. Once a user buys coins, they would choose an even range to deduct from the liquidity pool, making it as fair as possible for them. It is possible for all the coins on the exchange to become sub-premium, and this is just a natural consequence of the peg. It might be beneficial to inform buyers of exactly how liquid their purchase is before confirming it in any scenario. This would require the exchange to make a very small change to their UI for confirming purchases. Additionally, it may be a good idea to restrict the availability of sub-premium coins. The exchange could consider that sub-premium "thaws slow" and "freezes fast". Withdraw of frozen coins would have to be time locked. Depending on what information the exchange supplies and how accounts are set up, it may result in a slight variance of liquidity on withdraw. Because exchanges also maintain a significantly higher volume in transactions per second than a blockchain, multiple threads might be needed, in addition to faster running code. In extreme cases, it still may result in some latency or restriction on trades per second. To summarize, an exchange should give the API balance inquiries, notification of a completed trade (if trade history is not sent with balance inquiries), and requests to update the orderbook. They may also supply deposit transaction IDs, notification of a trade posted, and requests for withdraw of frozen funds. It can optionally change the UI for notification of liquidity, and it has the option of clearing the order book during a rate change, waiting for a trade to resolve it, or updating the orderbook. The software will change how it handles accounts depending on what the exchange wants.

2. Ignore the global supply rate:

The worst-case exchange scenario is ***"Let it sort itself out eventually"***. Users need to withdraw eventually, and when they do, they will receive a proportion of their coins as frozen. The fairness of such a system is highly debatable, as it is likely the exchange wouldn't even list a frozen balance to start with. Users would trade normally, and the API would attempt to

pay out based on their date of deposit, withdraw, and the total coins held by the exchange. The API would still need to spend across multiple accounts. Otherwise, the withdraw would be completely random as to what they would be receiving. The more random it is, the less likely a trader will see the benefits of the system. The user may notice that once the coins arrive in their wallet, it will be more difficult to get them back on the exchange. This also reduces the effect of freezing, as the exchange would be ignoring it all together. The only advantage is, the supply does decrease once the coins leave the exchange and exchange relationships may be easier to negotiate. There would still be no telling when stability would occur, as some users might keep their funds on the exchange as long as possible. This option is not ideal, and it is probably better to not use this exchange if those are the terms of engagement. However, if no other options are available, this still has a stabilizing effect because frozen deposits in attempt to bypass the peg would be declined. If such a system is used and an automated algorithm is used based on that exchange, it can also interfere with the voting system. This is because that exchange could prevent the price from properly correcting and it's possible that their data would have to be ignored. If other exchanges do it properly or better trades are offered elsewhere, it creates a decent arbitrage opportunity, which might incentivize withdraws from the problematic exchange. Regardless, it is better to trade direct than to deal with this type of exchange.

3. Use a decentralized exchange:

There are decentralized exchanges such as BitHalo/BlackHalo[7] and BitBay itself. Other exchanges have claimed to be decentralized, such as OpenLedger[8], although they do not completely eliminate counterparty risk. At this time of writing, there are not many decentralized exchange options. However platforms such as ***Blocknet[9] and Komodo[10] are working on atomic trading and will list BitBay upon completing their work.*** Despite the serious danger in using exchanges, people still have not learned their lesson and continue to use them without regard. Cryptocurrency exchanges pose a huge risk to consumers, as there have been many cases of internal theft. With a simple claim of "being hacked" they are able to avoid any legal consequences. This plausible deniability has resulted in some outrageous thefts, including the infamous Mt. Gox scandal where the exchange "lost" close to one billion dollars in Bitcoin. It can be estimated that nearly half of the entire Bitcoin supply has been stolen due to 3rd party escrow and exchange loss over time. Furthermore, if BitBay was to allow trading on an exchange, there is always the risk of "Phantom BitBay" and "Phantom Bitcoins", where the exchange simply sells coins they don't have. Unfortunately, exchanges have been known to do this and other equally unfair behavior. Some exchanges buy coins they don't have the funds for, as there is very little (if any) oversight to their accounting. With BitBay's dynamic peg, there is also a concern of this. The exchange might try to sell frozen coins, hoping users will not notice until it's too late.

This corrupt behavior is the very reason why decentralized exchanges are a logical alternative. BitHalo/BlackHalo (and now BitBay) was the first platform in the world to perform smart contracts (years before Ethereum). Using a technique called "double deposit escrow" AKA "unbreakable contracts" all arbiters, middlemen third-party escrow are naturally eliminated.

> *The advantages of these contracts is truly world changing. However people are still warming up to Bitcoin and have yet to wrap their mind around this new technology.*

The method of decentralized exchange within this platform would be through micro trading. A deposit is made in Bitcoin/Blackcoin/BitBay and the remainder is traded in multiple transactions. This method of trade is hardly used, as it requires traders to trade manually. There is currently no decentralized exchange interface (with charts and order books) in Halo. This means the orders need to take place OTC (over the counter, on internet relay chat or other social platforms) or on Halo's decentralized markets themselves. This has a few drawbacks. The orders would confirm slowly (users would have to wait for the Bitcoins to confirm) and the live orders could bloat the blockchain.

The concept of this double deposit micro trading was proposed in "NightTrader", which was the name for Halo's decentralized exchange. However at the time of writing, this platform has not been built, due to the concerns that the micro-trading method was too tedious and slow for traders. While BitHalo/BlackHalo and BitBay does support unbreakable contracts and decentralized markets, it has not been used frequently to exchange coins (although cash deals are becoming more frequent due to the extreme advantages). A faster version of NightTrader could be built; one that doesn't rely on micro trading and deposits. A series of fast, decentralized servers could be used for the orderbook and "Atomic Trading"[11] implemented. This would significantly speed up trading, would forgo the need for deposits, and would take fewer transactions to complete. "Atomic Trading" only supports coins with a feature called "checklocktimeverify". To our knowledge and up until this date, no atomic trading exchange has been completely built and experienced significant trading volume. Although beta atomic trading exchanges do exist in the coins Blocknet and Komodo. There was one other attempt at a decentralized exchange called "OpenLedger" which is affiliated with BitShares. While researching for this paper, it was difficult to isolate all of the details to their methodology. The BitShares platform seems to operate with different types of assets. In some cases, centralized exchanges can extend IOUs, such as CDDEK. In this case, a user holds their own private keys for their "Hot Wallet", and BitShares itself has a series of market pegged assets on their exchange. This does increase security as long as the asset is on top of BitShares. However, the market pegged asset issued can crash if the price of

BitShares crashes. Any exchange affiliations for cold storage is still at risk of loss. This is not a true method of decentralized exchange since there are several potential points of failure. We believe when a true decentralized exchange emerges and becomes popular, it will solve one of cryptocurrencies biggest problems and it will automatically give BitBay a place to trade, making a perfectly trustless system!

I. DECENTRALIZED BACKING, BANKING, SCALING AND THE LIGHTNING NETWORK

Even without an exchange, BitBay (and all coins that are derived from it) opens the door for "decentralized backing". Similar to how "Tether" promised to buy back coins at $1 per coin, BitBay users can do the same. Individual investors can back the coin themselves with fiat. In this case, the BitBay community and miners would choose an algorithm to deflate the available supply down to the current reserves. Their buyback policy might be pegged to the dollar or Bitcoin. Those reserves would gradually increase as the backers collect fees. It is true that a backer can withdraw financial support for his buyback policy, however this does not necessarily crash the coin. This is because, a new backer (whom is invested in the coin) can quickly take the place of the missing one. The backer would only need to ask the community to deflate to their comfort level. For example, John decides to back BitBay to the dollar with his own personal funds. John sees the advantage, since he holds BitBay, and can collect small fees for each transaction. He puts up 100,000 dollars of his own money. He backs the coin (1:1 at a dollar) and the community agrees to accept an algorithm to deflate to roughly 100,000 coins of liquidity. Thus, it would be impossible to crash the coin, since John has enough funds to buy back all of the available liquidity. This is his self-imposed risk. Let us say John decides to stop backing the coin. The holders of BitBay want to continue maintaining the price at $1, but John is no longer available to extend this guarantee. So they find a replacement, Jane, who only has 30,000 dollars. They decrease the amount of liquid coins to match her supply of dollars. Jane gradually collects trading fees, because Bitcoin holders want a safe haven for turbulent volatility swings. After a year of collecting fees, perhaps Jane has doubled her money. She then increases the available number of coins from 30,000 to 60,000 (and additionally the amount she has in reserve). Jane can also "roll" her peg price upwards to $1.05 per coin if she owns the extra reserves to honor the higher price. She might do this to increase the buy volume through attracting long term investors. This is, however, a hard peg and it is subject to finding a replacement backer if one pulls out. The idea is that there would eventually be multiple backers in a pool, and the backing would effectively be "decentralized".

Another hot topic for Bitcoin is the "Lightning Network"[12]. A problem that Bitcoin and other coins have faced is a lack of scalability in their current incarnations. They would not be able to handle the volume of VISA overnight. Also, it is difficult to ensure that the coins will maintain their "decentralized" nature, as confirmation of transactions will eventually move to central servers. Many of these issues involve individual computers failing to support the bandwidth of high volume transactions. Additionally, they cannot support the growing size of the ledger history. This ultimately forces them to prune the history, which in turn, cuts out valuable details and creates potential security flaws. The lightning network "daisy-chains" accounts together, using a beautiful mix of replaceable transactions and carefully timed commitments. Accounts hold these bi-directional micro trading channels (where IOUs are carefully constructed), which are replaced using many different lock times and throwaway keys. This works perfectly fine if users daisy chain peer-to-peer with wealthy holders. These large players can front the Bitcoins' routing debt for users who are connected to them. This method does cause some centralization, as users gravitate toward the ultra-wealthy for extra liquidity and speed.

A large network like this requires a significant number of reliable nodes among large holders of the coin. However, it is unclear if that would be a realistic outcome for every coin listed, as it would be difficult to accumulate a significant number of large holders (with fast servers that can route the debt). The Bitcoin community, in its original design, proposed a central hub which holds everyone's daisy chained IOUs in a large database of joint accounts. If the server loses its keys, there are many fail safes in place with the lock times and special hashes, and either user can withdraw what they are owed. Although, if the central hub does not put collateral in to route the debt, the system is no longer secure. There are many attacks, such as the central hub's ability to default on their own IOUs through collusion or hackers. While BitBay has no plans of developing a lightning network, the fact that its decentralized peg has a frozen asset class is a massive advantage to these lightning networks. This advantage occurs because a central hub can route payments by putting up their frozen coins as collateral. ***A company can simply invest in the frozen assets, purchase as much as they can, and use them to create this network.*** This gives the company increased staking power, increased voting power, fees from the peer to peer payments, and lending advantages. A central hub in this position enjoys nearly unlimited advantages. The users also benefit because transactions are instant, peer-to-peer, and with low risk of loss.

However, if the central hub is able to acquire too much of the stake power, the integrity of the network might be compromised. With this in mind, multiple hubs are still preferred. The lightning network is peer-to-peer, which keeps cryptocurrency decentralized. It also opens the door for decentralized exchange through micro trading, as there is fewer transactions

reported to the blockchain. Instead, a trusted IOU system is used, which allows transactions to be trusted on a peer-to-peer level.The BitBay network does not need lightning network to scale. It is a more simplistic approach to share the stake reward among other nodes, and allow the blocks to be dynamic in size. The network then merely prunes old, spent transactions. Although pruning does prevent a new user from knowing all transactions, we feel that over time very old transactions become less relevant. There are arguments that pruning is slightly less secure, however a reputation system for the staking nodes can strengthen this system. In the meantime, BitBay enjoys the same transaction speed of Bitcoin, and there is no need to worry about scaling beyond that any time soon.

Summary

This entire system has been proposed. The techniques listed in this paper cover almost everything a cryptocurrency needs to be a viable decentralized payment system. All completely free of third parties and volatility. ***Moreover, this system appeals to investors because they can see their investment grow in value due to "dynamic peg".*** This brings a whole new, fun, and exciting world to the crypto trader, like never seen before. The rolling peg allows for assets similar to bonds and loans, backing, improved lightning networks, voting, futures, multiple various asset classes, and nearly unlimited economic options - all with an enforcement layer and completely "trustless".

The key to successfully solving many of today's problems is through the force of technology. Many cryptocurrency applications are proposed as secure, when in reality they are not. When closely examined, it becomes apparent that a third-party weakness or potential for gaming the system exists in some way, shape or form. This includes many "Smart Contract" based systems, which lack enforcement layers. ***Typically, the best place to start looking for a solution to the problem is at the root.*** In this case, we looked at "supply and demand" and found that by controlling supply, we can solve the problem of volatility in a decentralized way - a way that does not rely on third-parties. The consequences of the frozen and liquid asset classes are truly mind blowing. It would be too much to list in a single paper. Traditional currencies for practically all of modern history have been known to be shrouded in secrecy, and abusive of their monopolistic and hierarchical power. This has caused a loss of freedom and happiness for all of humanity, in ways that most people are probably not even consciously aware of. On some level, this perceived collective loss of human rights is truly felt by many of us. Finally, this loss can be impacted, as smaller currencies can now find their niche in society. They can now offer benefits to the consumer which are similar to (and more lucrative) than government backed currencies. It offers an alternative with great advantages that traditional currencies are incapable of producing.

The decentralized peg is perfect for small projects, indigenous tribes, impoverished nations, small companies, and abandoned currencies. It obliterates volatility - allowing for healthy and stable growth and directly responding to the needs of the people invested in it. In fact, it could be the one feature that Bitcoin itself needs to break into the mainstream. It allows the stability and strength of larger currencies with the technological benefits of cryptography. That is not to say this system will cause revolutionary change overnight. Either way, we believe this paper will inspire further research into the subject.

We will finally have a tool that can truly create more freedom for those that need it.

References

[1] https://bitcoin.org/

[2] https://bitbay.market /

[3] https://bitshares.org/technology/price-stable-cryptocurrencies/

[4] https://www.nubits.com/about/white-paper/

[5] https://tether.to/

[6] https://basis.io/

[7] https://bithalo.org/

[8] https://bitshares.openledger.info/#/create-account

[9] https://www.blocknet.co/

[10] https://komodoplatform.com/

[11] https://en.bitcoin.it/wiki/Atomic_cross-chain_trading

[12] https://lightning.network/


BitBay