Better Than Money

Interfungible, asset-backed, transparent yet private

Money, as a universal medium of exchange, has world-changing advantages over directly bartering goods for other goods. By temporarily storing fungible value, money lets trades be flexibly split into parts that differ in amount of value as well as in space and time. A single national money has been able to provide stability of purchasing power, but only one-size-fits-all. It cannot provide the customized stability of barter, where each user has their own custom portfolio of assets. The new medium for universal exchange introduced here achieves the best of both. It attains the purchasing-power stability of custom portfolios through a temporary store of *interfungible* value. Payments drawn from the portfolio of assets held by the user paying are automatically disbursed into the portfolio of assets held by the user receiving the payment. Full asset backing, transparency of the books, privacy in payments, immediate irreversibility, and even investment by users of value previously un-investable, are all further advantages of this new kind of money—or perhaps it is not a kind of money at all but rather something *really* new. Let’s just call it *Better-Than-Money*!

The participants in the Better-Than-Money system are: an *issuer*, some commodity *providers*, and a number of *users*. The issuer creates an ephemeral medium of exchange, backed by a set of individually-fungible commodity-like assets, here called “commodities money” or *cMoney* for short; an advancement over single-asset so-called “commodity money,” such as gold-backed currencies, though here it only exists episodically as will be explained. The *i*’th commodity provider commits to managing the participation of their own specific one of the commodities, c*i*Money, in the set according to the rules of the system. The user’s smartphone app can be as convenient as any electronic payment app, while also offering an AI’s tips on customization of the portfolio. These suggestions could be based on what it knows of the user’s spending patterns and, for instance, on price change forecasts. The app and the user can be referred to nearly interchangeably as if they are the same entity. In fact, these advantages have been enabled by new cryptographic protocols and app platforms that can securely conduct the protocols for their users.

Users of Better-Than-Money, or *BTM* here for short, select their own allocation of commodities, all from the same set. And the “commodities” underlying a BTM system can be anything that is itself fungible and for which custody arrangements are practical: national currencies, whatever traditional physical commodities, anything offered today as an exchange-traded fund, or even cryptocurrencies. But different users of BTM may employ the same commodities differently. If some commodities are national currencies, for instance, they can provide the stability of a so-called “basket” of currencies for some users, or alternatively hedge anticipated expenses abroad for other users. Even though users of BTM hold only non-delivery ownership, food commodities can protect them against grocery price changes and energy commodities against utility rate changes. This is because successful hedging, like successfully investing, provides consumers with additional spendable value. Users can be individuals or businesses.

Users can choose portfolios and strategies easily. For instance, social investing allows one to simply copy a portfolio template from all manner of other people. But tips from AI can improve strategy and customization to an effectiveness exceeding that of the most sophisticated investors today. Also, offerings themselves will become easier to use, since the more relevant or attractive a particular asset is for users, the greater the likely share it will attain, and the more rewarding the economics for the respective commodity provider. (See figure 1.)

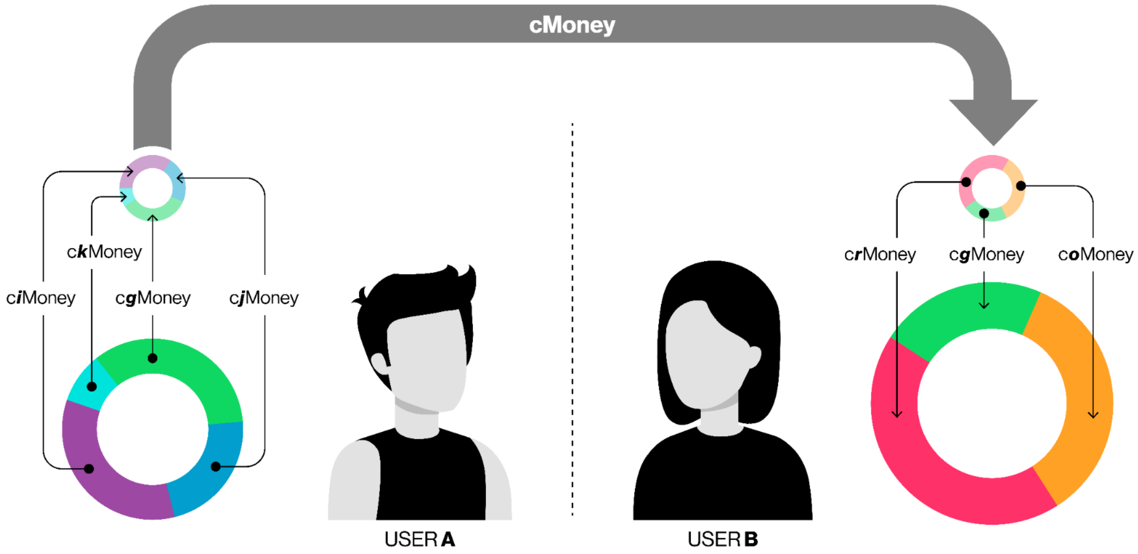

FIGURE 1: User "A" pays user "B" an amount of value, which is represented by the size of the two small pie-chart circles. Sometimes the two users may think of that value as denominated in the same currency or they may each regard it as denominated differently. No matter how they think of the value transferred, however, it is represented by the size of the small pie charts, which are equal in size since the two must represent an identical amount of value. The value is taken from the portfolio of A, in the proportions of the four assets in that portfolio; the value is disbursed to B, in the assets and proportions of B's three-commodity portfolio. Transactions occur only in a single standardized but brief "window" of time, in which assets in the portfolio of A are converted briefly to a calculated amount of cMoney and then converted back to increase the assets of B. The small faded pie charts simply illustrate the ephemeral value being transferred, while the larger pie charts represent the enduring asset portfolios of the two parties affected by the BTM transaction.

Better for conducting transactions? Users of BTM have freedom to choose whatever asset or portfolio of assets they wish to hold for purposes of trade. Contrast this with needing to hold a particular money, which is a kind of debt that is non-interest bearing, inflationary, and subject to default, from a central or commercial bank. For one thing, economic and social stability are enhanced by how BTM allows the risks to be spread across the assets backing particular transactions—attacks on or failures of national currencies, or runs on banks or delayed settlement systems, could no longer devastate. Such liquidity crises are becoming dramatically more likely and potentially catastrophic as the winds in both the environment and social media grow stronger. BTM also helps address income inequality, by allowing a broader class of users to access efficient payments and to accumulate wealth while participating in growing economic prosperity.

The ETF or "Exchange Traded Fund" was created by Blackrock two decades ago to give ordinary equity market investors exposure to a wider range of investments. Today roughly ten thousand ETFs represent ten trillion USD in assets under management. Such assets and custody arrangements would be well suited to serving as commodities here. This also means plenty of on-ramps and off-ramps for users to choose between, since each commodity provider can offer both to users. Participants in the informal economy and the unbanked, as well as those straddling multiple national economies and involved in remittances, can easily and immediately obtain benefits. With BTM, transfers can be nearly instant, inexpensive, and private. Moreover, removing the friction of 2% of global GDP in fees for payments by the poor would allow the global economy to flourish and all boats to rise.

Better for loans and payments over time? Defining the value of a loan in terms of a mutually-agreed custom basket of commodities, which is easily accomplished with BTM's platform, provides stability for lenders and borrowers alike, in effect a widely usable and customizable version of the World Bank's SDR fixed currency basket. Debt becomes more attractive, robust, and efficient—facilitating economic growth. Such mutually-custom baskets, more generally, would work similarly well for all manner of prearranged payments. Examples include leases, insurance, annuities, other payments under contract, and supply chain.

Better for investments? What was only possible in principle with very large values would now be available through BTM to everyone for everyday use. This is because friction in investing is reduced even beyond lower fees, by removing restrictions and ensuring extensive interoperability. Of course, a simple BTM transaction could be roughly duplicated in effect, though only when individual amounts are large enough, by a series of *a la carte* transactions in today's systems. However, the additional "hops" would imply greater implicit and explicit fees, delays, multiplicity of interface idiosyncrasies, and even risks. Also, enhancing input to society's investments can be a good thing. Moreover, rebalancing of portfolios, albeit gradually, to new ratios and even new selections of commodities, can in effect be cost-free to users, since direct and indirect costs are already being borne by payment transactions that can yield the changes as side effects.

The "on-ramps" to the system are simple. A user can purchase c*i*Money from provider *i* by for instance transferring a sum of fiat money, via whatever means outside the BTM system, to that provider. In return, the provider in effect mints and then transfers to that user the purchased c*i*Money, representing the same value but now effectively transformed into the commodity. The provider holds the asset in custody, ideally with the user as the beneficial owner in case of provider default, which is possible since the c*i*Money issued is a claim on uniquely represented value. Similarly, any commodity provider can also serve as an "off-ramp." For instance, when a user signs over some c*i*Money to provider *i* that provider can in exchange return, outside of the BTM system, the same value to the user for instance in fiat currency.

If a user were to receive cMoney, the user's app would divide it among the phone's selected commodity providers in anticipation of future purchases or investment returns. Similarly, when a user wishes to make a purchase, the user app could in effect itself sell portions of its commodities back to their respective providers in exchange for the cMoney the app would then pay out in the purchase. To accomplish these divisions or re-combinations, users would in effect be purchasing c*i*Money with cMoney or obtaining cMoney by returning c*i*Money to its respective provider. This is conceptually how BTM uses the ephemeral “singleness” of cMoney to create interfungibility between multiple assets.

Actually, users need never hold cMoney. The sale of assets back to the respective providers can be orchestrated by the issuer, who would in effect immediately use the cMoney obtained to purchase assets on the payee’s behalf, in the proportions the payee had initially encrypted into its request for payment. (See figure 2 and the Appendix for more detail.)

Implementation of the BTM system is assumed here to be based on eCash 2.0 (see chaum.com/ecash-2-0 and xx.network). Each provider operates its own eCash 2.0 system for the token of its commodity, and the issuer does likewise for cMoney. All the tokens are publicly-visible, yielding transparency of where all value is in the system at any time. The public result of transactions between cMoney and c*i*Money, and changes in the amount of a commodity held in the system by providers, is thus made transparent. The tokens may look like they are merely large random numbers; but they are actually encrypted *images* resulting from the application of a cryptographic *one-way function*, sometimes called a “hash function.” The initially secret corresponding input to each hash function output is called its *preimage*. (For simplicity, payments will often be described here as transfers of value, leaving implicit details of the underlying denominations and multiplicities of coins.) Each token is associated with the issuer or with a respective provider, and the changes in the published tokens are verifiably in accordance with the stipulated price in cMoney of the respective c*i*Money commodity at the instant of the transfer. (This will be described in more detail with reference to figure 3.)

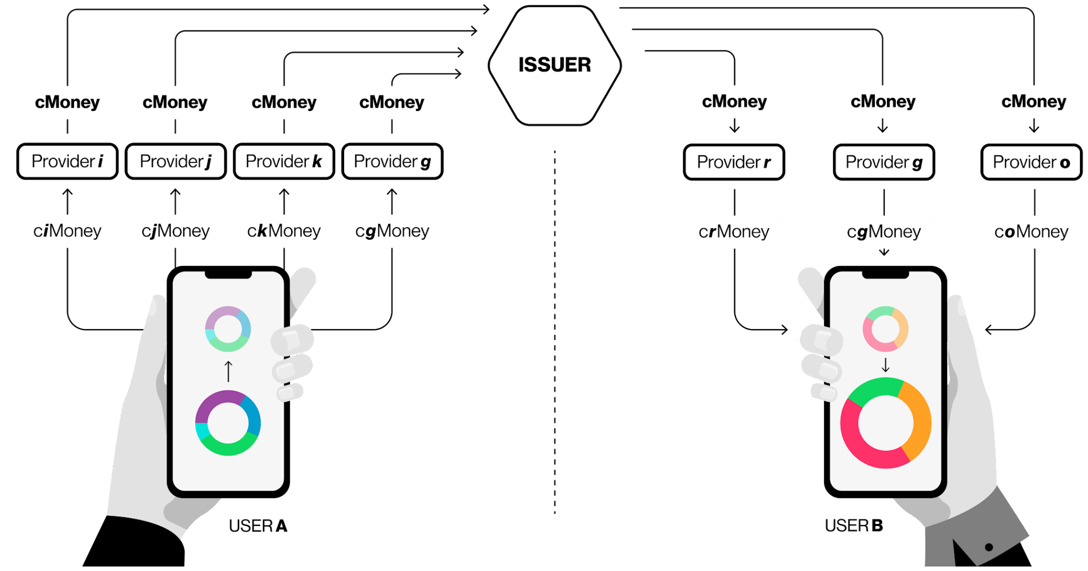

FIGURE 2: A payment by user A's phone to user B's phone first sources cMoney by liquidating a proportionate amount of user A's portfolio at the respective providers. Next, the issuer disburses this to the providers in the proportions requested by B's phone. These providers then in effect make this value available to B's phone for future transactions.

What can here be called *buffers* allow the issuer to ensure essentially immediate *irreversibility*, sometimes called finality of payments, without having to wait for any provider. To facilitate this, providers can be obligated by contract to hold some c*i*Money as a kind of "commodity buffer." The amount required to be held can be stipulated, say, as an upper and lower percentage of the value of the total amount of the commodity held by that provider. So, for instance, a minimum of 7% and a maximum of 15% of total holding to be maintained as a commodity buffer in the respective currency could be contractual for a provider with a prescribed volume. Conformance with the buffer formula would be publicly verifiable using the value of tokens in the commodity buffers and respective commodity pools. Providers ideally would publish evidence of holdings, making the extent of commodity buffer holdings publicly verifiable. Similarly, each commodity provider would have a "liquidity buffer" comprising cMoney tokens, and conforming to similar rules. These two buffer types let irreversible transactions be quickly committed by the issuer, without needing the provider's immediate participation. A user-issued sell order, for instance, should cause the respective provider to give the user, in cMoney, the value of its commodity at the current rate. This would, as mentioned, typically be done by the issuer as part of a transfer. The way such a sale works, in essence, is that the user receives the cMoney immediately from the issuer, covered by a hold on value in the provider's liquidity buffer; this hold remains until the provider settles with the issuer. More specifically, a temporary lock is placed on the c*i*Money sold until control over it as well as title to it is transferred from the provider's token pool, for instance by transfer to the provider's commodity buffer.

Similarly, a user-issued buy order, which would be paid for in cMoney, should cause the provider to transfer like value in c*i*Money to the user. Typically, this would be the second part of the transfer to a payee orchestrated by the issuer. The way this works, in essence, is that the provider is covered immediately by value inserted into its liquidity buffer by the issuer; but that value is locked until released by the purchasing user. More specifically, the user exclusively has the key to a lock placed on c*i*Money in the provider's liquidity and commodity buffers, for the same amount as purchased. When the provider then delivers to the user an acceptably perfected ownership in the purchased c*i*Money token(s) visible in the provider's token pool, also locked with the same key, the user releases the key and all the locks are simultaneously opened by that one key. (Both types of orders are further detailed in the appendix.)

Prices of commodities are crucial to system operation and must reflect actual global market prices. If any trader could use the better-than-money system for arbitrage or spot transactions, this might overwhelm the system and at least potentially interfere with its medium-of-exchange function. To avoid such problems, prices would be somewhat less attractive than on spot markets, or value-dependent direct fees would be somewhat higher. In case of a sudden large change in price that might cause a flood of traders to use the system, finite buffers can be a help. Payments can also be limited and can even be shut down temporarily, as in some current markets, when there is such a price change or if no reliable price information is available..

To realize the technical system in practice, time is discretized and divided into a series of small non-overlapping time “windows.” In each brief window, prices for all commodities are fixed in cMoney. These prices are stipulated in each provider’s agreement with the system based on specific formulas and market prices for each commodity, sampled ideally near the end of the previous time window. Setting the prices and any fees in this way should also let the books balance during the instant of each window.

A basic BTM system can then be realized with pairs of cooperating transactions conducted in the same time window. Value from one transaction can atomically be a part of another paired interfungible transaction, where sell orders by a user become irreversible by being tied to paired buy orders from a second user (as illustrated in Figure 2). Again, payments leaving a portfolio are translated first to cMoney and then to withdrawals deposited to the recipient’s portfolio—all in a single time window.

Buffers will require refilling, and asset pools may grow or shrink. When a provider receives a preponderance of buy orders for its c*i*Money, it will eventually have to obtain more of the asset, such as by purchasing on the open market, presumably using value obtained from the cMoney it has received. To off-ramp this value for purchase of more assets on the open market, a provider can, for instance, purchase tokens with cMoney from another provider in the system and then sell these back to that provider on the open market outside the system. Similarly, when sell orders dominate for an extended period, the provider would presumably reduce the amount of the commodity it holds in custody by, for instance, selling it on the open market and on-ramp the proceeds to obtain cMoney for its liquidity buffer. Resulting changes to the size of the respective commodity buffers would be publicly visible and would affect the total amount of value in the system. Providers should, however, be able to keep a large enough distance between actual changing buffer balances and the stipulated limits so that user orders can always be handled instantly. This would mean enough liquidity buffer cMoney to meet expected sell orders and enough commodity buffer to meet buy orders.

There are various incentives for commodity providers to join and participate in the system. For one, there would be buy/sell spread income in the contractual formulas that determine the price of their commodity in cMoney, based on a contractually defined market price rule. For another, there can be explicit fees due providers, such as per transaction. Some sort of fee split with the issuer seems natural and could be enough to support the issuer function. Issuer support can also be via the proceeds of auctioning commodity-provider slots and even the commodity-provider analog of anchor store deals for malls. Such auctions can set the rule parameters and establish each winning bidder as the exclusive provider of their respective commodity.

The system can be decentralized, with competing parts spurring overall growth. A natural split is between unregulated and regulated assets types, such as separate providers for blockchain and traditional assets. However, the software on your phone could seamlessly hide such a split from you. It’s easy to see that there can also be competing providers for the same asset. But further multiplicities are also possible, such as multiple competing issuers, though this might mean duplication of at least liquidity buffers. Thus there are multiple independent but ultimately complementary growth paths. Moreover, the maturity of the eCash 2.0 technology and asset provision arrangements, such as those used by ETFs, mean that the financial and technical building blocks are already proven. Thus better-than-money does not rely on any hard to achieve assumption and has robust growth opportunities, making it an attractive way to do well by doing good.

For greater efficiency in smaller transactions, fee structure can incentivize users to accept use of smaller subsets of their selected assets. But these subsets can be designed to vary so that ultimately, across many transactions, the user's whole portfolio converges towards the desired proportions. (Similarly, rounding, that is cryptographically protected against being unfairly gamed by either side of a transaction, can reduce the precision—and hence the number of coin denominations needed in low-value payments—yet keep the results converging towards exact values.) Moreover, by selecting a portfolio for use in outgoing payments different from that for incoming payments, users can gradually morph their complete investment portfolio as mentioned above, while sharing the transaction costs with transfers that would be made regardless. The target of the morphing could even be changed dynamically.

The issuer must allow cMoney to transfer between providers so that they can maintain their respective liquidity buffers; however, it could also even allow anyone to transfer cMoney as a kind of "supracurrency." A person receiving cMoney in payment could in turn then pay that cMoney to a second person. This could in theory be repeated a number of times, forming a sequence of person-to-person transfers. The fact that the cMoney simply travels along such a sequence, without changing the total value in the system, would be readily visible as rolling entries in the issuer's cMoney token pool. When someone in the sequence wishes to hold the value they received in this way for some period, they can be expected to convert it to assets they wish to hold through the respective providers, rather than just becoming a further link in that cMoney chain. This last person in the sequence has then, at least in effect, been on-boarded as n BTM user.

The price of cMoney, relative to whatever commodity basket, can be set by the issuer to be inflationary, deflationary, or even essentially stable. This is because cMoney only has to have a fixed value episodically, during each time window, but those values are, as astrophysicists say, “relative to the fixed stars”—that is, relative to whatever basket of assets is believed in. To disincentivize the holding of cMoney, the issuer could make it inflationary; to incentivize the holding of cMoney, the issuer could make it deflationary; and to incentivize its use in trade, its value could be set relative to some basket. But setting it relative to a fixed basket would in some sense defeat the elegance cMoney obtains by having no particular fixed basket but rather only an episodic ratiometric one. However, to avoid arbitrage when cMoney is allowed to trade separately, its value would need to reflect the amount issued and the value of assets committed to the system.

Risks include custody risk and the solvency and technical robustness of the BTM system’s entities. Provider agreements could address these risks by requiring that: (a) users have title to specific assets within the commodity represented by their c*i*Money; (b) titles to commodity buffers are in effect joint between the issuer and the respective commodity provider; and (c) the total amount of issuer cMoney outstanding never exceeds the total combined unused portions of commodity buffers.

Conferring auditable title interest in the underlying commodity or pool portion to users, as required by (a), is accomplished today differently according to national legal frameworks. If a provider were to default, or its technical systems were to fail, there might be exposure to loss for transactions that have been initiated but have not yet perfected user ownership in the underlying commodity. Since reduction in commodity buffer size would as mentioned only be triggered for perfected transactions, the issuer could always take ownership of the commodity buffer portions involved, by (b), and ultimately either perfect user ownership or liquidate the assets to refund users. Finally, if the issuer ceases operation or defaults, the commodity providers backing the system take possession of the commodity buffers, which are adequate by (c), and liquidate portions in some stipulated pro rata manner to make users “whole” by adequate repayment.

Unlinkability of transactions might seem at odds with an ability to audit a system for financial soundness. The system presented here, however, provides both at the same time. System users can be afforded unlinkability of transactions from payer to payee, at least within the anonymity sets of all like transactions, as provided by eCash 2.0 (as will be detailed further in the appendix). Nevertheless, the transparency of posted data, including all commodity prices in cMony at each window and total number of tokens in the buffers and pools, ensures that the system's soundness is publicly auditable in aggregate at each instant.

When commodity providers issue their own c*i*Money to users, the aggregate total amount outstanding per provider is public because the tokens in the respective buffers and pools are publicly visible. The use of eCash 2.0 technology for all coins issued in the system, already mentioned, ensures however that each coin is unlinkable to its transaction of origin. This results from eCash 2.0's "one-way" privacy property: when the cMoney or c*i*Money is spent, the payer can always reveal the payee, but the value remains unlinkable to the transaction in which it was obtained by the payer. And when a payment is to be made to a user in c*i*Money, actually an anonymous withdrawal is made at the expense of the payer but with beneficiary key supplied by the payee in encrypted form to the payer in the request for payment.

Appendix—cMoney Transfer Mechanics Detailed

Speed in payments can be measured in delay to irrevocability. If the issuer can bring transactions to irrevocability quickly enough, which means in general without the need to wait for providers, it can combine, into a single transaction window, the transfers from the paying user with the resulting transfers to the paid user. In this way the system attains the goal of user-to-user transfers across commodities. Providers can then take more leisurely and varied times, the need for which can differ significantly between commodities, to perfect transfer of ownership interest. Other advantages of unilateral transaction consummation by the issuer relate to scalability and cost. Since high-availability and high-speed digital presence need not be replicated, considerable economies of scale result. Moreover, if providers could cause delay of irrevocability, the overall worst-case response rate of the system would be that of the slowest provider. All potential provider-caused delay is avoided by issuer-locking of transaction details. Providers are solely involved in the non-time-critical settlement of their transactions, which are unlocked only once everything is verified by the other parties to the particular transaction.

The issuer can create irreversibility simply by being able to: issue new cMoney to users; “lock” and “unlock” portions of any provider’s liquidity buffer or commodity buffer; and “inject” fresh cMoney into slots of a provider’s liquidity buffer. Locking and unlocking is simply realized by a publicly visible online posting, exclusively under the control of the issuer, who chooses the preimages and then posts their images. These images are “unlocked” by posting the corresponding preimage, which anyone can then easily verify as yielding the locking image. (See now Figure 3.)

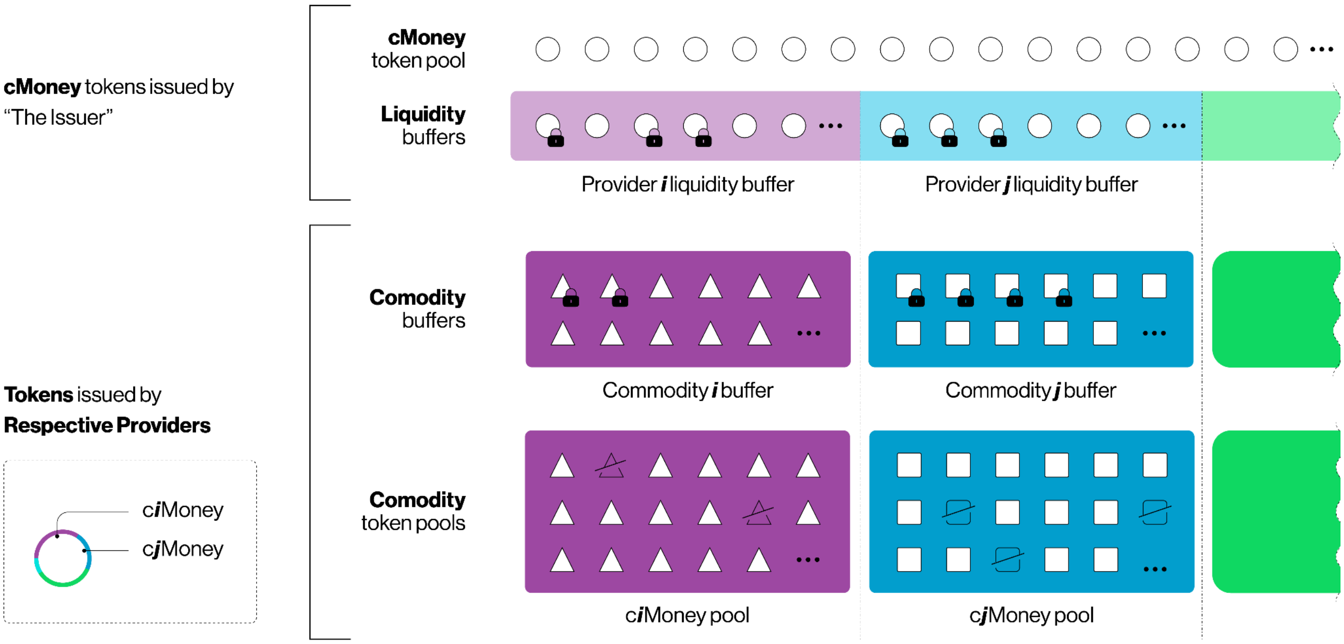

FIGURE 3: The better-than-money system is based on publicly-posted cryptographic tokens, illustrated as small white-filled shapes. Along the top of the figure is the pool of cMoney tokens (round), signed and posted by the issuer, which are divided into a single cMoney pool and separate liquidity buffers for each respective provider below (indicated by dot-dash border lines and background colors and ellipsis for those not shown). Below the liquidity buffers are the tokens issued and posted by each respective provider: its commodity buffer and its commodity token pool. Also illustrated is the issuer's ability to lock buffer value and mark pool items (as canceled with a stripe).

Scalability of transaction processing is greatly facilitated by the uniqueness of cryptographic tokens and the locking by hash images, which underlie system security. These allow processing by the issuer to be divided not only between commodities but further subdivided between ranges within respective buffers and pools (sometimes called sharding or vertical data partitioning). The issuer processes each user-to-user transaction with one of its own (horizontally-scaled) transaction engine instances. These communicate with each one of the issuer's respective buffer and pool processes involved, which do the actual locking, unlocking, and canceling of images related to respective providers. So that buffer size-minimums can be monitored in real time, providers are required to give digitally signed notice to the issuer sufficiently in advance of whenever they reduce the value of their buffers or increase that of their pools. Such true scalability not only ensures that the issuer can complete a large number of combined transactions within each discrete time window, but also lowers overall latency delay to irrevocability.

When a user in effect returns c*i*Money to its provider in exchange for cMoney (a sell), the issuer posts, in the next available window, four things: (i) signatures validating cMoney images freshly added to the cMoney pool by the issuer, which crucially can be used to make a related payment to a counterparty; (ii) images that lock a corresponding amount of cMoney in the provider's liquidity buffer; (iii) images that lock an amount of c*i*Money in the provider's commodity buffer; and (iv) a cancel mark on the particular value in the commodity pool that was sold. Locks are performed by posting cryptographic hash images, and these locks have a single unlocking preimage known to the issuer. Once the provider has moved the canceled c*i*Money from its commodity pool to its commodity buffer and has signed over the cMoney in its liquidity buffer to the issuer, the provider requests posting of the unlocking preimage, ideally before penalties accrue. The final result is that liquidity has been made immediately available to the selling user and the provider has paid cMoney to the issuer and moved the returned commodity from its pool. (See left side of figure 4.)

When a user makes a payment in cMoney for purchase of c*j*Money (a "buy"), the issuer independently posts two things during the first available time window: a locked injection of fresh corresponding value into the liquidity buffer of provider *j*; and a lock on the corresponding amount in the commodity buffer of provider *j*. This time, the preimages for the locking images are known to the purchasing user. The injection is by signatures that give value to empty slots in the provider's liquidity buffer. Once the provider has perfected the purchaser's custody of commensurate freshly created c*i*Money, then the purchaser should post the single unlocking preimage (penalties accrue if either perfection of the title change or the unlocking request are unreasonably delayed). The final result is that the purchaser has accepted the title to the c*j*Money as conclusively-perfected, while the provider has in its buffer the cMoney that was paid for that amount of commodity *j*. (See right side of figure 4.)

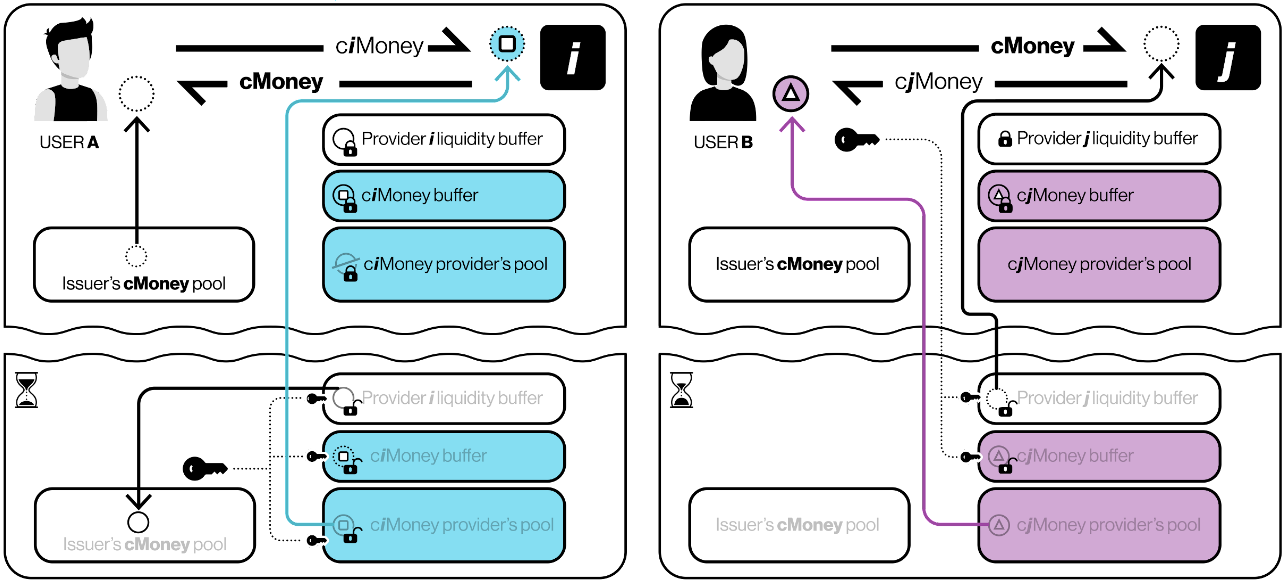
c*i*Money
cMoney
i
USER **A**
Provider *i* liquidity buffer
c*i*Money buffer
c*i*Money provider's pool
Issuer's **cMoney** pool
Provider *i* liquidity buffer
c*i*Money buffer
c*i*Money provider's pool
Issuer's **cMoney** pool
cMoney
c*j*Money
j
USER **B**
Provider *j* liquidity buffer
c*j*Money buffer
c*j*Money provider's pool
Issuer's **cMoney** pool
Provider *j* liquidity buffer
c*j*Money buffer
c*j*Money provider's pool
Issuer's **cMoney** pool

FIGURE 4: Two separate transactions are shown across the top: user A obtains cMoney by selling c*i*Money back to provider *i*; and user B buys c*j*Money from provider *j*. But if A asks the issuer to use the resulting cMoney to directly fund B's requested purchase of c*j*Money—a single transaction window in effect transfers A's c*i*Money to B's c*j*Money. The issuer immediately makee both the sale by A and the purchase by B irrevocable without contacting either provider. This is done by marking the c*i*Money as spent, taking cMoney from the provider *i*'s liquidity buffer and injecting this cMoney into provider *j*'s liquidity buffer (though the rest the transaction is locked). The locked parts are then unlocked, each side potentially with its own delay, and the transactions are then fully completed when unlocked as indicated in the respective lower portions of the figure.

Unlinkability can be achieved between the payment that originally funded a token in a pool and the spending of that token from that pool. The original information defining the destination of the payment, presumably communicated user-to-user as part of the request for payment, is actually an encrypted but typical eCash 2.0 withdrawal request. This includes the image for the eventual token to be signed and posted, but in a form that remains hidden while mixed in batches through multiple nodes until it is ultimately revealed during spending. It also includes a blinded form of the same image, again not revealing the image, that will confer ownership of the image when it is revealed in signed form during spending. The funding and the actual image spent, which has taken an empty committed slot in the pool, are thus unlinkable.