



**Nu**

Jordan Lee  
September 23, 2014  
[nubits.com](http://nubits.com)

## **Contents**

1. [The Problems Being Solved by Nu](#)
2. [Overview](#)
3. [Voting](#)
4. [Parking](#)
5. [Unparking](#)
6. [Park Rates](#)
7. [Custodial Grants](#)
8. [Custodians](#)
9. [Buy Side Liquidity](#)
10. [Liquidity Pool Tracking](#)
11. [Trading Bot](#)
12. [Use Case for Dual Side / Liquidity Provider Custodian](#)
13. [Use Case for Sell Side / Specific Purpose Custodian](#)
14. [When to Lower and Raise Park Rates](#)
15. [Additional Currencies](#)
16. [If USD Becomes Unstable](#)
17. [When Nu Is Obsolete](#)

## The Problems Being Solved by Nu

---

Bitcoin was a phenomenal innovation. For the first time individuals could hold an asset without counterparty risk and transfer it to anyone else on the network quickly and privately, if desired. It had some flaws, which include a high cost of maintaining the network and the disassociation of control of the network (given to miners) and ownership of its assets (Bitcoin holders). Peercoin improved upon Bitcoin by dramatically reducing the cost of network maintenance and giving control of the network to the owners of network assets.

Both of these networks contain a critical flaw which Nu resolves. These networks permit the purchase of scarce units used in the networks which function much like shares. If the value of the network rises, the value of these "shares" rise. This dynamic has been critical to the success of these networks as it allows anyone to purchase a stake and benefit from promoting the network. These networks have simultaneously been promoted as currencies but have not functioned well as such. Currencies must have a stable value to be effective, while Peercoin and Bitcoin have exhibited exceptional volatility. Many argue volatility will end with the high liquidity that will accompany widespread adoption. While volatility will decrease with greater adoption, it is unlikely volatility will ever be less than occurs with large cap stocks such as Google or Microsoft. This is still an unacceptable level of volatility for a currency. Let us suppose I am wrong and that volatility will be eliminated in these networks. In that case they would serve well as currencies but poorly as shares, because they would not appreciate, nor give dividends. This would likely cause a selloff of these "shares", thereby introducing volatility once again.

The critical flaw is that Peercoin and Bitcoin use the same fungible unit for share and currency functions. Shares must have the capacity to appreciate and reflect changes in the perceived value of the network while currency must remain stable regardless to be effective. It is impossible to accommodate these diverse pricing needs in a single unit.

## Overview

---

The solution is a network with two types of units which are not fungible: shares and currency. Shares represent ownership of the network and their quantity should not change to accommodate changes in the level of demand for them. If demand increases, the price should rise proportionately. They should also provide dividends from network revenues. The supply of currency, however, should dynamically adjust up and down in response to changes in the level of demand for the currency so that the price is always stable. A network with these characteristics can be most easily implemented as a fork or extension of Peershares.

Just like any Peershares implementation, Nu is controlled by shareholders who own NuShares and mint blocks with them using proof of stake. Unlike other Peershares networks, Nu also verifies and transmits transactions of currency units called NuBits.

Nu permits holders of NuShares to manage the quantity of NuBits without dependence on any off blockchain mechanisms. Additional NuBits can be created when shareholders vote to do so and will be placed in the custody of a recipient chosen by shareholders called a custodian. There can be as many custodians as the shareholders elect to have and they can be changed at the whim of shareholders. Custodians may adopt a pseudonym if they choose to do so. Because custodians can be pseudonymous, many in number and abandoned at the will of shareholders, third parties cannot gain control of the network by leveling threats against custodians. Being a custodian doesn't impart any control over the network, only over a finite quantity of network revenues. NuBits created and sold are network revenues that can be used by custodians for Nu operating expenses and dividends. Dividends will be paid only in Peercoins, which means custodians must purchase Peercoins in preparation to distributing them. Therefore, demand for Peercoins will increase proportionate to the amount of dividends paid by Nu.

Conversely, the system can reduce the available supply of NuBits through a separate mechanism called parking, where holders of NuBits volunteer to take their currency out of circulation for a period of time in exchange for a monetary incentive. Shareholders can dynamically adjust the reward offered for parked NuBits (which varies by the duration) by voting for whatever yield curve they think is appropriate when they mint a block. When organic NuBit demand is in decline, shareholders will create synthetic demand for them by offering reward for parked NuBits. Some entities that would not be interested in NuBits as a currency will purchase them simply to park them for the reward, thereby compensating for the decline in organic demand. Shareholders will raise the reward offered to whatever level is required so that organic and synthetic demand combined will not decline. As organic demand increases once again, the offered park rates will be reduced until organic demand reaches its previous peak, at which time no reward will be offered for parking.

The NuBit price will be suppressed to one USD from custodians maintaining huge sell walls at one USD and supported at one USD by shareholders offering a reward on parked NuBits to create synthetic demand when there are market signals that demand is in decline. While early in the network's life minute to minute NuBit demand could decline suddenly and cause the price to momentarily drop below one USD, shareholders will quickly push the price back into the custodial one USD sell wall by increasing park rates. This will punish anyone who sold below one USD and reward anyone who purchased below one USD. Soon it will become clear that selling NuBits below one USD is always a losing trade and buying them below one USD is always a winning trade, which will prevent its decline. A special type of custodian can be compensated to provide deep liquidity to sellers of NuBits at one USD, preventing the price from falling even in the case of a sudden drop in demand that is too quick to allow for park rates to be raised.

## Voting

---

Shareholders have the privilege of voting when they mint a block. The Nu client and user interface will permit a shareholder to configure their votes manually. Optimal voting values

may change quickly so shareholders may elect to use an automated data feed from a source they have decided to trust with their vote. A variety of entities will provide data feeds of their recommended votes, providing diversity and competition in this automated voting advice. When they mint a block, their votes (manual or from data feed) will be placed in the coin stake transaction on the blockchain. Each of three types of votes should be user configurable as manual or from data feed individually. A user may choose a data feed for park rates votes but manually enter other votes, for instance. Votes are weighted by the share days (known as coin days in Peercoin) consumed minting the block and in the case of motions and custodial grants, must also receive votes in 5001 out of 10000 blocks in order to pass. More will be said about vote counting in the [Park Rates](#) and [Custodial Grant](#) sections of this document.

Shareholders may place three different types of votes:

1. **Custodial grants of NuBits:** To vote to grant a custodian NuBits, a shareholder will simply place the *currency type* (only NuBit for now but later we may add currency units pegged to EUR, GBP, etc), *public address* and *grant amount* in their user interface, which will accept multiple entries at once. A shareholder could simultaneously vote to grant one custodian a million NuBits who is promising dividends to shareholders while voting to grant another custodian 200,000 NuBits for core software development. Custodians are identified only by NuBit address in the protocol.

When a shareholder mints a block, his vote will be placed in the PoS coin stake transaction as a series of currency type values, RIPEMD-160 values (a version of a public address that is 20 bytes in length) and `uint64` values indicating the number of satoshis to grant. A pay-to-script-hash address may be substituted for the RIPEMD-160 if multisignature transactions are being used.

To understand how grants to custodians will be created and how votes are counted, read the [Custodial Grant](#) section.

2. **Yield curve for parked NuBits:** Rather than voting for a single park rate, shareholders should cast separate park rate votes for a variety of durations. Durations to be voted on will begin with 1 block. Other durations to be voted will be successive doublings: 2 blocks, 4 blocks, 8 blocks, etc. An `uint8` will be used to represent the block count. A value of 0 will mean 1 block, 1 will mean 2 blocks, 2 will mean 4 blocks, 3 will mean 8 blocks, etc. The park rate vote will be expressed in an `uint32` as the number of hundredths of a satoshi to pay per NuBit parked. Therefore, the park rate vote to be placed in the coin stake transaction will come from a two dimensional array where the first dimension consists of an `uint8` and an `uint32`. Shareholders can effectively set a minimum and maximum parking interval by the block counts that they omit from their vote. Consider this sample park rate vote which uses decimal notation:

<b>Duration Code</b>	<b>Park Rate (satoshis for entire duration per NuBit)</b>
13	3
14	6
15	13
16	30
17	65
18	150

A Duration value of *12* will be interpreted as 4096 blocks, *13* as 8192 blocks, *14* as 16384 blocks etc. with *19* meaning 524288 blocks. This shareholder is voting for the premium to be zero for 4096 (duration code *12*) and fewer blocks. Likewise, the shareholder is voting for a premium of zero for 524288 (duration code *19*) and greater blocks. A user can park NuBits for any duration for which the premium is not zero, such as 6000 blocks or 14000 blocks. In these cases a weighted average of the adjacent premium votes will be used to calculate the exact premium.

The client user interface will feature an interface that allows the user to add manually configured park rate votes for each doubling interval of time. Duration code will be converted to a block count and then to an interval of time (presented as x days for instance). Instead of entering a number of satoshis, the user will be asked for an annual park rate which can be converted to hundredths of satoshis by the client.

<b>Duration Code</b>	<b>Number of Blocks</b>	<b>Approximate Time</b>
0	1	1 minute
1	2	2 minutes
2	4	4 minutes
3	8	8 minutes
4	16	16 minutes
5	32	32 minutes

<b>Duration Code</b>	<b>Number of Blocks</b>	<b>Approximate Time</b>
6	64	1 hour
7	128	2.125 hours
8	256	4.250 hours
9	512	8.500 hours
10	1024	17 hours
11	2048	1 day, 10 hours
12	4096	2 days, 20 hours
13	8192	5 days, 16.5 hours
14	16384	11 days, 9 hours
15	32768	22 days, 18 hours
16	65536	45 days, 12 hours
17	131072	91 days
18	262144	182 days
19	524288	1 year
20	1048576	2 years
...	...	...

The park rate vote interface should feature a button to add a shorter duration and another to add a longer duration. When either one of these buttons is clicked the first time, the 131k block duration appears in a grid. Additional rows representing adjacent durations appear in the grid appear above or below depending on which of the two buttons is clicked. All other durations will implicitly be zero when no value is entered. The amount of time each block count will consist of should be calculated and put that time value in the UI and interpret as block counts in the code. Next to each duration

is a text box to enter the park rate the user would like for that particular duration. Bear in mind that while we will have 1 minute block spacing initially, we expect this to change so try to write the code so that it doesn't have to be changed if the PoS block target spacing changes later.

3. **Motions:** Because motions can take any form whatsoever, they cannot be enforced by protocol. No attempt is made to do so. Rather, the protocol can only record votes for motions and which can then be examined to verify whether a particular motion has been passed. It will be left to shareholders to ensure that a motion which is passed is implemented.

A motion may be proposed by literally anyone. Though it will likely be composed of text, it can take the form of any electronic document or file. A RIPEMD-160 hash of the motion should be constructed (40 hexadecimal chars in length) and distributed with the motion. To vote for a motion, the 40 hexadecimal characters should be entered in the user interface or acquired from a data feed which will require 20 bytes of space when placed in the PoS coin stake transaction on the blockchain.

A motion which receives a vote in 5001 out of any 10000 consecutive blocks and the majority of share days (coin days) in those 10000 consecutive blocks is considered passed. A blockchain explorer (which we will fund the development of before initial release) will find and display links to motions passed.

The PoS coin stake transaction will be limited to 1000 bytes in length by the protocol, although typically it will be around 150 bytes. In addition to votes and coin stake it will contain the actual park rates that apply to parked transactions in the block (in addition to the minter's votes).

## Parking

---

Parking is when a holder of NuBits volunteers to have all the outputs associated with a particular address placed in a parking transaction in exchange for a promise from the network that when the funds are unparked they will have a larger amount of NuBits.

When NuBits are parked, the user will select a duration that they will be parked for. The protocol permits NuBits to be parked for any number of blocks. They will be unspendable for the duration selected via an [OP\\_RETURN](#) transaction. This has the effect of reducing the monetary base for the duration that funds are parked. Said another way, it increases demand for NuBits as some entities will desire to have them for the sole purpose of parking them. This mechanism can be used to increase demand for NuBits to ensure the price does not fall below one USD when organic demand is in decline.

On the blockchain, parking transactions will be marked as such and include the duration of the parking as a number of blocks. The number of blocks an output is parked can be any non-zero positive integer.

## Unparking

---

After the duration passes that was chosen when NuBits were parked, the client will automatically create a coinbase transaction that creates the parked amount plus the reward at the user-configurable address using the information in the `OP_RETURN` transaction they were parked in. The client will track all parked outputs contained in the local wallet including how many blocks they have until maturity. This will be updated with each new block received or created. When a parked output in the wallet passes its parking duration, this coinbase transaction will be generated. Clients that receive this coinbase transaction will compare with blockchain votes and balances to ensure it corresponds to a park rate reward owed but not yet paid. In Bitcoin and Peercoin, coinbase transactions are a mechanism used to create new coins that are a proof-of-work block reward. We adapt coinbase transactions to create NuBits for park rate rewards.

## Park Rates

---

A mechanism used to provide incentive to NuBit holders to park their funds. When NuBit demand is growing beyond all previous peaks in demand, it is anticipated that shareholders will offer no reward for parked NuBits. When demand for NuBits is in decline, shareholders will vote to offer reward for parked NuBits. Shareholders will offer sufficient reward to create enough demand for NuBits to support the price at one USD. Shareholders don't need to offer a single park rate for all durations. They can set different park rates for different durations, creating a dynamic yield curve. The way shareholders will vote for this is discussed in the Voting section.

## Custodial Grants

---

When shareholders vote to grant a custodian a certain amount of currency, it is called a *custodial grant*. These are created using a coinbase transaction to create new currency at the custodial address voted for. When clients check whether a custodial grant sent to them is valid, they will check the expansion coinbase to ensure it matches the custodial address for which they will have already received 5,000 votes for and that it matches the custodial address for which the majority of coin days out of the last 10,000 blocks (including the transmitted expansion block) have voted for. Custodial grants are evaluated as a currency type, address and quantity of currency. If any one of these three elements vary, it is considered a different vote.

NuBits will only be created at a particular address once. If shareholders wish to make multiple grants to a single custodian the custodian must use a different address for each grant.



## Custodians

---

Custodians will behave somewhat like politicians running for office. They will say:

Vote to give me  $x$  quantity of NuBits at address *Bd8sd18dneRmedkSmes2omkd4gGu*. When I receive them I will do  $x$ ,  $y$  and  $z$  with them.

Some custodians will use their NuBit grants to fund development of the core protocol. Others will use the funds to fund other software development that will benefit Nu. It is possible a custodian could use a grant to start a NuBit lending institution. Another custodian may be compensated to provide deep buy side NuBit liquidity. Most importantly, some custodians will use revenues to buy Peercoins and distribute them as dividends. All custodians will sell the NuBits granted to them at the price of one USD. It is likely custodians will demonstrate they hold a substantial quantity of NuShares so that shareholders know custodians' interests align with their own. It is likely that at some point a custodian will disappoint shareholders, but the damage they can cause is limited to the value of the grant they have already received.

A custodian can be a collection of individuals or entities by utilizing multisignature transactions on custodial grant addresses.

Custodians receive grants from shareholder voting. Refer to the [Voting](#) section for more details.

## Buy Side Liquidity

---

It is important that deep NuBit liquidity exist at one USD, on both the sell side and buy side. Custodians provide the sell side liquidity as they place NuBits granted to them for sale. This leaves the buy side, the side where liquidity supports the price at one USD. While the market will naturally have a buy side, it may be thin, particularly when demand for NuBits is in decline. A pool of liquidity is needed to stabilize the price, to assure holders of NuBits that they can exit a large position at will and it is also needed to provide the signal to raise park rates before the price has a chance to drop below one USD. For instance, if there is total pool of buy side liquidity of 100 million dollars, and demand for NuBits declines, the buy side liquidity will begin to be consumed as holders of NuBits sell. Shareholders need to gain awareness of this pool decreasing, as it will be their signal to raise park rates to increase NuBit demand and replenish the buy side liquidity pool. It is important that shareholders increase park rates enough before this buy side liquidity is consumed, because if that occurred, the price could drop below one USD briefly.

## Liquidity Pool Tracking

---

To provide maximum decentralization, each person or organization providing liquidity should be a custodian. The NuBit address they receive a grant at will be used to sign messages. A new RPC called `liquidityinfo` will be implemented. The purpose of calling the method is to have the Nu client propagate the order information through the entire network of Nu clients. The goal is to have all Nu clients be completely aware of the size of the buy and sell liquidity pool at all times in a way that doesn't rely on any centralized infrastructure. This info must be accurately propagated so that shareholders know whether new custodial grants are needed (to expand the sell side pool) or whether park rates must be raised (to expand the buy side pool). Similarly, if park rates are higher than 0 the sell side pool can also be expanded by lowering park rates. As it begins to become unbalanced, shareholders must know so they can adjust park rates or make custodial grants. If either liquidity pool is completely depleted, the peg will be lost until the pool is replenished. Temporarily losing peg will harm the value of the network. While in the early days of the network this may occur, we expect to be able to make the peg extremely reliable by increasing the liquidity pool size and shortening shareholder reaction times to changes in the pool sizes.

A single supporting RPC called will be employed whose signature will be:

```
Value liquidityinfo(const Array& params, bool fHelp)
```

The parameter array will have the following four elements converted to the following types: `char currency`, `int64 buyAmount`, `int64 sellAmount`, `string grantAddress`.

The second parameter is needed because Nu is designed to support multiple currencies (for example units pegged to EUR, CAD etc). For now it just supports NuBits, pegged to USD. The `char` value for this is 'B'.

The `buyAmount` and `sellAmount` parameters are the number of satoshis of the currency. 10,000 satoshis = 1 NuBit.

The info passed into `liquidityinfo` should be transmitted to all connected peers, similar to transactions. Liquidity info should never be recorded in the blockchain, but should be handled in a manner similar to the memory pool that manages unconfirmed transactions in some ways. When a client is started up, it should receive the complete liquidity pool information from a connected peer. Thereafter, it will listen for messages containing liquidity info that are signed by a custodial address (an address that has been the subject of a custodial grant within the last 260,000 blocks (about six months)). The memory pool dedicated to liquidity will maintain a record for each unique combination of `grantAddress` and `currency`. Here are sample contents of the memory pool:

Grant Address	Currency	Sell Amount	Buy Amount
BQFqqMUD55ZV3PJEJZtaKCsQmjLT6JkivJ	B	100000000	0
1F5y5E5FMc5YzdJtB9hLaUe43GDxEKXENJJ	B	0	43903000000
PFWU1xDF1GaXFRUbC31gEXGHWyzAaweJcX	B	23900000	68099980000

Whenever the trading bot placing an order, cancels an order, or an order is filled, it will make a call to the `liquidityinfo` RPC, sending out updated information about its specific portion of the liquidity pool to all peers. When a client receives a valid and signed liquidity info message, it examines the liquidity memory pool to see if there is a record with a matching Grant Address and Currency. If there is, the new message will replace that record. If the *Grant Address* and *Currency* does not yet exist in the liquidity memory pool, it should be added.

The aggregate liquidity pool info should be displayed in the client user interface. It might look like this:

Currency	Sell Total	Buy Total
NuBits	18,098,400	15,909,432

Shareholders can use this info to make decisions about custodial grants and park rates. Most shareholders will prefer these decisions be automated, and we have talked about doing this by allowing them to subscribe to data feeds. Data feed providers will use this information determine the contents of their data feed.

An additional RPC called `getliquidityinfo` should have a currency parameter and return the *Grant Address*, *Sell Amount* and *Buy Amount*, similar to the table depicted above.

## Trading Bot

---

Only custodians will use the trading bot (NuBot) and relay liquidity data to other Nu clients. Within this subsystem there are two types of custodians: sell side and dual side custodians. Dual side custodians are custodians whose specific function is to provide liquidity for compensation, and they will initially only provide buy side price support. Once their buy order for NBT is partially filled, the bot should then create a sell order for that NBT. In the case of sell side custodians, the liquidity they provide is secondary to another goal such as funding core development, marketing NBT or distributing Peercoin dividends.

They want to spend the proceeds of their NBT, so under no circumstance will they provide buy side liquidity. Such custodians may also directly spend NuBits to fund operations, in which case they would not use NuBot at all. The trading bot must permit a user to indicate they are either a sell side or dual side custodian. This will affect the trading bot's behavior as detailed in the use cases below.

## Use Case For Dual Side / Liquidity Provider Custodian

---

First someone who wishes to fulfill this role must seek shareholder approval via the custodial grant mechanism. Say a particular liquidity provider or LP custodian has 10 million USD he wishes to use to provide NuBit liquidity. He would expect compensation for lost opportunity cost (he could otherwise invest those funds in rental property, stocks or bonds) and for the risk of loss via an exchange default, such as we have seen with Mt. Gox and others. While the market will continually reprice this, let's say in our case the prospective LP custodian decides a 5% return every six months is fair compensation for lost opportunity cost and risk of exchange default. So, he promises shareholders to provide 10 million USD/NBT worth of liquidity for one year in exchange for 500,000 NBT. Shareholders approve this using the grant mechanism and he is granted 500,000 NBT. Now he must provide 10 million in liquidity constantly over the next year. Alternatively, once shareholders have sufficient credibility, LPCs may be compensated after providing liquidity services. Liquidity may be provided through a single exchange or multiple exchanges. Let's say he does this with a single exchange. He opens an account with the exchange, then deposits \$10 million worth of BTC. He exchanges the BTC for USD.

Now he is ready to make use of the trading bot. An appropriate exchange will expose a trading API and NuBot must implement the API for that specific exchange. Each implementation of a specific exchange API should implement an interface called IExchange to standardize the way the trading bot interacts with these diverse exchange APIs. Doing this will allow the LP custodian to enter authentication information into the trading bot for his exchange account. He will then use the user interface in our trading bot to place a buy order for 10,000,000 NBT on the exchange. The price will not be exactly one USD. It will be one USD minus the exchange transaction fee. If the exchange charges a 0.2% transaction fee, he will place a buy order for 10,000,000 NBT at a price of 0.998 USD. Let us suppose his order is partially filled in the amount of 1,000,000 NBT. Now his exchange account will contain 9,000,000 USD used to fund an order for 9,000,000 NBT. There will also be 1,000,000 NBT in the account. The trading bot should automatically and immediately place these 1,000,000 NBT for sale at a price of 1.002 USD (one USD + a 0.2% transaction fee). If this order fills, then the bot should use the USD proceeds to immediately place a buy order for NBT at 0.998 USD. All funds should be continually on order and the LP custodian's funds should not be depleted by transaction fees.

When an order is placed, canceled or filled (even partially), the `liquidityinfo` RPC must be called on the Nu client, which will be detailed below.

## Use Case For Sell Side / Specific Purpose Custodian

---

In some cases custodians will spend the NuBits directly and not use the trading bot at all. For instance, if core developers accept NuBits as compensation then Jordan Lee will simply distribute NuBits granted to him directly without the need for any exchange.

Let's examine the case where a 10,000,000 NBT custodial grant is given for the purpose of distributing a shareholder dividend in Peercoin. Such a custodian will deposit 10,000,000 NBT in a single or multiple exchanges. In our use case we will use a single exchange. Once the NBT deposit is credited, the custodian will start the trading bot, indicate they are a sell side custodian and indicate that orders should be created, although nothing specific about the order should be entered by the user. The trading bot should offer the entire balance of NBT for sale using the formula of one USD + transaction fee + one pricing increment. Let's say our exchange has a transaction fee of 0.2%. Some exchanges allow the fee to be discovered through their API, while others do not. If the fee can be found through the API, it should. If not, the user should be asked to specify the transaction fee. Let's say this exchange supports 4 decimal places in its order book on the NBT/USD pair. Using our formula above, the trading bot would place a sell order for 10,000,000 NBT at a price of 1.0021. The reason it should be 1.0021 instead of 1.002 is that we want dual side sell orders to be executed first, so their funds can be returned to providing buy side liquidity.

Each time an order is placed, cancelled or filled (even partially), the `liquidityinfo` Nu client RPC method should be called.

Details about this method can be found in the Liquidity Pool Tracking section. Calling `liquidityinfo` will have the effect of transmitting the size of the buy and sell liquidity pool the local trading bot is managing to all known Nu peers.

## When to Lower and Raise Park Rates

---

I have described the mechanisms by which pools of buy and sell side liquidity will be created and maintained. When the sell side pool is decreasing relative to the buy side pool, park rates should be lowered to decrease demand. When the buy side pool is decreasing relative to the sell side park rates should be raised to increase demand. How much to raise or lower park rates cannot be determined with perfect accuracy, but it can be constantly readjusted to maintain the liquidity pools within a certain range of proportions of one another. History will also provide useful information about how much to adjust park rates for a given change in the liquidity pools. When the sell side pool is decreasing and park rates are already zero, this is the ideal state of the network as it necessitates the creation of additional NuBits (to enable continued suppression of the price) giving revenue to the network.

## Additional Currencies

---

After the initial release of Nu, it is possible that additional currencies will be added. Nu is designed to permit this with very few code changes. For instance, a new currency pegged to one euro per unit could be added. A currency where the peg rate is altered to adjust for inflation or deflation is also possible, creating a stable currency that retains purchasing power over the long term. The protocol enforces that each currency is not fungible with any other, meaning inputs and outputs must be of the same type and cannot be mixed. The same mechanisms used with NuBits (custodial grants and parking) will be used to stabilize the price of other currencies. Voting for custodial grants and parking yield curves will be completely independent of NuBits. Shareholders can simultaneously place votes for as many currencies as they like.

## If USD Becomes Unstable

---

Regarding the scenario of USD becoming worth nothing or otherwise becoming unstable and unsuitable as a target of a peg: While I do not think this is likely in the near or medium term, it is possible. I would like to note that I hope this will not occur while USD is used in its present role as the world's reserve currency, as the global instability it would cause would be devastating to humanity. NuBits will not cause USD hyperinflation. If successful it will induce the Federal Reserve to be more efficient and compete better, which will benefit nearly everyone and make our world much more peaceful. In the edge case that USD hyperinflation does occur, shareholders may at that time pass a motion to peg NuBits to another currency, commodity or basket of goods.

## When Nu Is Obsolete

---

While Bitcoin is a first generation cryptoasset, Peercoin was the beginning of the second generation defined by proof of stake. Nu heralds a third generation of cryptoassets featuring stable value managed by shareholders. Will there be a fourth generation? Likely. I do not yet know what will be its defining characteristics, when it will arrive or whether it will make Nu obsolete. Nu is more adaptable than Bitcoin or Peercoin with its voting mechanisms and shareholders are likely to devote considerable revenues to updating it as well as research and development. While I believe the system can likely be sustained for a very long time, it would be foolish to believe it could last forever, as in century after century. Someday it will be replaced by a superior system based on technology not foreseeable today.

When this occurs, NuBit demand will decline permanently. The end of the currency will be marked by park rates rising to unprecedented highs and then going still higher until the vast majority of NuBits are parked. When market participants reach a unanimous consensus that NuBits are worthless, then they will suddenly drop to zero value from one

USD. As long as a small group of speculators believe there is even a small chance NuBit demand will reach a new all time high the price will remain one USD. As the currency shows signs of stress and serious decline in levels of use NuBits will pass from ordinary businesses and people to speculators willing to take large risks for large rewards. Ownership of NuBits will centralize somewhat as the currency shows signs of stress. Failure of the currency is not synonymous with failure of the network. If there are other currencies offered by Nu, they will continue to be unaffected.

In the space between now and obsolescence, there is much that Nu can do to benefit shareholders and its users.

Finally, it should be noted that Nu is experimental software at this point. It may not work as intended. However, shareholders will be tenacious in repairing any defects that are found.

---