Collypto Whitepaper

Matthew M^{c}Knight
John Bulwer

Table of Contents

Abstract

Collypto is the world’s first functional flatcoin. A flatcoin is a cryptocurrency that is analogous to a stablecoin, except that its value is “pegged” to purchasing power, rather than a secondary currency, security, or commodity. “Collypto” is derived from the words “collateralized” and “cryptocurrency”. Collypto Credits are 100% collateralized with publicly traded, high-volume, intrinsic value assets in the categories of real estate, materials, food, precious metals, and energy. Collypto embodies the spirit of the gold standard in that the currency is represented by a securely held intrinsic value store. It transcends the gold standard by employing the use of a combination of societally relevant assets to provide a more meaningful representation of what is valued in an economy, based on consumer spending behavior. We have also introduced features such as verified transactions, fraud prevention measures, and recourse in instances of theft or fraud, to provide consumer protections that do not currently exist in the cryptocurrency industry.

Introduction

“The most exciting phrase to hear in science, the one that heralds new discoveries, is not, ‘Eureka! I've found it,’ but, ‘That's funny…’”
-Isaac Asimov

We didn’t start this project with the intention to create a cryptocurrency. Collypto (named later) was originally designed as an inflation-proof store of intrinsic value. To put it in perspective, we spent almost two years designing, creating, improving, and backtesting the Collypto Algorithm to build and maintain the dynamic allocations of our asset pool, but it took us less than six months to implement, verify, and deploy our blockchain contracts. This massive time disparity is a testament to the robust native functionality of Ethereum as a customizable solution to the Byzantine Generals Problem and maintaining a decentralized ledger, but more on that later. First, we need to address the proverbial elephant in the room.

Traditional cryptocurrency has failed

Promises of life-changing money have been replaced with horrific accounts of billions of dollars in losses due to accidents, theft, schemes, and dishonest actors. Bitcoin, Ethereum, and other cryptographically secure blockchains were designed to operate as peer-to-peer payment systems, but upwards of 95% of digital asset trades are executed through centralized exchanges (Cryptopedia, 2022). After over a decade of market exposure, most people could never imagine paying for goods and services with bitcoin, ether, or any existing cryptocurrency.

Where did it all go wrong?

To answer this question, we must first understand that it is fundamentally impossible to create a functional currency using the native “transaction” token of any blockchain. The transaction token of a blockchain only represents the intrinsic value of conducting transactions on that blockchain and the speculative value of the token asserted by the market.

When people attempt to price something that has no intrinsic value *to them*, the only value that asset can have becomes the value they can obtain by selling it at a higher price. This is the literal embodiment of the greater fool theory. It is why the price of traditional cryptocurrency[1] has always functioned as a cyclical pyramid scheme and can only ever attract people who would want to use it for that purpose. These people aren't evil, and they aren't consciously creating this system. A cyclical pyramid scheme is the natural result of a population continuously attempting to price a purported asset that is intrinsically worthless.

Traditional cryptocurrency advocates often claim that a token's value is a product of its scarcity, but scarcity alone does not dictate value. "Scarcity" is a noun used to express a description of supply when pricing something in relation to its demand. A currency that represents the wealth of humanity cannot have an immutable supply cap because the wealth of humanity is continuously growing. Citing the relationship between the scarcity and value of dollars in an attempt to create a meaningful model for valuing cryptocurrencies is akin to a financial cargo cult[2].

The very notion of a decentralized currency is itself an oxymoron. A currency is a standard, and a standard must be centralized. Fiat currencies have effectively achieved a decentralized value consensus in a manner analogous to the aggregated acceptance of legally mandated traffic laws. $50 has the same purchasing power at Taco Bell as it does at Home Depot, even if the goods and services themselves are different because we have all developed a collective sense of the value of the dollar. Transaction tokens can be minted using a decentralized system, but those tokens will never be able to function as a meaningful currency. The next generation of currency cannot evolve from existing fiat systems, but rather, it must be deliberately engineered to track purchasing power without requiring a pre-existing consensus of its value.

But they made it halfway!

Satoshi's solution was an innovative and unprecedented effort to separate currency from government, and attempting to create a currency using native blockchain tokens has been a telling trial of human nature. Many of the lessons that we've used to create Collypto would not have been available without the examples of Bitcoin and its successors. The Wright brothers were pioneers in their approach to controlled, heavier-than-air flight. While their innovations revolutionized the world, we do not use Wright Fliers for commercial air travel. We have built upon their creations and have continued to adapt our solutions to meet the needs of modern society. Similarly, we must continue to build upon the foundations laid to create a functional currency with universal appeal that maintains relevance in today's world.

What was missing?

A solution to the Byzantine Generals Problem is necessary but not sufficient to solve the problem of creating a currency. Blockchain technology is an amazing innovation, but right now, it is functionally used as nothing more than a decentralized messaging service that runs on volatile transaction tokens and offers no unique value to society. Every application or financial instrument currently available on a blockchain could be replaced with a superior traditional product.

To achieve mainstream adoption, any cryptocurrency must offer users value that they could not obtain using traditional financial instruments. Such a currency would need to tokenize something that is intrinsically valuable and worth the cost of transacting on a decentralized system. This currency would need to outperform any fiat currency that has ever existed, or people would never give it a chance. It would need to preserve the purchasing power of its owners and be functionally immune to economic or inflationary circumstances. There is only one kind of currency that could ever meet all of the criteria: the mythical flatcoin that has been theorized but never successfully implemented until now.

Why we need the blockchain

Collypto was originally designed as an algorithmically managed asset pool that would function as an inflation-proof store of intrinsic value. To tokenize that store of value, we needed a way to prevent double-spending or counterfeiting. To provide indemnity to our users, we needed a way to exert force as an arbiter without resorting to physical violence. The blockchain is the only system in existence that allows us to solve both of these problems to effectively privatize currency.

By using the Collypto asset pool as a centralized collateral store and the decentralized transaction protocols of the Ethereum blockchain, Collypto effectively eliminates fraud within the system without necessitating the use of traditional enforcement mechanisms that are typically employed by governments (arrest and incarceration). The features of Collypto provide a pathway to indemnity for users without becoming an enforcement arm of any government.

If a flatcoin were a nuclear bomb, the collateral asset pool would be the payload, and the blockchain would be the high-tech delivery mechanism that can send it within a centimeter of its target. Without the delivery mechanism, the payload cannot be properly utilized, and without the payload, the delivery mechanism is ineffective. We need both the collateral and the blockchain to make an effective flatcoin. In this whitepaper, we will demonstrate how we have integrated these two systems to create the most powerful currency in human history.

What is Collypto?

Collypto is the world's first functional flatcoin. A flatcoin is a cryptocurrency analogous to a stablecoin, except that its value is "pegged" to purchasing power rather than a secondary currency, security, or commodity. Collypto embodies the spirit of the gold standard, in that the currency is represented by a securely held intrinsic value store. Collypto transcends the gold standard by employing the use of additional assets to approximate value. Using a combination of societally relevant assets provides a more meaningful representation of what is valued in an economy.

When the gold standard was in effect, one US dollar would correspond to a specific quantity of gold (e.g., 1/35th of an ounce). Likewise, one Collypto Credit corresponds to a specific allocation of real estate, steel, lumber, copper, corn, soybeans, wheat, crude oil, gold, silver, and platinum represented in the Collypto Index (described later) using futures and ETFs.

Unlike stablecoins that use arbitrary minting and burning activities or fiat currencies to maintain a price peg, the value of Collypto Credits is guaranteed by the off-blockchain assets that we hold as collateral.

Collypto maintains purchasing power, regardless of fluctuations in the market or the value of other currencies. The Collypto Algorithm tracks the Collypto Index to maintain the Collypto asset pool whose value is tokenized to create Collypto Credits. The market price of a credit will fluctuate over time, but its value will remain constant. The price of a credit directly corresponds to the underlying price of the assets held in the asset pool as collateral. Collypto effectively tracks purchasing power to counter the impact of inflation on fiat currencies. Collypto holders should note that an increase in the market price of their credits is generally the result of the diminished purchasing power (inflation) of their native fiat currencies. The associated price adjustment of Collypto Credits serves to maintain the purchasing power of their holders. This ensures that the holders' value is preserved, regardless of the state of the overall economy.

How do we define value?

In order to create a politically and economically agnostic inflation hedge that can function as the collateral of a flatcoin, we must define and track a set of asset allocations that represents the fundamentals of purchasing power itself. Such an asset pool would need to be composed entirely of securities whose prices are primarily a reflection of the intrinsic value system of humanity. Before we continue, it's important to understand that our perception of value can be defined as the sum of two contributing factors:

1. *Intrinsic value* refers to the outcome-independent base value of an asset.

2. *Speculative value* refers to the potential amount of value an asset would gain in the event of an assumed outcome.

To put this in perspective, let's say you buy a lottery ticket. For purposes of this example, the ticket costs \$2, the jackpot is \$1,000,000,000, and you have a win probability of 1 in 200,000,000.

That means that the total value of that ticket could be defined as follows:

$$V_{total}(asset) \ = \ V_{intrinsic}(asset) + V_{speculative}(asset) \ * \ P(success)$$

- $V_{total}(asset)$ is the total value of the asset (the lottery ticket) to its owner.
- $V_{intrinsic}(asset)$ is the intrinsic value of the asset. For purposes of this example, this would be the literal value of the materials that comprise the physical ticket, which is assumed to be negligible.
- $V_{speculative}(asset)$ is the value the asset would gain in the event of the desired speculative outcome (the ticket has the winning numbers).
- $P(success)$ is the probability of the desired speculative outcome.

Let's apply the equation to our lottery ticket example.

$$V_{total}(ticket) \ = \ {\sim}0 + \$1{,}000{,}000{,}000 \ * \ (1/200{,}000{,}000) \ = \$5$$

Since the calculated value of the lottery ticket is greater than the cost, it makes sense to buy. In this situation, it would technically be possible to arbitrage the entire gain if you could purchase every possible ticket. For purposes of this example, assume you are limited to purchasing a single ticket.

Finally, they announce the winning numbers! Unfortunately, they aren't the numbers you picked. This means the updated value of your lottery ticket would be

$$V_{total}(ticket) \ = {\sim}0 + \ \$1{,}000{,}000{,}000 \ * \ 0 \ = \$0$$

The ticket is now worthless because its value was almost entirely speculative and predicated on the success of an outcome that did not occur.

Collateralizing a currency

Using collateral assets to guarantee the value of currencies is a popular approach to stabilizing their value and ensuring consumer confidence. As demonstrated above, collateralizing a currency with secondary assets allows us to explicitly define its intrinsic value quotient and minimize the speculative volatility that would otherwise be caused by political and economic forces. The primary consideration in selecting a collateral mechanism for a currency is to ensure that it retains its relevance to society as a store of intrinsic value. Currency is not acquired for immediate consumption, but to purchase goods and services at a later date. For it to be effective in that role, a user must have a high degree of confidence that the currency can be exchanged in the future with minimal loss of value. This raises the question of what assets one would need to select to function as the collateral of a flatcoin. Here, we discuss various mechanisms commonly used to collateralize currencies.

Nothing

Choosing not to collateralize a currency is an attempt by the issuing body to transact in the marketplace based on their reputation and/or creditworthiness. Such currencies have no intrinsic value and are particularly susceptible to crises of consumer confidence. In market booms, these currencies give the appearance of being sound and stable assets. In market downturns and small-scale negative liquidity events, these instruments present themselves to be the volatile, highly speculative assets that they are. When these currencies are issued by private entities (either centralized or decentralized), they are only valued by market sentiment and scarcity. When governments issue uncollateralized currencies, they essentially maintain their value through force and the ability to require the use of those currencies for tax payments and official purposes.

Government Fiat

If a single fiat currency were utilized as collateral, users would be relying on the issuing body of the fiat currency to ensure its value. This defeats the entire purpose of creating a secondary currency, as it only increases the potential risk to its users. If, in an attempt to diversify the collateral, the issuing body utilizes a pool of different fiat currencies to reduce volatility, they are also doomed to failure. This is because global economic factors do not affect each country independently. A crisis that causes one country to mint large amounts of currency (e.g., the COVID-19 pandemic) will inevitably have a similar effect on countries with connected economies.

Traditional Cryptocurrency

Collateralizing a currency with traditional cryptocurrencies (bitcoin, ether, etc.) exposes the issuer to all the risks associated with using fiat currencies as collateral, as well as all the risks associated with using nothing as collateral. The currency issuer’s value proposition has effectively been delegated to the cryptocurrency issuer, and they are now wholly dependent upon the value and stability of one or more secondary cryptocurrencies. Additionally, since traditional cryptocurrencies aren’t collateralized by an intrinsic value store, the currency issuer would also be subject to crises of consumer confidence brought on by the volatility and highly speculative nature of the underlying cryptocurrencies.

Using multiple traditional cryptocurrencies as collateral is analogous to using multiple fiat currencies as collateral. Popular cryptocurrencies, such as bitcoin, ether, ADA, and SOL have historically been positively correlated (Yue, 2022). This means that, like using multiple fiat currencies, collateralizing with multiple cryptocurrencies makes for a poor diversification strategy.

Stocks/ETFs

A single stock may arguably be as volatile as a traditional cryptocurrency. One advantage that a stock has over traditional cryptocurrencies is that a stock’s value is largely based on a company’s performance. Even if the company were to declare bankruptcy, the shareholders would be entitled to a portion of the company’s assets, commensurate with the equity that they hold. ETFs are better diversified than individual stocks, but ETFs often have no relationship to purchasing power or anything that people want to buy. How people value a set of companies has no relationship to the fundamentals of purchasing power.

Commodities

The relative scarcity and universal nature of a commodity have made this an effective option at various points in history. However, the potential volatility of any one commodity is too substantial for it to be the only reference point for intrinsic value to operate as a global currency. Commodities isolate the intrinsic value quotient more effectively than any other financial instrument, but a single commodity does not reflect or track aggregated purchasing power.

Decentralized value consensus

To encapsulate purchasing power using assets whose value is predominantly intrinsic, we must utilize a defined set of commodities that represent the fundamental value assessments of humanity. These assets also need to be highly liquid and publicly traded in order to maintain that consensus in real-time. To represent aggregated purchasing power, the assets of an intrinsic value hedge must also be effectively diversified with minimal correlations of price and overlap of fundamentals.

Two problems arise when attempting to define a discrete list of assets to serve as our collateral: we have no way to quantify intrinsic value, and the intrinsic value of any asset is subject to change. So how do we achieve value consensus in the real world? We determine what people intrinsically value, not by asking them to price arbitrary tokens in a vacuum, but by watching where they "vote" with their wallets. By analyzing consumer spending habits, we can create a meaningful set of asset allocations that models the intrinsic value assessments of a population and can be algorithmically tracked and maintained in the asset pool.

Consumer spending analysis

To build a substantive metric of intrinsic value, we reviewed a broad cross section of the economy to identify the core elements of US consumption. We then took this information and identified tradable market assets that would represent average purchasing power within the US. The chosen assets and their relative percentages in the Collypto Index are not intended to perfectly mirror consumer spending. Instead, we have used consumer spending and GDP data as a guide to create an index that effectively approximates real-world purchasing power and preserves it over time. The commodities chosen are those that are not only correlated to consumer spending but also indicate by their trading volume that sufficient market liquidity exists for them to be held in the Collypto Index.

Some may ask why we didn’t simply use the published Consumer Price Index (CPI) to identify how to best represent purchasing power. The simple answer is that you can’t buy the CPI in the market in any meaningful way. It is a measurement that falls short in its ability to indicate current real-world purchasing power due to its cross-category substitution of items in the index and the averaging of prices across longer periods of time. The normalizing effect of using geometric means on interpolated data in its index calculations also minimizes the perceived impact of price changes over the reported period. By analyzing consumer spending patterns and identifying the core inputs in the market that drive that consumption, we were able to generate a much more accurate measure of consumer purchasing power that is updated in real-time.

Block indexing

Before we could meaningfully translate aggregated spending habits into a configurable model, we first needed to build the model and the software necessary to execute it.

We initially attempted to create an algorithm that utilized a constant mix strategy with weighted corridors. This strategy consists of setting price-based percentage weights for different assets and writing an algorithm that ensures the asset pool maintains those percentages within a small margin of error (the corridor size). This approach was fundamentally flawed, and we soon abandoned it.

To understand why a constant mix strategy cannot be used to maintain a store of value, imagine if we had implemented the Collypto Algorithm as a simple constant mix strategy (corridors are omitted, and prices are simplified for illustrative purposes) with the following set of allocations:

Sample Allocations - Constant Mix Strategy

Asset	Price per Unit	Unit	Allocation Percentage
Corn	$25	1 bushel	25%
Steel	$1,000	1 short ton	25%
Oil	$100	1 barrel	25%
Gold	$1,000	1 ounce	25%

Table 1

If we have an initial value of $100,000 in the asset pool, the allocations of assets we would need to purchase are calculated as follows:

$$Q_{initial}(corn) = Floor(\$100{,}000 * 0.25/\$25) = 1{,}000\ bushels$$

$$Q_{initial}(steel) = Floor(\$100{,}000 * 0.25/\$1{,}000) = 25\ short\ tons$$

$$Q_{initial}(oil) = Floor(\$100{,}000 * 0.25/\$100) = 250\ barrels$$

$$Q_{initial}(gold) = Floor(\$100{,}000 * 0.25/\$1{,}000) = 25\ ounces$$

- $Q(asset)$ is the total quantity of an asset that we must hold to maintain our allocations.
- The $Floor$ operation means that we always round down to the nearest integer so that we never need to buy fractional shares or use margin to maintain the asset pool.

At first, this approach seems to make sense, but what happens if the price of corn doubles to $50 per bushel, but the price of the other assets remains the same?

Since we are already holding these assets, the updated dollar value of our asset pool can be calculated as follows:

$$\begin{aligned} V_{final}(portfolio) &= 1{,}000\ bushels * (\$50/bushel) + \ 25\ short\ tons * (\$1{,}000/short\ ton) + \\ &\quad 250\ barrels * (\$100/barrel) + 25\ ounces * (\$1{,}000/ounce) \\ &= \$125{,}000 \end{aligned}$$

- $V(portfolio)$ is the total value of the combined assets of our asset pool.

The price of one quarter of the assets in the portfolio has doubled, so the overall portfolio value has increased by 25%. Except that we’re using a constant mix strategy, so now, we would need to rebalance the asset pool as follows:

$$Q_{final}(corn) \ = Floor(\$125{,}000 * 0.25/\$50) \ = 625\ bushels$$

$$Q_{final}(steel) \ = Floor(\$125{,}000 * 0.25/\$1{,}000) \ = 31\ short\ tons$$

$$Q_{final}(oil) \ = Floor(\$125{,}000 * 0.25/\$100) \ = 312\ barrels$$

$$Q_{final}(gold) \ = Floor(\$125{,}000 * 0.25/\$1{,}000) \ = 31\ ounces$$

Notice that the ratios of gold to corn and steel to corn in our original asset pool were both 1 to 40, but after the price of corn increased, the resulting ratios are now approximately 1 to 20. In addition, the ratio of oil to corn in our original asset pool was 1 to 4, but the resulting ratio is now approximately 1 to 2.

This outcome is unacceptable because we have altered the fundamental relationship between the collateral allocations, decreasing our asset pool’s exposure to underlying fluctuations in the price of corn and increasing it to fluctuations in the price of our other three assets. If the price of corn doubles again, the value of our asset pool will only increase by $31,250, not by $50,000, which is what we would expect if our asset exposures had remained constant. To maintain constant exposure across both scenarios, we can’t use price percentages based on a secondary currency. We need to use an index.

Think of an index like a list of groceries. It contains items and quantities, and we need to buy all the items on the list to fill one cart. We would refer to that cart as a "block", and the objective of block indexing is to fill as many carts as possible. In other words, we see how many complete and fractional total blocks can be allocated using the total value of the asset pool. We then buy the total corresponding quantity of each asset. Using block indexing allows us to maintain constant collateral asset exposure, regardless of price fluctuations in the underlying assets.

Sample Allocations - Block Indexing

Asset	Price per Unit	Unit	Allocation Quantity
Corn	$25	1 bushel	40
Steel	$1,000	1 short ton	1
Oil	$100	1 barrel	10
Gold	$1,000	1 ounce	1

Table 2

Assume we are still starting out with an initial value of $100,000 in the asset pool, except we're using a block indexing strategy instead of the constant mix. In this scenario, we need to change the equations to the following:

$$\begin{aligned} V_{initial}(block) &= (\$25/bushel) * 1\ bushel * 40 + (\$1{,}000/short\ ton) * 1\ short\ ton * 1 + \\ &\quad (\$100/barrel) * 1\ barrel * 10 + (\$1{,}000/ounce) * 1\ ounce * 1 \\ &= \$4{,}000 \end{aligned}$$

$$T_{initial}(blocks) = \$100{,}000/(\$4{,}000/block) = 25\ blocks$$

$$Q_{initial}(corn) \ = Floor(25\ blocks * (1\ bushel/block) * 40) \ = 1{,}000\ bushels$$

$$Q_{initial}(steel) \ = Floor(25\ blocks * (1\ short\ ton/block) * 1) \ = 25\ short\ tons$$

$$Q_{initial}(oil) \ = Floor(25\ blocks * (1\ barrel/block) * 10) \ = 250\ barrels$$

$$Q_{initial}(gold) \ = Floor(25\ blocks * (1\ ounce/block) * 1) \ = 25\ ounces$$

- $V(block)$ is the total dollar value of an entire block of the index (combined allocations of all assets).
- $T(blocks)$ is the total blocks that can fit in the asset pool.
- $Q(asset)$ is the total allocated quantity of an asset that we must hold to approximate the index.

Notice that we arrive at the same physical allocations as we did in the initial phase of the constant mix strategy. The total dollar value of a block is calculated by multiplying the price per unit and the allocation quantity of every asset in the index and then adding them together. The total blocks in the asset pool can be found by dividing the asset pool value by the dollar value of a single block (rounded down to the nearest integer).

Now let's again assume that the price of corn doubles and the prices of the rest of our assets remain constant. That means corn now costs $50 per bushel, steel is still $1,000 per short ton, oil is still $100 per barrel, and gold is still $1,000 per ounce, and since we were initially holding the same assets as in the constant mix strategy, the resulting dollar value of our asset pool is now $125,000.

Now let’s rebalance the asset pool using block indexing. The index allocations and resulting asset pool allocations will not change, but we will verify the resulting values for illustration.

$$\begin{aligned} V_{final}(block) &= (\$50/bushel) * 1\ bushel * 40 + (\$1{,}000/short\ ton) * 1\ short\ ton * 1 + \\ &\quad (\$100/barrel) * 1\ barrel * 10 + (\$1{,}000/ounce) * 1\ ounce * 1 \\ &= \$5{,}000 \end{aligned}$$

$$T_{final}(blocks) = \$125{,}000/(\$5{,}000/block) = 25\ blocks$$

$$Q_{final}(corn)\ = Floor(25\ blocks * (1\ bushel/block) * 40)\ = 1{,}000\ bushels$$

$$Q_{final}(steel)\ = Floor(25\ blocks * (1\ short\ ton/block) * 1)\ = 25\ short\ tons$$

$$Q_{final}(oil)\ = Floor(25\ blocks * (1\ barrel/block) * 10)\ = 250\ barrels$$

$$Q_{final}(gold)\ = Floor(25\ blocks * (1\ ounce/block) * 1)\ = 25\ ounces$$

$$\begin{aligned} V_{final}(portfolio)\ &= 1{,}000\ bushels * (\$50/bushel) + 25\ short\ tons * (\$1{,}000/short\ ton) + \\ &\quad 250\ barrels * (\$100/barrel) + 25\ ounces * (\$1{,}000/ounce) \\ &= \$125{,}000 \end{aligned}$$

- $V(portfolio)$ is the total value of the combined assets of our asset pool.

We can see that our resulting asset pool allocations are identical to where they were before the price of corn doubled, and our asset exposures (the ratios between allocated quantities of assets) have remained constant. Unlike the constant mix strategy where a price change in one commodity leads to changes in the relationship between the physical allocations in our asset pool, block indexing keeps physical allocations constant. This results in a more accurate and consistent reflection of changes in purchasing power at any given point in time. In fact, the gold standard can be understood as a block indexing system that tracks the value of gold using blocks with a single asset allocation.

In practice, the block indexing process conducted by Collypto’s automated trading system is far more complex than the sample shown above. Our real-world implementation also accounts for many other factors, such as real-time data retrieval, block fractionalization, order randomization, and configurable allocation corridors to effectively track the Collypto Index and account for technical requirements beyond the scope of these illustrations.

Defining the Collypto Index

As demonstrated above, it isn't enough to take a set of simple percentages that represent fundamental consumer spending and allocate the asset pool based on the dollar value of those assets, but this raises the question of how to translate data that is based on percentages of consumer and national spending into block allocations of the Collypto Index.

The actual Collypto Index contains 40 allocations of publicly traded intrinsic value assets (ETFs and futures) in the categories of real estate, materials, food, precious metals, and energy. Whenever possible, we have utilized financial instruments where the assets are physically held, such as REIT ETFs for real estate and precious metals ETFs that represent securely held physical stores. For perishable assets or those that are not practical to store, we have utilized diversified sets of futures contracts that track the value of the underlying physical commodities.

In the case of ETFs, the expense ratios introduce collateral decay[3] that we have accounted for in our allocation provider, which adjusts the total allocations of real estate, gold, silver, and platinum that must be purchased to maintain the same asset exposure as our original rebase date. The rebase date is effectively the reference point that we used to initialize the Collypto Index to the baseline allocations of a single block of assets that tracks aggregated purchasing power.

In the case of futures, we ran historical contract data through an optimizer that utilizes a machine learning algorithm to determine contract allocations that minimize the block discontinuity[4] of our aggregated contract prices for each product across rollover periods. This ensures the highest possible correlation to the spot price of their respective underlying commodities. The resulting allocations of these futures contracts were then tempered via depth-of-market surveys to verify the volume and market demand of underlying assets prior to rebasing the allocations of the Collypto Index.

To rebase these allocations into the baseline composition of the Collypto Index, we first examined the results of the consumer spending analysis and determined a working set of starting percentages for every product in all five asset categories. Next, we determined a date that would function as the reference point for rebasing the Collypto Index from baseline percentages into constant asset allocations for both futures and ETF assets. We chose March 3rd, 2013, as our rebase date, approximately the date that the market recovered to its previous peak following the Great Recession. Finally, we rebased the allocations of each product into a block of assets that preserved the notional value relationships of their respective baseline percentages on the rebase date. The value of the Collypto Index is calculated at any given time by taking the sum of fair market values of all assets in a single block and dividing the resulting notional value by a constant factor.

After creating the initial version of the Collypto Index, we ran many additional tests with various permutations of baseline product percentages to determine their effects on the resulting historical index compositions and backtested asset pool performance using market data from 2012 through the present. We finalized the Collypto Index using baseline allocations that best tracked the results of our consumer spending analysis over the last decade.

Utilizing the Collypto Algorithm to track the Collypto Index in our asset pool allows us to effectively maintain and tokenize purchasing power and peg the “price” of a Collypto Credit to the decentralized value consensus of the underlying intrinsic value assets, rather than a secondary currency or commodity. This value-peg system is the defining behavior of a flatcoin and the fundamental nature of Collypto.

Baseline Composition of Collypto Index

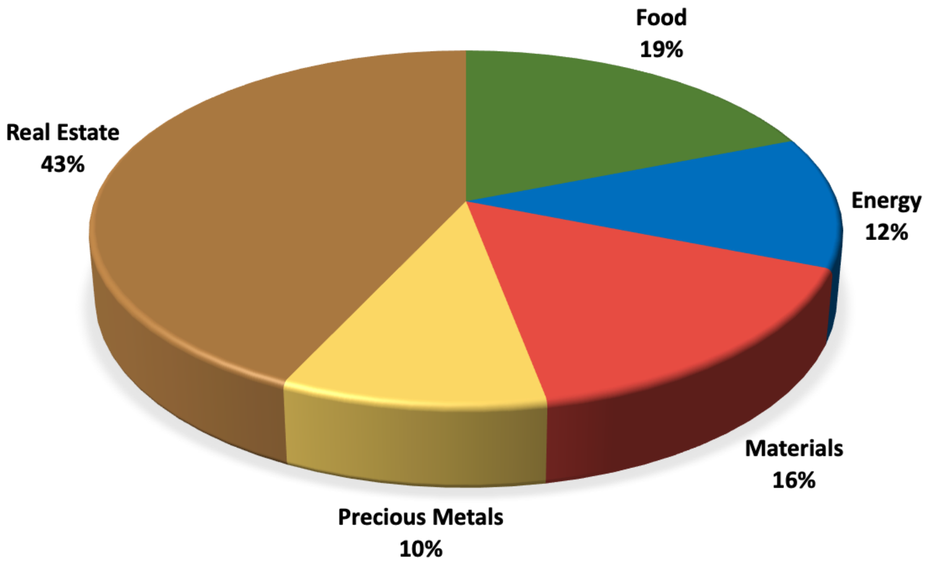


Figure 1

Collypto Index vs. Consumer Price Index (CPI)

To illustrate the relationship between the Collypto Index and purchasing power, we have compared the backtested "historical" values of the Collypto Index and Collypto Credit to the Consumer Price Index (CPI) over the last decade. The demonstrated relationship between these datasets is not artificial, and the CPI is not a contrived trendline for those graphs. The only adjustment made to these values was to normalize them to a starting value of 100 so that we can view their relative behaviors from a common point of origin. The date range displayed is from January 3rd, 2012, through November 30th, 2022, which represents the earliest available date of our historical market data through the most recent value of the CPI.

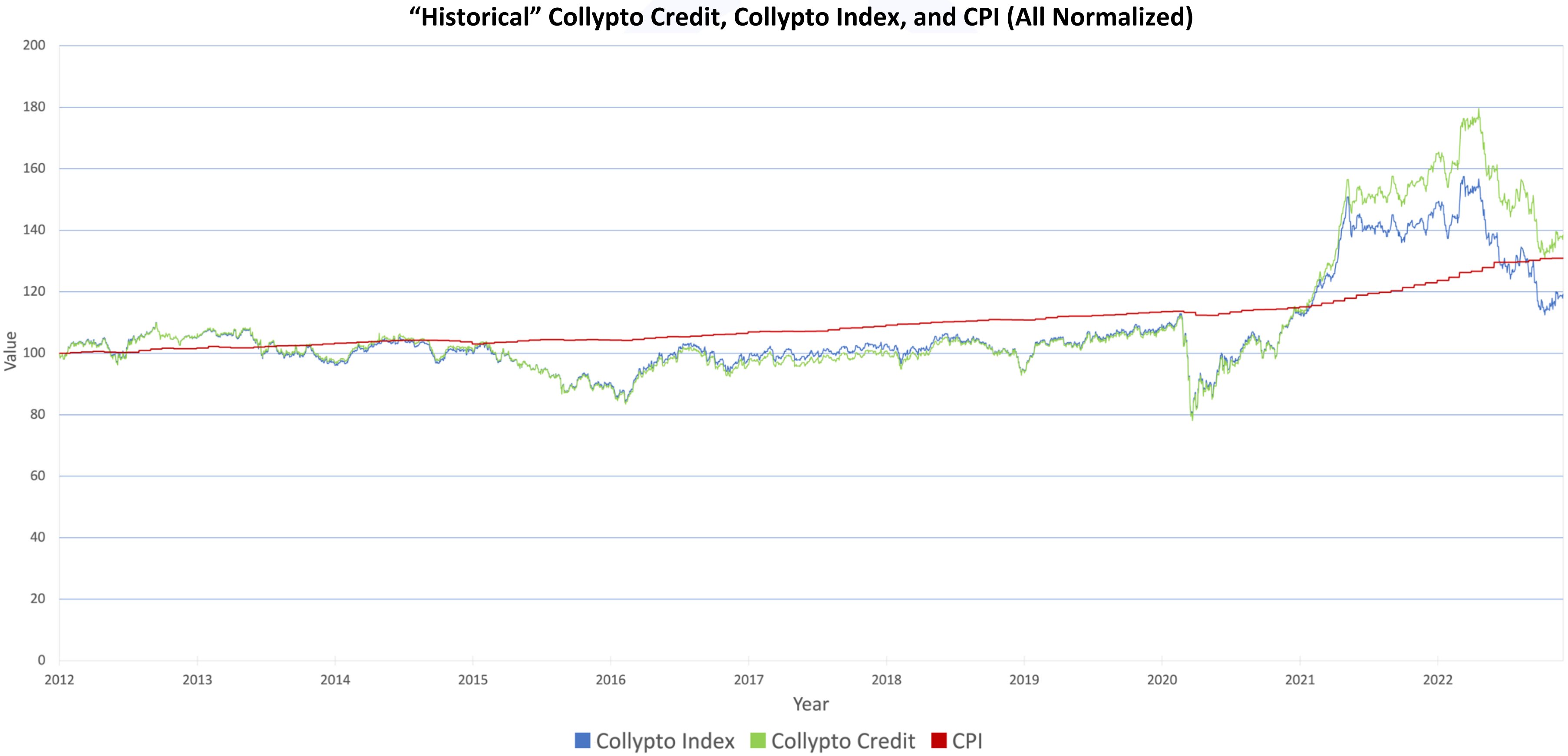


Figure 2

CPI data from "Consumer Price Index for All Urban Consumers: All Items in U.S. City Average", by Federal Reserve Bank of St. *Louis. St Louis Fed*, 13 Dec 2022.

The performance of the Collypto Index underscores its ability to accurately measure and track purchasing power in any market cycle to a much higher resolution than the CPI ever could. The Collypto Index serves as a benchmark by which we evaluate credit performance. The deviation between the backtested performance of the Collypto Index and the Collypto Credit is a byproduct of block discontinuity and an assumed maximum margin of 1% for our allocation corridors.

As a real-time measurement, the Collypto Index gives a more precise gauge of purchasing power than the CPI at any point in time. It should be noted that, as a measure of value, the price of Collypto has an inverse relationship to the price of the dollar. Upward trajectories of credit price indicate periods where the purchasing power of the dollar is in decline, whereas downward trajectories of credit price indicate periods where the purchasing power of the dollar is increasing. Since Collypto is fundamentally an intrinsic value store, fluctuations in the market price of credits are indicative of changes in the prices of underlying collateral assets.

The limitations of CPI are most clearly evidenced over the course of the COVID-19 pandemic. During March 2020, the Collypto Index depicts a sharp downward trend. This decline does not reflect a drop in the value of Collypto, but rather an increase in the purchasing power of the dollar. As the world grappled with the economic uncertainty associated with a global pandemic, the demand for the US Dollar temporarily skyrocketed in a significant deflationary event. Meanwhile, this barely registered as a blip on the CPI due to significant averaging and interpolation.

Immediately following the March 2020 deflationary dip, the Collypto Index and credit values began to trend upward and returned to baseline levels as inflationary policies were enacted to assuage economic fears. By November 2020, the inflation brought on by quantitative easing created a market response that even outpaced the Collypto Index. It is at this juncture that we see an increase in tracking error as the credit price rises more quickly than the Collypto Index to keep pace with the market price changes of the underlying assets. After this point, credit values continued to track the Collypto Index, but in a parallel path that clearly illustrates the lasting impact of the inflationary period. While this period was marked by historically high inflation rates, the delayed averages used to calculate CPI depict nothing more than a gentle upward curve.

Blockchain integration

A blockchain can be understood as a massively decentralized ledger that has maximized data integrity at the expense of all other properties, so we limit direct utilization of the blockchain to only where necessary. To that end, we implement as much functionality as possible using off-chain solutions and support applications. Our customer data is maintained in a secure CRM, and our product and application data are stored in cloud-based data repositories. We recognize the blockchain only as a form of proof of ownership and identity, not law or absolute truth.

The fair market value of a credit is calculated by dividing the total value of our asset pool by the total number of credits in circulation. The total number of credits in circulation is calculated by taking the total supply (converted to credits) and subtracting the current credit balance of the Collypto Vault. We will always utilize designated vault accounts when minting new credits prior to issuing them to customers, and the location of vaults will be explicitly defined and listed on our website. Minting and burning activities will always be coordinated with our automated trading systems to ensure that the fair market value of our token is accurate, each credit corresponds to a constant portion of the asset pool, and the collateralization state of our system is never disrupted. Uncollateralized credits will never be issued into circulation.

Denominations of Collypto

In accordance with the conventions of Ethereum, a single credit is divisible into up to 10^18 slivers, where the "sliver" is the smallest unit of precision. The sliver is the standard token of the Collypto contract, and all Collypto blockchain transactions are technically conducted in slivers, rather than credits. The “credit” is the standard denomination of Collypto, and prices of goods and services should be stated in credits and other traditional currency denominations listed in the table below when conducting common transactions.

Denomination	Total Credits	Total Slivers
Credit	1 credit	1e18 slivers
Half	0.50 credits	5e17 slivers
Quarter	0.25 credits	2.5e17 slivers
Tenth	0.10 credits	1e17 sliver
Cent	0.01 credits	1e16 slivers
Half-Cent	0.005 credits	5e15 slivers
Sliver	1 quintillionth of a credit	1 sliver

Table 3

Issuing Credits

Issuance is the process by which we collateralize credits and place them into circulation. Newly minted, uncollateralized credits (blanks) are stored in the Collypto Vault. Before credits are issued, we purchase a corresponding amount of new assets and place them in the asset pool, utilizing the Collypto Algorithm to determine exactly what and how much to buy. This process is entirely automated in our brokerage trading application. Once minted credits have been collateralized, they are issued and enter the circulating supply. Credits are only issued into circulation after we secure the assets that guarantee their value.

Credit Issuance Process

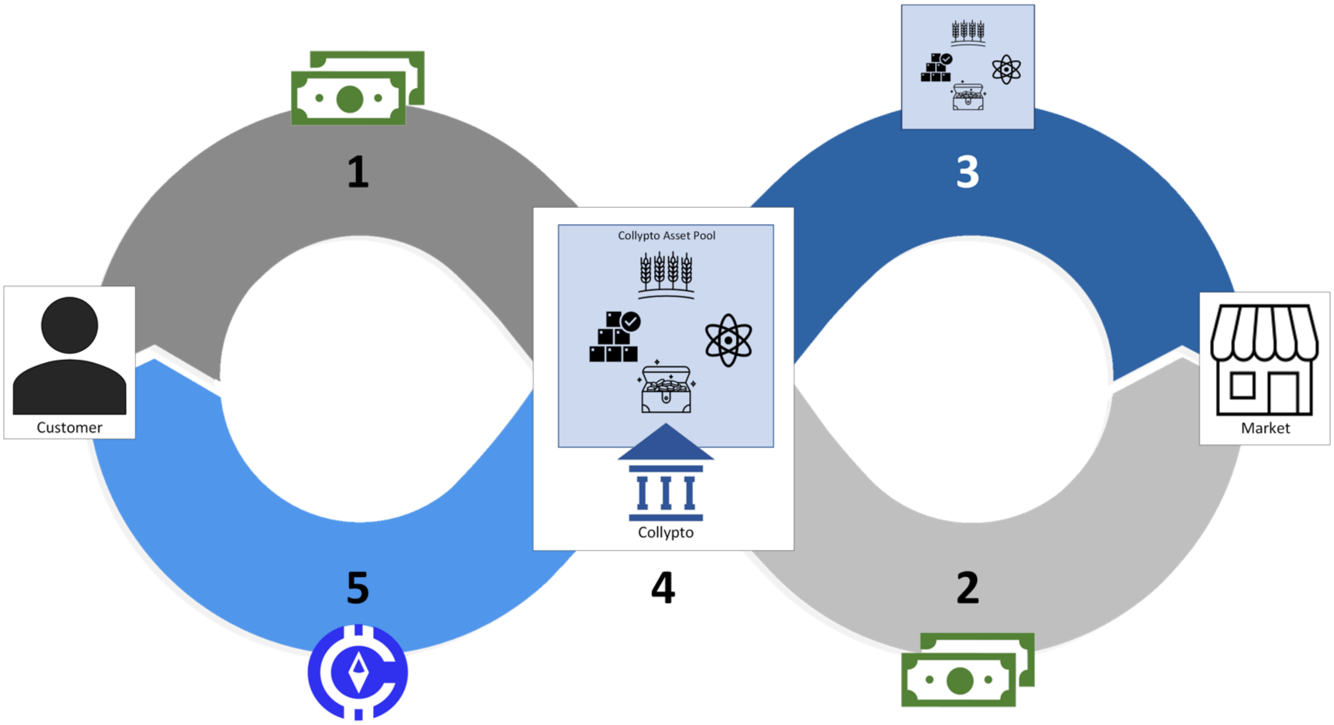


Figure 3

1. Customer submits their cash to buy credits.
2. Collypto uses the cash to purchase assets that will collateralize the credits.
3. Purchased assets are added to the Collypto asset pool.
4. Blanks are minted in the vault as needed.
5. Credits are issued to the customer.

Redeeming Credits

Redemption is the process by which we remove credits from circulation in exchange for their fair market value in fiat currency. When credits are redeemed, we sell a corresponding amount of collateral assets in the asset pool, utilizing the Collypto Algorithm to determine exactly what and how much to sell. This process is entirely automated in our brokerage trading application. A cash payment is then made to the seller, based on the current market value of their credits. Redeemed credits may be burned or stored as blanks in the Collypto Vault.

Credit Redemption Process

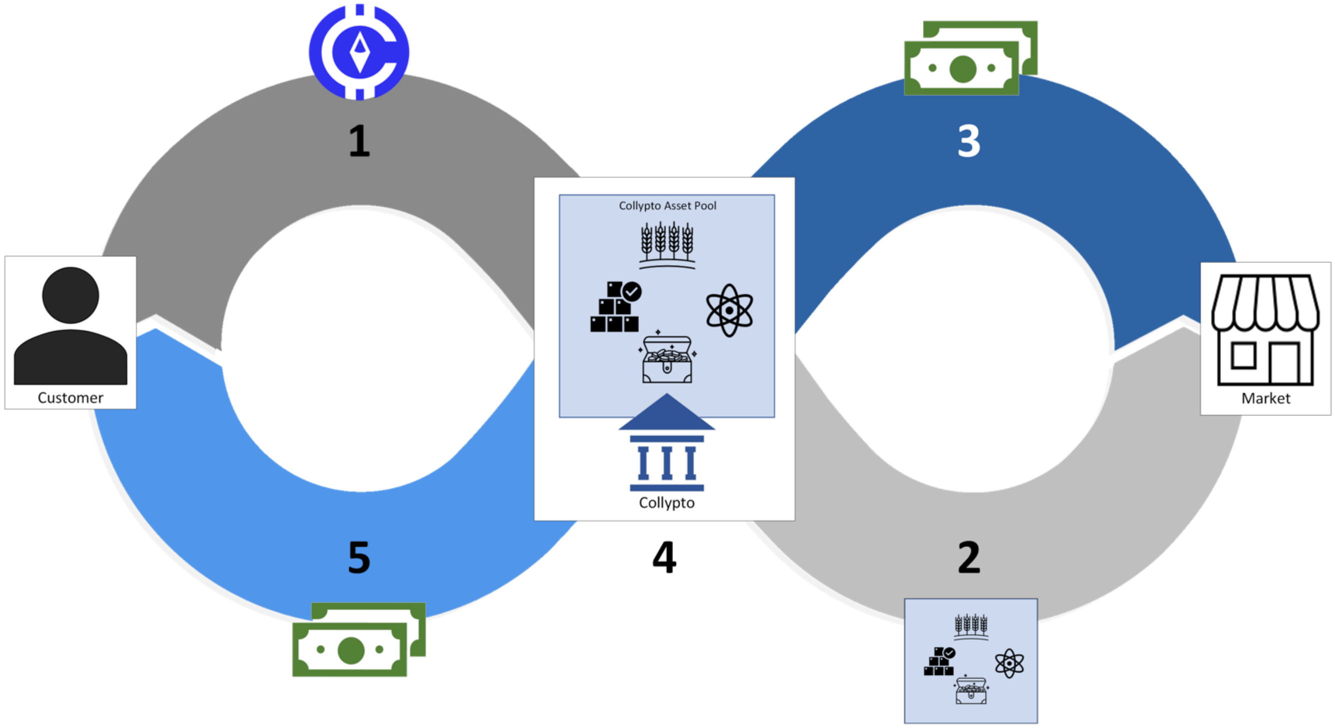


Figure 4

1. Customer submits their credits to be redeemed for cash.
2. Corresponding collateral assets are identified in the asset pool.
3. The identified assets are sold at the current market price.
4. Blanks are burned in the vault as needed.
5. Cash is paid to the customer.

Enhanced User Status

Collypto provides an on-chain user status for all Ethereum accounts that represents a simplified public counterpart to our internal CRM records and may be queried in the Collypto contract using view functions and a target address. The `UserStatus` record of an Ethereum account has two properties: `status` and `info`. The value of `status` may be designated as one of five values at any given time, depending on the known off-chain status of the account's owner.

The `info` property of an account's `UserStatus` is an empty string by default. When this property is defined, it will be formatted using a modified JSON style list of comma-separated name/value pairs as demonstrated in later sections. Private user information will never be included in the `info` property of Ethereum accounts. This property will be used exclusively to display public medallion information and on-chain messages for users.

The five possible values of `status` are as follows:

Unknown
This is the default status of all Ethereum accounts. An `Unknown` account may belong to no one, a contract, a user who has not completed the Collypto KYC process, or a user who has registered the account with us and chosen not to designate it as their medallion.

Pending
An account with a status of `Pending` is currently undergoing the Collypto KYC process for user verification and may be designated as that user's medallion account in the future.

Verified
The `Verified` status will be assigned exclusively to medallion accounts belonging to users that have completed the Collypto KYC process. Verified transactions can only be conducted using a medallion account as their recipient.

Suspect
The `Suspect` status is reserved for accounts that are currently under public investigation for illegal activity or suspected of conduct that violates our terms of service.

Blacklisted
The `Blacklisted` status is reserved for accounts belonging to known malicious actors, individuals, or contracts under government sanction, and individuals or contracts that have violated our terms of service. Blacklisted accounts are automatically locked and may not conduct any type of transaction in the Collypto contract.

Collypto Medallion System

Users will have the option of obtaining a `Verified` status on a chosen account (their medallion) after successfully completing the Collypto KYC process. A medallion is essentially a pseudonymous way of proving to the world that Collypto has a user's information on file, and they are legitimate, even if other users can't personally verify their private information. It functions as a user's on-chain proof of identity while keeping their personal information private. This is analogous to the way the blockchain allows users to prove that they own a private key without sharing it with others. Users will always have legal recourse through Collypto if someone attempts to defraud them of their credits using a medallion or a registered account.

Not your medallion, not your identity

- Medallions are available exclusively through Collypto Technologies, Inc.
- Medallions are not NFTs, they are an on-chain representation of a user’s verification status in our off-chain internal systems.
- The value of medallions is derived exclusively from the public trust in Collypto Technologies, without which, they would be worthless.
- The Collypto Medallion System is the cornerstone of our extended contract functionality and user protections.
- Upon verification of identity, we will allow our users to select a single Ethereum account that we will refer to as their “medallion”.
- Medallion accounts may be changed upon request, but a medallion transfer will always require the user to repeat the Collypto KYC process.

A medallion-certified Ethereum account will always have a `status` value of `Verified` and `info` value in the format `{'creationDate':'yyyy-MM-dd','expirationDate':'yyyy-MM-dd','message':''}` where the `message` property is optional. These values may be retrieved using the `userStatusOf` function of the Collypto contract, and a user’s medallion account can be trivially verified using the `isVerified` function.

The `creationDate` is the date that the user received their original medallion certification, the `expirationDate` represents the last day the medallion will be valid before its owner will need to reverify their identity, and `message` is an optional property that we can use to display an on-chain message for any Ethereum account. A user's `creationDate` will never change unless we are required to issue them a new medallion after their previous one was compromised by a malicious actor. If a user loses their physical medallion wallet and contacts us, we will verify their identity and move their medallion status to a new account that they will also need to verify ownership of, in addition to repeating the Collypto KYC process. Our verification and status retrieval functions can be used to check the `status` of an account before referencing it in any contract operation or the transfer of any Ethereum-based token.

A medallion isn't just proof that someone is verified and accountable. It also means that their country is represented in the Collypto Index. The Collypto Medallion System is universal and will never be segregated into national or jurisdictional partitions. We will only award medallion status to a country after determining that it provides sufficient legal recourse to protect property rights and contains compatible financial instruments for us to hold assets there. If a country doesn't offer acceptable recourse to us and our users, we cannot issue medallions to its residents. Medallions are not localized by region because recourse and property rights are not relative. If Collypto cannot insure a foreign asset, it will not be included in our index. At launch, the United States will be the only country with medallion status, and medallions will only be available to its residents.

Registered Accounts

Verified users can register several Ethereum accounts with us, but individual users will only be allowed one medallion account at launch. The rest of their Ethereum accounts will be registered in our off-chain internal systems. Account registration is strictly confidential, and the `status` value of registered accounts will be `Unknown` in the Collypto contract, and the `info` value will default to an empty string.

Verified Transactions

In addition to all standard ERC-20 functions and extended user functions, we have also implemented verified counterparts of these operations which provide an additional safety corridor for our users in the event of theft or fraud. Verified transactions allow users to have the confidence that they are transacting with a user who has already been verified through Collypto’s KYC process, facilitating both internal and legal recourse. We have provided the sample flow of a successful Verified Transfer below.

Verified Transfer Process

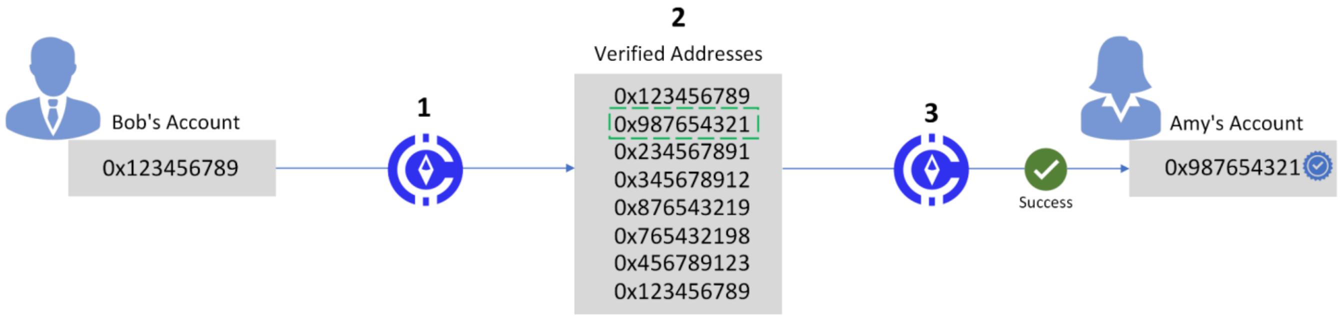


Figure 5

1. Bob initiates a Verified Transfer to Amy.
2. The Collypto contract checks the `status` of Amy’s account.
3. Since Amy’s account is verified, the Verified Transfer is processed successfully.

When a verified function is called, it automatically checks the `status` of the receiving account. If it is `Verified`, the transaction will go through. Otherwise, it will revert with an error message explaining that the recipient is not verified.

This solves two ubiquitous problems that are devastating to both accessibility and adoption in the crypto space.

1. Users can easily check each other's verification status before sending a payment to confirm that Collypto has the recipient's KYC information on file, so they know they have recourse in cases of fraud.

2. Users cannot accidentally mistype an address in any verified transaction, since the odds of the mistyped recipient address belonging to an unintended verified user are effectively the odds of successfully brute forcing someone's private key (functionally impossible with present technology). In this way, Collypto uses the minuscule probability of a hash collision to serve users, rather than punish them for common mistakes.

Virtual Cold Storage (VCS)

Cold storage devices protect tokens by insulating them from the typical entry points of attackers. Since transactions are signed within the device, the user's private key is never exposed. This makes it impossible for the device and its associated funds to be compromised remotely. Virtual Cold Storage (VCS) gives users all the benefits of traditional cold storage with the additional barrier of Collypto's security infrastructure.

VCS has a two-tiered structure, allowing users to choose the level of added security they deem best for their account. They have the option of using a Limited Freeze to freeze a specified number of credits or a Complete Lock to prevent all transfers of credits to or from their account. VCS users maintain custody of their credits at all times and will never have to share or surrender their private keys. Regardless of the option chosen, users will only be able to unlock their account or unfreeze their credits once their identity is confirmed by Collypto Technologies.

Limited Freeze

- Freezes only a specified number of credits in an account to prevent transfer by a malicious actor
- Credits do not leave a user's account but are "marked" so they cannot be used in transactions
- Leaves any unfrozen credits available for use
- Account remains unlocked and may be used to conduct operations
- Frozen funds are impervious to theft or hardware loss
- Malicious actors can't transfer frozen credits, even if they have a user's private key
- No requirement to surrender custody of credits

Complete Lock

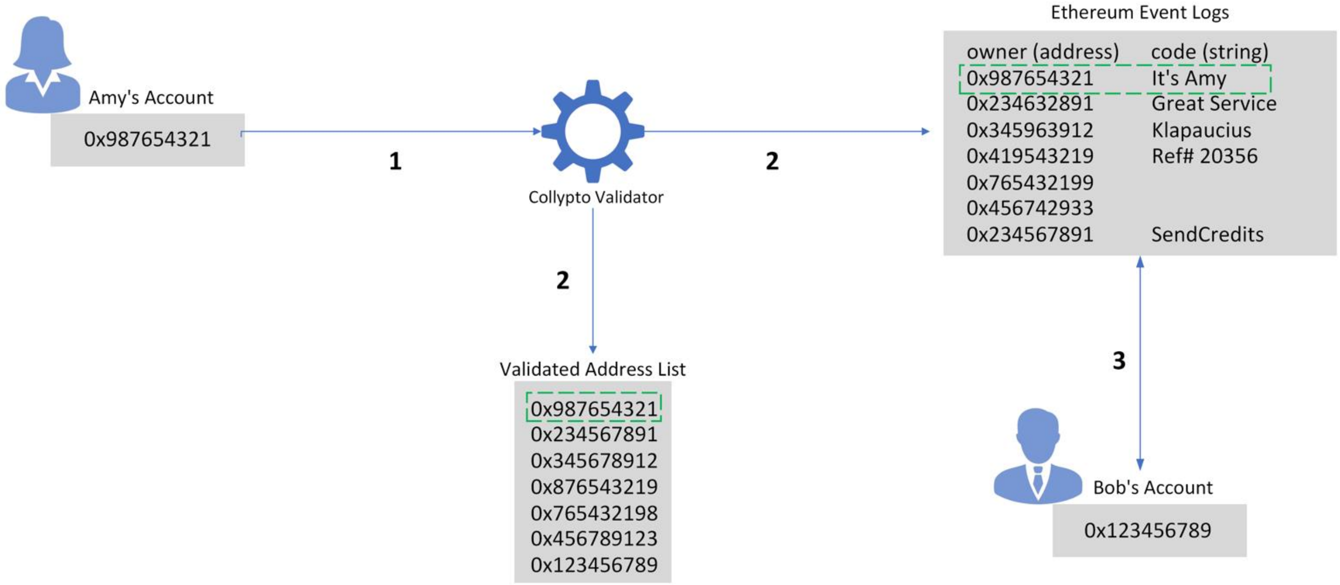
- Locks the account, preventing it from sending or receiving credits until it is unlocked
- Locked accounts may only be used to conduct view and allowance operations
- Locked accounts are impervious to theft or hardware loss
- Malicious actors can't transfer credits to or from locked accounts, even if they have a user's private key
- No requirement to surrender custody of credits

Account Validation

Some of the biggest losses in the crypto space have come from people sending crypto to the wrong address and losing their money forever. Account validation is a feature that allows users to add their addresses to the Validated Address List of the CollyptoValidator contract to maximize the integrity of peer-to-peer transactions. Before sending credits, a user will be able to confirm that they have a valid address and that their recipient is in possession of the specified address.

Though this functionality appears similar to verification status and verified transactions, validating addresses does not require transacting parties to complete the Collypto KYC process and provides no recourse in cases of theft, fraud, or account compromise. The purpose of this feature is to serve as a transactional safeguard in regions where we do not issue medallions and user verification is not available. Validation may be repeated any number of times using custom validation codes for a given Ethereum account. The account validation process is illustrated below.

Account Validation Process



Ethereum Event Logs

owner (address)	code (string)
0x987654321	It's Amy
0x234632891	Great Service
0x345963912	Klapaucius
0x419543219	Ref# 20356
0x765432199	
0x456742933	
0x234567891	SendCredits

Validated Address List

0x987654321
0x234567891
0x345678912
0x876543219
0x765432198
0x456789123
0x123456789

Figure 6

1. Amy submits a validation transaction to the Collypto Validator with the validation code "It's Amy".
2. Amy's public address is added to the Validated Address List, and the resulting event can be found in public Ethereum event logs with the validation code "It's Amy".
3. Bob checks the Ethereum event logs to confirm Amy's address and validation code.

The CollyptoValidator contract is immutable and has already been deployed to the Ethereum mainnet. As we will explore later, wallet developers can integrate the CollyptoValidator into their applications for Ethereum-based accounts and check the validation status of any address prior to conducting a transaction.

Contract architecture

With the exception of proxies, Collypto utilizes three primary contracts to conduct blockchain operations: Collypto, CollyptoManager, and CollyptoValidator. The Collypto contract is an upgradable, enhanced ERC-20 implementation that contains all of our standard, management, utility, and ownership operations. The CollyptoManager is a disposable contract that encapsulates access to the management and utility functions of the Collypto contract. This encapsulation allows us to stratify authorization to conduct these operations into “Prime” and “utility” operator classes which are assigned to designated operator accounts following the principle of least privilege. All operator classes of Category 3 and 4 are considered utility classes.

We separate authorizations into these layers to maximize internal accessibility while maintaining a strict hierarchy of protected functions. The two exceptions are the Admin and Master accounts, which are respectively utilized for Collypto contract upgrades and emergency access control. The CollyptoValidator is an immutable support contract that we have implemented to allow users to validate ownership of their Ethereum accounts using an entirely on-chain process.

Collypto Operator Classes

Class	Abilities	Category
Admin	Can update the Collypto contract	0
Master	Can conduct all contract utility, management, and ownership operations	1
Prime	Can conduct all contract management operations	2
Arbiter	Can explicitly change the `status` and `info` properties of any account	3
Dispatch	Can forcibly transfer any number of available credits between any two accounts	3
Unfreeze	Can unfreeze any number of frozen credits in any account	3
Mint	Can create any number of credits in a target account (cannot make total supply > MAX uint256 value)	3
Burn	Can destroy up to the total available balance of credits in a target account	3
Unlock	Can unlock any locked account	3
Freeze	Can freeze any number of available credits in any account	4
Lock	Can lock any account, prohibiting it from sending or receiving credits	4

Table 4

Our operator classes are stratified into five categories based on the severity of a breach of a majority of the corresponding private keys and the required response we would need to take to resume normal operations.

Breach Recovery Protocols

Severity	Response
Category 0	Terminate all operations on the current Collypto contract instance, deploy new proxies and contract instances, and airdrop balances for all users on the new instance of the Collypto contract.
Category 1	Utilize the `terminateContract` function of the Collypto contract to pause and reinitialize the Collypto contract to reference a new Master account and management contract, reverse any invalid transactions, and unpause the Collypto contract.
Category 2	Remove the current manager address (instance of the management contract) from the Collypto contract, pause the Collypto contract, add a new management contract instance to the Collypto contract with updated operator authorizations, reverse any invalid transactions, and unpause the Collypto contract.
Category 3	Pause the Collypto contract, remove compromised operators from the management contract, add updated operator authorizations to the management contract, reverse any invalid transactions, and unpause the Collypto contract.
Category 4	Remove compromised operators from the management contract and reverse any invalid transactions.

Table 5

Credits can never be stolen without recourse since our underlying collateral will always be insured, our privileged operations will always be monitored, and Collypto will never redeem credits for a user without identity verification and a standard waiting period. In the case of a breach of Category 3 or above, the Collypto contract will be paused, and all redemption operations will be temporarily suspended until we have verified that all fraudulent transactions have been reversed and accounts have been restored to a valid state. All privileged operator keys are maintained exclusively in offline hardware wallets.

Collypto Contract (ERC-20)

Overview

We have followed general best practices of Solidity and OpenZeppelin, including operation reversion on failure, allowance support functions, and a transparent upgradable proxy structure that utilizes an initializer function instead of a constructor. In addition to the required and verified versions of standard ERC-20 operations and extended operations, we have included functionality for minting and burning tokens, freezing and unfreezing tokens, locking and unlocking Ethereum accounts, pausing and unpausing this contract, and forcibly transferring tokens, as well as enhanced management and ownership operations. We have also included a user status model that represents the off-chain status of an address owner in our internal systems, as well as operations to allow us to update the user status of Ethereum accounts and provide public visibility of Ethereum account status. All balances, transfers, and other token operations are stored and conducted in slivers, and third-party applications must perform necessary denominational conversions prior to conducting transactions.

View Operations

`name, symbol, decimals, totalSupply, balanceOf, frozenBalanceOf, availableBalanceOf, allowance, userStatusOf, isLocked, isUnknown, isVerified, isBlacklisted, isSuspect, isPending, isManager, isOwner, isPaused, initializationIndex, isInitialized`

This contract contains all required ERC-20 view functions and several additional view functions that allow users to reference the state of our enhanced contract features. View operations can be conducted by any account.

Standard Operations

`transfer, transferFrom, approve, increaseAllowance, decreaseAllowance`

In this contract, we have implemented all required ERC-20 functions and events, including two additional functions that allow users to increase or decrease the allowance of a specified account. Transfer operations can only be conducted by unlocked accounts, and blacklisted accounts cannot conduct non-view operations.

Verified Operations

`verifiedTransfer, verifiedTransferFrom, verifiedApprove, verifiedIncreaseAllowance, verifiedDecreaseAllowance`

In addition to implementing all required ERC-20 functions and extended functions, Collypto also provides users with "verified" versions of all public token operations. Any call to one of these functions requires the recipient address to have a `status` value of `Verified` (corresponding to a medallion account), otherwise, the operation will revert.

Utility Operations

`updateUserStatus, forceTransfer, freeze, unfreeze, lock, unlock, mint, burn`

This contract contains eight utility operations delegated to the stratified set of operator classes used to conduct transactions through the multiplexed access controls of the CollyptoManager contract. These functions will only be utilized by authorized systems and representatives of Collypto Technologies and will never be available to the public.

User Status

Collypto maintains a `UserStatus` record for every possible Ethereum account in the `_userStatuses` mapping, and each corresponding `UserStatus` record contains exactly two properties: `status` and `info`. The `status` value of a `UserStatus` record defaults to `Unknown`, and the `info` value defaults to an empty string. We refer to a user's primary verified Ethereum account as their "medallion", and upon receiving a request for verification from a user, we will update the `status` value of their medallion account to `Pending` as we conduct the Collypto KYC process.

When a user has completed our verification process and they have confirmed ownership of their medallion account, we will use the `updateUserStatus` function to update the `UserStatus` record corresponding to their medallion address to contain a `status` value of `Verified` and an `info` value of a string in the format

```
{'creationDate':'yyyy-MM-dd','expirationDate':'yyyy-MM-dd','message':''}
```

where `creationDate` and `expirationDate` are dates represented as strings in ISO 8601 standard date format and `message` will default to an empty string.

Users may register other non-medallion Ethereum accounts in our off-chain internal systems, but the `status` of those accounts will remain `Unknown`, and their `info` will remain an empty string. The `creationDate` and `expirationDate` properties will never be included in the `info` value of non-medallion accounts. In addition to the verification model, we have also included functionality to mark accounts as `Suspect` or `Blacklisted` if they are implicated in criminal activity, government sanctions, or other violations of our terms of service. Ethereum accounts with a `status` of `Blacklisted` will automatically be locked and unable to send tokens, receive tokens, or perform allowance operations.

The `status` and `info` of a specified Ethereum account may be retrieved using the `userStatusOf` function, and the various `status` values can be checked using the `isUnknown`, `isPending`, `isVerified`, `isSuspect`, and `isBlacklisted` functions for each possible status.

Force Transfer

The `forceTransfer` function allows us to forcibly transfer tokens that were stolen or fraudulently obtained from a malicious actor's Ethereum account back to the Ethereum account of a victim without corrupting the collateralization state of our system.

Freeze/Unfreeze

In addition to the standard `_balances` mapping for Ethereum account token balances, we also include the `_frozenBalances` property which is used to represent the total frozen tokens in a user's Ethereum account. The `freeze` function allows us to freeze a specified number of tokens during an investigation or government sanction. This facilitates the "Limited Freeze" feature of our Virtual Cold Storage (VCS) service for verified users. Frozen tokens can only be unfrozen using the `unfreeze` function which allows us to unfreeze a specified number of frozen tokens in a target account. The total account balance can be retrieved using the standard `balanceOf` function, the frozen balance can be retrieved using the `frozenBalanceOf` function, and the available balance can be retrieved using the `availableBalanceOf` function.

Lock/Unlock

As an additional security feature, we maintain the `_lockedAddresses` mapping which represents the lock status of all Ethereum accounts. The `lock` function locks a specified account, preventing it from sending or receiving tokens. This facilitates the "Complete Lock" feature of our VCS service for verified users and provides a secondary safeguard for blacklisted accounts. Locked accounts can only be unlocked using the `unlock` function, and the `isLocked` function can be used to check whether a specified account is currently locked.

Mint/Burn

We have implemented standard `mint` and `burn` functions with respective `Transfer` events from and to the zero address in addition to their respective `Mint` and `Burn` events. Both mint and burn operations may be conducted on any target account, rather than a hard-coded vault account. This allows us the flexibility to change our vault location without requiring an update to this contract. Minted tokens are always unfrozen by default, and frozen tokens cannot be burned without first being unfrozen.

Management Operations

`pause, unpause, addManager, removeManager`

In addition to the utility operations listed previously, this contract contains four management operations that facilitate disaster recovery and allow us to securely update our management contract. These functions will only be utilized by authorized systems and representatives of Collypto Technologies and will never be available to the public.

Pause/Unpause

This contract includes functions to `pause` and `unpause` all public operations. This is essential in the event of a security breach and allows us to mitigate the damage that could otherwise be caused by a malicious actor or institution. Any majority key compromise of our management contract that is Category 3 or above would require us to pause the contract to rectify the situation and reverse all malicious transactions. The running state of this contract is maintained in the `_isRunning` Boolean property and can be checked at any time using the `isPaused` function.

Secure Management Transfer

In order to allow our management contract to be replaceable, this contract includes functions to add and remove a single manager address from the `_managerAddresses` array (the manager list). These functions can only be called by a management account, which includes the Master address (maintained in the `_ownerAddress` property of this contract). The Master address cannot be updated or removed by a standard management account.

Management transfers utilize the `addManager` and `removeManager` functions to ensure that the new management address is added to the manager list before the old one is removed, and a manager account cannot remove itself from the manager list. This means that, unlike the traditional ownership model, it is impossible for our team or systems to accidentally lose control of this contract by accidentally typing in the wrong address value for a new management contract account. With the exception of contract updates, the manager list will only ever contain a single address value stored at `_managerAddresses[0]` (the management contract address) for support operations.

Ownership Operations

`initialize, purgeManagers, terminateContract`

Collypto maintains an additional layer of security beyond the level of our management account, and that is the Master address maintained in the `_ownerAddress` property of this contract. In addition to being able to conduct management operations, the Master address can also conduct three additional operations to initialize this contract and provide recourse for disaster recovery (up to and including a

Category 1 breach). These functions will only be utilized by authorized systems and representatives of Collypto Technologies and will never be available to the public.

Contract Initialization

As previously stated, this contract is upgradable, and its storage model is compliant with OpenZeppelin upgradable contract requirements. The storage model of this contract has been optimized for our solution requirements, and this contract utilizes the `initialize` function, rather than a constructor, to define its name, symbol, owner, and management address. The initialization state of this contract is maintained in the `_isInitialized` Boolean property and may be checked using the `isInitialized` function. The initialization counter of this contract is maintained in the `_initializationIndex` property which allows us to track how many times it has been updated and may be checked using the `initializationIndex` function.

Purge Managers

In the event that the majority of Prime keys that control the management contract are compromised, the Master account can be used to purge all management addresses from the manager list using the `purgeManagers` function. At this point, we would need to deploy a new instance of the management contract, and we would then use the Master account to add the management contract address to the manager list.

Contract "Termination"

In the event that the Master key itself is compromised, it can still be used to call the `terminateContract` function which clears the manager list, resets the value of `_ownerAddress` (the Master address) to the zero address, resets the value of `_isInitialized` to `false`, and pauses this contract. At this point, we would need to use the Admin account to update this contract with a new value for the `_ownerAddress` (the address of the new Master account). Next, we would need to use the new Master account to reinitialize this contract with a new address for the updated management contract. This termination function does not reference the `selfDestruct` operation, as that would effectively break our logic contract and render it unusable.

CollyptoManager Contract

Overview

This contract operates as an application-specific multiplexed multisignature access control system with social recovery. Operational authorizations are stratified into operator classes, where each operator class may conduct a defined set of operations. Management and utility operations are enacted via a proposal system where the proposal ID is generated using the enumerated `Operations` value and data parameters of the operation itself. We have not designed this contract to be upgradable. When updates are required, we will simply discontinue use of the current contract instance in favor of a new instance of this contract.

Proposal System

When a utility or management operation function is called by an authorized operator, a proposal is created with an ID that is generated using a uniquely deterministic hash of its input parameters. When a given proposal receives a majority of approvals from its authorized operators, this contract will attempt to execute it on the Collypto contract and will return the proposal ID and the Boolean result of its execution.

Proposals are stored in the format defined in the `Proposal` struct, and `Proposal` records are stored in the `_proposalMap` for O(1) retrieval. All active proposal IDs are maintained in the `_proposalList` for aggregated reference and deletion. Consequently, this contract does not support redundant proposals, and a `Proposal` record with a given `id` value and parameters may be recreated and executed multiple times, but the `index` value of each subsequent `Proposal` record will be unique for a given instance of this contract because it represents the value of `_currentProposalIndex` when that proposal was created, and the value of `_currentProposalIndex` cannot be reset.

When an operator calls a utility or management function using an Ethereum account belonging to the required operator class, this contract will determine whether they represent a majority of operators of that class. If the required operator class has less than two operators, the operation will execute immediately, and the `id` value of the generated `Proposal` record will be returned with the execution result, otherwise, that `Proposal` record will be stored in the `_proposalMap`, its `id` value will be added to the `_proposalList`, and the `id` value of the generated `Proposal` record will be returned with a `false` value indicating that the proposal was not immediately executed.

Subsequent calls to the same utility operation with identical input parameters will add the approval of the calling operator's address (if authorized) to the `approvers` list of the target `Proposal` record and the `_hasApproved` mapping for O(1) access. When an approval breaks the majority threshold of greater than half of the authorized operators of a given class, the proposal will be executed, the corresponding `Proposal` record will be deleted, and the `id` value of that `Proposal` record will be returned to the calling operator with a Boolean value representing the execution result of that proposal.

Operator Classes

`Prime, Arbiter, Dispatch, Freeze, Unfreeze, Lock, Unlock, Mint, Burn`

There are nine distinct operator classes in this contract that may be used to conduct various operations within the Collypto contract and this contract itself. A "management operation" can be defined as any operation that may only be conducted by an operator with a Prime key, and a "utility operation" can be defined as any operation to be executed in the Collypto contract that requires a key belonging to any of the other eight classes in the `OperatorClasses` enumeration. Operator authorizations are maintained in the `_authorizedOperatorMap` for O(1) retrieval, and the list of authorized operators for each operator class is maintained in the `_authorizedOperatorList` for aggregated reference and removal.

View Operations

`getCollyptoAddress, getProposalRecord, getActiveProposals, totalActiveProposals, getCurrentProposalIndex, getCurrentOperators`

This contract contains six view operations that may be conducted by an operator using any Ethereum account to retrieve proposal information, check the Collypto contract address, and view authorized operators by class.

Privileged Operations

`revokeApproval`

This contract contains a single privileged operation that may be conducted by an operator using an authorized account to revoke an approval on a proposal that was previously approved by that account.

Utility Operations

`updateUserStatus, forceTransfer, freeze, unfreeze, lock, unlock, mint, burn`

This contract contains eight utility operations that correspond to the utility operations of the Collypto contract. Each utility operation may only be conducted by an operator using an Ethereum account that is authorized as the operator class required for the specified operation.

Management Operations

`pause, unpause, addManager, removeManager, updateCollyptoAddress, removeProposal, purgeProposals, addOperator, removeOperator, purgeOperators`

This contract contains ten management operations that are used to maintain the address and running state of the Collypto contract and regulate operators and proposals within this contract. Management operations may only be conducted by an operator using an Ethereum account with Prime authorization.

Collypto Contract Management

All Collypto contract management and utility operations are multiplexed through the proposal system of this contract and directed upon execution to the address maintained in `_collyptoAddress`. In the event that we need to update the Collypto contract in a way that doesn't require a corresponding code change in this contract, we may update the Collypto reference address using the `updateCollyptoAddress` function. If we need to update this contract, the `addManager` function allows Prime operators to add a new management contract address to the Collypto contract, and the `removeManager` function allows Prime operators to remove the current management address of this contract from the Collypto manager list using the new management contract instance. The `pause` and `unpause` functions allow Prime operators to pause the running state of the Collypto contract and unpause it to respectively suspend and resume users' ability to conduct public Collypto operations.

Proposal Removal

There are two management functions that may be utilized by Prime operators to permanently delete proposals, removing their corresponding `Proposal` records from the `_proposalMap` and `_proposalList` (all proposals in those data structures are considered "active proposals"). The `removeProposal` function removes a single `Proposal` record with the `id` value provided, and the `purgeProposals` function removes all active proposals.

Operator Management

If we need to add one or more authorized operators to a given operator class, we can utilize the `addOperator` function to add individual address authorizations for that class. To revoke operator authorizations, we can utilize the `removeOperator` function with a provided operator class and address to remove the provided address from the authorization mappings, or we can utilize `purgeOperators` with a provided operator class to clear all authorizations for that class. Prime operators cannot be purged, and a majority compromise of Prime operators would mean that we would need to use the Master account to purge the manager list of the Collypto contract and update it to reference a new instance of this contract.

CollyptoValidator Contract

This contract allows any user to prove ownership of an Ethereum account by calling the `validate` function with a provided `code` string as the input parameter, and it allows any user to check the validation status of any account by calling the `isValidated` function with the account address provided as the `targetAddress`.

Since `validate` can only be called by the owner of the private key corresponding to the Ethereum account to be validated, and the `AccountValidated` event includes the account address and validation code, it is impossible for a user to validate ownership of an account that they do not own. Consequently, it is impossible for anyone to corrupt the `_validatedAddresses` mapping where validation status is maintained for all accounts.

The `validate` function can be repeated any number of times for a given Ethereum account, and an arbitrary validation code can be used to prove ownership of the account itself without requiring any off-chain functions or layer 2 applications. Validation codes are arbitrary, and their only purpose is to allow a user to demonstrate account ownership to another user or institution.

Users can check the validation status of any Ethereum account by calling the `isValidated` function and providing the account address as the `targetAddress` input parameter. This function returns a Boolean value indicating whether the corresponding account has been previously validated by the caller or another operator. Utilizing the `isValidated` function to check the validation status of a recipient Ethereum account prior to conducting transfers of any Ethereum-based token effectively eliminates the risk of mistyping a recipient address, provided its owner has already validated the account at least once using the `validate` function.

Immediate applications

Endowments

Endowments steward donations to help their respective non-profit organizations fund strategic initiatives. Most of the money is invested to ensure that funds are available for the organization in perpetuity. Funds that are allocated to upcoming or current projects must be readily available. Replacing the funds that are stored as cash and cash equivalents with Collypto Credits would enable endowments to satisfy their liquidity needs while eliminating a portion of their exposure to inflation risk. Securing funds for multi-year projects in credits, rather than dollars, eliminates the need for market and inflation-driven budget adjustments once the projects begin.

Retirement and Pension Funds

Retirement and pension fund administrators match their clients with the most appropriate mix of financial products to achieve their retirement goals. Collypto Credits would be a suitable addition to a client's portfolio as they get older and begin to place a higher value on principal preservation and pension payouts. Unlike bonds, the price of credits adjusts for inflation in real-time, providing greater cost predictability and allowing retirees to maintain a consistent standard of living, regardless of market conditions.

Insurance Companies

Insurance companies are required to have enough assets to satisfy their liabilities, plus an additional statutorily mandated surplus. While most of an insurance company's assets are held in bonds to provide inflation protection, some must be held in cash to meet claim obligations to their customers. Holding credits instead of cash for claim obligations effectively eliminates the risk of inflation while still providing a highly liquid, stable asset necessary to maintain compliance.

Municipalities

Local and regional municipalities have annual budgets to ensure the safe management of their jurisdictions and funding of capital projects. Capital project funds are usually held as cash or bonds, giving them direct exposure to inflation risk. Holding credits for capital projects would aid in controlling rising project costs, particularly in an inflationary environment.

Construction Projects

Major construction projects often span multiple years. Using credits to represent project funds that are not required for immediate expenses will serve to insulate the overall budget from market-driven cost changes. The use of credits also reduces the required cost contingency and helps to mitigate the risk of budget overruns.

Virtual Vaults

Accounts that utilize enhanced security features, like Virtual Cold Storage, provide users with an added layer of protection for their credits. Maintaining physical vaults with large cash deposits exposes the cash to inflation risk and poses security and safety concerns. Utilizing virtual vaults provides an inflation hedge for the funds and nullifies any attempt to take them by force.

Emergency Funds

Financial experts recommend that individuals and organizations maintain an emergency fund to navigate unexpected life circumstances. Emergency funds are not an investment and should remain highly liquid. Using credits for an emergency fund provides value preservation and safeguards it from inflation risk.

Risk Diversification

The risk parity model of stocks and bonds is commonly used to customize the risk profile of asset portfolios. While we anticipate the advent of credit-based bonds in the future, at present, the stability of credits makes them an ideal replacement for prime tranches of bonds whose objective is to minimize the impact of inflation.

Peer-to-Peer Transactions

The most fundamental application of Collypto is to function as a global equitable currency. Collypto was designed as a safe and effective medium for peer-to-peer transactions. For international payments, credits can be utilized to circumvent predatory intermediaries and fluctuating exchange rates. For domestic transactions, Collypto provides a stable pricing mechanism for goods and services and facilitates recourse in the event of theft or fraud.

Future applications

Validated Ethereum Transactions

To improve accessibility in non-medallioned countries, the Collypto Validator could easily be integrated into commonly used wallet applications, such as MetaMask. Confirming the validation status of a recipient address before any transfer operation would serve as a transactional safeguard between pseudonymous parties and effectively eliminate the risk of accidental transfers of Ethereum-based tokens. A sample flow of a successful validated transfer is shown below.

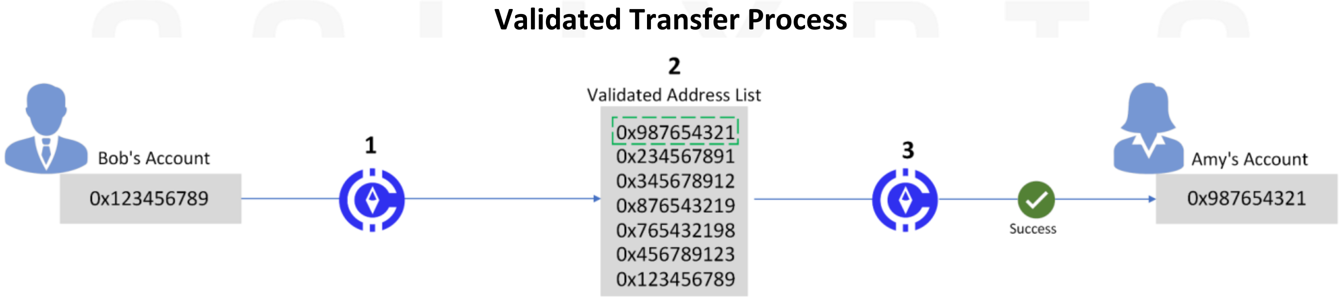

Validated Transfer Process

Figure 7

1. Bob submits a validated transfer request.
2. The Validated Address List is checked for Amy's address.
3. Since Amy's address is on the Validated Address List, the validated transfer is successful.

Automated Ether "Top Off" Subscriptions

As ether becomes commonly utilized to fuel credit transfers, there will be a need for users to maintain a consistent balance of ether to facilitate personal transactions. A savvy business could provide a subscription service that monitors Ethereum wallets and "tops them off" with ether when their balance falls below a certain threshold.

Blockchain Credit Score

The creation date of a user's Collypto Medallion is only reset in the event of a breach of their medallion account. Someone could theoretically reference a user's medallion age as a metric to determine the creditworthiness of that account and compute blockchain credit scores that could be utilized by third-party applications.

Medallion-Verified Scam Filter

Once Collypto has achieved widespread adoption in the general population, a layer 2 application could theoretically be integrated with a user's cellular service to provide a robust identity solution and authenticate inbound callers. Users and institutions would need to periodically verify medallion accounts with their cellular providers, but this system would reduce phone scams significantly.

The Collypto Standard

The crypto industry is infamous for its lack of oversight. The absence of consistent and comprehensive regulation serves as a double-edged sword to its participants. Common practices that have long been established in traditional financial markets are completely outlawed, while the lack of a codified legal framework has paradoxically allowed bad actors to engage in risky behavior, fraud, and unethical business practices. These all serve to undermine consumer confidence in the industry and throttle mainstream adoption.

One of our most important duties as a private reserve is to ensure that our organization never assumes the role of a government. It is essential that Collypto remains separate from any government entity so that we can hold each other accountable. Until our guarantees are legally codified, our terms of service will effectively provide a privatized form of substantive due process to our users predicated on anti-fraud laws, allowing us to self-regulate using the government as a proxy. This is not an acceptable long-term solution for any industry. As such, we will ultimately seek to translate our consumer protections into federal regulations, where applicable, while maintaining them in our company policies. We will support any regulations that we believe are necessary to protect our users, including full collateralization requirements, mandatory holdings reports, and increased enforcement against theft and fraud in the cryptocurrency space. We will also actively oppose any government action that we deem to be in bad faith or harmful to our users.

The Collypto Standard is divided into three directives, each containing articles with explicit applications to our governance and business processes. These articles will be directly incorporated into our terms of service, empowering both our internal team members and the public to hold us legally and morally accountable.

Maximize Accessibility

- We will design our systems, naming conventions, and processes to be as intuitive as possible for non-technical end users.
- We will never transfer digital assets to a third party without authoritative or legal recourse for recovery.
- We will provide internal recourse for verified users to recover misplaced or stolen funds, regardless of the amount.
- We will collateralize the Collypto Index exclusively with highly liquid, publicly traded assets that represent purchasing power.
- We will never charge fees to conduct user transactions.

Maintain Transparency

- We will provide users with a current and accurate representation of the collateral composition and equivalent dollar value of a Collypto Credit.
- We will publish a monthly Summary of Holdings on the first business day of every month that contains a complete list of our holdings, including the name, ticker, asset pool percentage, and dollar value of each asset.
- We will conduct and publish an annual third-party audit through a certified accounting firm to verify our stated holdings and credit compositions.
- We will never employ or retain services from anonymous parties.
- We will never promise a specific rate of return on credits.

Protect the Public Trust

- We will maintain 100% reserves in explicitly defined assets to collateralize circulated credits.
- We will utilize collateral assets exclusively to collateralize and circulate credits.
- We will never use collateral assets for other operating expenses.
- We will never loan collateral assets to third parties.
- We will never accept kickbacks for selecting specific assets or asset classes for inclusion in the Collypto Index.
- We will never use our blacklist capriciously. It will only contain addresses belonging to users who have violated our terms of service or are sanctioned by government mandate.
- We will never sell users’ private information to third parties
- We will maintain full insurance coverage on the entire asset pool.
- We will never tolerate, condone, support, or partake in terrorism or terrorist organizations.
- We will never assume the role of a government.

Conclusion

We have outlined the process used to create an algorithmically maintained store of intrinsic value and tokenize it on the Ethereum blockchain, creating the world's first functional flatcoin. We have utilized block indexing to preserve the exposure of our underlying collateral assets which track a meaningful index of allocations that accurately reflects purchasing power. Our Ethereum contract implementations follow industry standards with enhanced consumer protections that provide users recourse against fraud and theft. We have also introduced verification and validation mechanisms to systematically maximize the integrity of transactions of credits and other Ethereum-based tokens. Blockchain technology is an outstanding innovation for maintaining a decentralized transaction ledger. With the addition of a centralized store of intrinsic value and systems to prevent and indemnify users against fraud, it has allowed us to solve the problem of creating a global equitable currency.

Building Collypto has been the greatest challenge and professional accomplishment of our lives, and we are proud and excited to share it with the world. The flatcoin represents the next step in the evolution of currency, blockchain, and financial technology. We fully anticipate that, as long as humans use some form of money in the future, it will ultimately consist of a decentralized transaction ledger and a tokenized store of intrinsic value based on the fundamental assets of their time. We look forward to the day when our currency provides true equity of opportunity to anyone with an internet connection who wants to contribute value to the world. The mechanics of Collypto are complex, but the concepts that drive it are simple, and we will always make time to explain for those who sincerely wish to understand.

Works Cited

1. Cryptopedia Staff. (2021, April 6). *The State of Centralized Exchanges*. Gemini. https://www.gemini.com/cryptopedia/centralized-exchanges-crypto#section-trading-on-a-centralized-exchange

2. St. Louis Fed. (2022, Dec 8). *Consumer price index for all urban consumers: all items in U.S. city average (CPIAUCSL)*. Federal Reserve Bank of St, Louis. https://fred.stlouisfed.org/series/CPIAUCSL

3. Yue, Frances. (2022, Jan 13). *Will the crypto market always follow bitcoin's price lead? It may not in the future, this asset manager explains.* Distributed Ledger. https://www.marketwatch.com/story/different-crypto-will-be-less-correlated-as-healthcare-stocks-wont-move-in-the-same-way-gold-etf-moves-a-crypto-asset-manager-says-11642101628

Appendix

1. “Traditional cryptocurrency” refers to any currency derived from uncollateralized blockchain tokens whose supply is issued as a miner or validator reward.

2. “Cargo cult” refers to the misguided belief that performing rituals or mimicking specific behavioral patterns will yield a desired result. This phenomenon arises due to a fundamental lack of understanding of the contextual relationship between actions and their consequences. The most notable modern examples of cargo cults come from the Melanesian Islands during and after World War II. Islanders observed the activity of troops at airfields and noticed how the routines of tower and ground controllers resulted in supply drops. After the airfields were abandoned, islanders would sometimes mimic the behavior of the soldiers in an attempt to cause more supplies to be delivered to the island.

3. “Collateral decay” refers to the gradual degradation of underlying collateral behind ETF instruments due to their expense ratios. We account for collateral decay in the Collypto Algorithm using an allocation adjustment factor that is calculated in real-time by compounding reported expense ratios between the rebase and reference dates.

4. “Block discontinuity” refers to the deviation between the combined notional values of a set of futures contracts allocated to represent a product and the subsequent set of contracts that must replace it over a given rollover period. Contract allocations are optimized to minimize the block discontinuity of their respective futures products, but this factor cannot be entirely eliminated since futures contracts have defined expiration dates and are not contiguous financial instruments.