In Blockchain We Trust?

Martin Glazier

Abstract

Blockchain is the distributed ledger technology underlying bitcoin and other cryptocurrencies. Further applications of the technology are now being explored by corporations and governments the world over. Although it has often been suggested that blockchain's distinctive potential as a technology lies in its connection to trust, the precise nature of the connection remains obscure. I aim to clarify it. I argue that blockchain is a 'trust transformer': it turns a generalized trust in a community into a particularized trust in a certain database, one collectively produced by the community in a special way. Different applications of blockchain, however, generate different levels of particularized trust in their associated databases. Some applications generate trust only in the integrity of the database. Other applications generate trust not just in the database's integrity but in its accuracy too. Which of these categories an application falls into depends on how the facts memorialized in its database are metaphysically grounded.

At the heart of bitcoin and other cryptocurrencies lies the 'distributed ledger' technology known as blockchain. The technology's promise has spurred interest and investment from corporations and governments around the globe, many of whom have explored applications of blockchain far beyond its original role in cryptocurrency.

But what exactly *is* the promise or distinctive potential of blockchain? Many have suggested that the answer to this question lies in blockchain's connection to *trust*. But they have had conflicting views of the nature of this connection.

Satoshi Nakamoto (2008), the pseudonymous inventor of blockchain (and bitcoin), sees the technology as, in a sense, opposed to trust. He took himself to have 'proposed a system for electronic transactions without relying on trust' (8). For Nakamoto, blockchain renders trust unnecessary.

But others have seen things differently. They have stressed blockchain's ability, not to get rid of trust, but to create it. Writing in the *New York Times*, the financial journalist Andrew Ross Sorkin (2018) declared:

> The blockchain is ultimately about solving society's ultimate challenge: trust. Or rather, lack of trust. Blockchain is about using technology to create a shared sense of trust by a group of disparate participants.

The present paper aims to clarify the relationship between blockchain and trust.

Philosophical discussions of trust have tended to focus on trust in individuals. But a proper account of blockchain's connection to trust must appeal to two non-individual forms of trust. The first is a kind of generalized trust in a community as a whole that does not require trust in any particular individual.[1] The second is a particularized trust in a certain database, one collectively produced by a community in a special way.

Blockchain, I will argue, is a 'trust transformer'. It turns the first, generalized form of trust into the second, particularized form of trust. Different applications of blockchain, however, generate different levels of particularized trust in their associated databases. Some applications

[1] Hawley (2017) offers one of the few philosophical discussions of trust in groups.

generate trust only in the *integrity* of the database. Other applications generate trust not just in the database's integrity but in its *accuracy* too. Which of these categories an application falls into depends, we will see, on how the facts memorialized in its database are metaphysically grounded.

§1 Integrity

Before we can understand why blockchain is a trust transformer, we have to understand what it is in the first place.[2]

Blockchain is sometimes portrayed as a quintessential digital-age technology, inseparable from the internet and impossible without advanced cryptography. But in fact the technology is not a distinctively digital one. It is simply a method, or system, for preserving a sequence of data. This system allows a community to collectively create and maintain a database or 'ledger' that no individual member of the community has the power to alter. Such a system *can* be implemented digitally, but it can also be implemented in other ways. To gain a clear understanding of how the system works, we will postpone discussion of computers and cryptography until later in the paper.

We will begin by examining a non-blockchain system. This system is simple but requires a problematic form of trust from its users. By modifying the system so that it no longer requires this form of trust, we will arrive at a blockchain system.

Imagine a community of collectors who acquire and trade rare books. One's status in the community depends on how impressive one's collection is. And so each member of this community is concerned to prove to the others which books she has. How can she do that?

[2] The locus classicus of blockchain is Nakamoto (2008). A useful recent presentation, containing much more detail than the present paper, is found in chapter 2 of Bailey et al. (2024).

Here is one way. Any time a member of the community wants to prove she has a given book, she can go door-to-door showing it off to everyone. But these books, let us suppose, are so old and so fragile that they cannot often be exposed to light. The community needs another way.

So let us suppose that the community instead creates a *ledger* to record which books everyone has. A special, one-time-only conference is held, and a consensus is reached on who has which books. The consensus is memorialized in the ledger and entrusted to an *administrator* for safekeeping.

However, these antiquarians do not simply sit on their collections. They trade with each other. And so they want to keep the ledger up to date, reflecting the current state of each member's collection. Here is one way that can be achieved. Suppose Alice transfers a certain rare book—a Gutenberg Bible, say—to Bob. When the transfer occurs, Alice writes on a slip of paper 'Alice transfers the Gutenberg Bible to Bob'. Then she signs it (in an unforgeable way) and submits it to the administrator of the ledger.

The administrator now checks the ledger to ensure that it satisfies one of two conditions: either (a) the ledger says that Alice had the Gutenberg at the one-time-only conference and *does not* contain a statement saying that she transferred it to someone else, or (b) the ledger contains a statement saying that someone transferred the Gutenberg to Alice and no *later* statement saying that she transferred it to someone else. If Alice's statement meets either of those conditions, the administrator deems it *valid* and adds it to the ledger. Otherwise it is deemed *invalid*.

If all goes well, this system will keep the ledger up to date. But it requires the community to trust the ledger administrator, and one might worry that that trust is misplaced. For example, the administrator might overlook one of the statements submitted to her, or she might misjudge one as valid when it is really invalid.

These shortcomings can be mitigated by making the ledger and all submitted statements publicly viewable. If the whole community can see the ledger and all the statements that have been submitted, then each community member can verify for herself that the administrator has been careful and thorough. Each person can verify, that is, that all and only the valid submitted statements have been added to the ledger.

But the system remains open to abuse. Suppose that the administrator of the ledger is Alice herself. And suppose that, after transferring the Gutenberg Bible to Bob, Alice decides to boost the status of her friend Carol by making it appear that *she* has the Gutenberg. She can write down on a slip of paper 'Alice transfers the Gutenberg Bible to Carol', sign it, and submit it to the ledger administrator—that is, to herself. In the ledger, Alice can then *replace* the earlier statement, the one that says she transfers the Gutenberg to Bob, with this new fraudulent statement.

If Alice does this devious deed, most members of the community will never know. Of course, any of them may inspect the ledger, and whoever does will see that the submitted Alice-Bob statement is not included in it. But this omission will appear entirely appropriate. For by the standards of the fraudulently altered ledger, the Alice-Bob statement is invalid. After all, according to the altered ledger, Alice had already transferred the Gutenberg to Carol!

In view of its potential for abuse, it is reasonable for the community to adopt this system only if it trusts the ledger administrator. Now there might be a member of the community who is widely thought to be so upstanding as to be beyond suspicion, in which case the system might well be fine. But there also might not be any such person. If the community is sufficiently large there might not be any member who is even *known* to all the others, let alone trusted by them. It is worth considering alternatives.

If the community wants to avoid trusting any *one* person to properly administer the ledger, it can make the ledger not only publicly viewable, but publicly administered. Rather than leaving the sole copy of the ledger in the hands of a single person, the community can distribute copies of it to everyone. Whoever wishes to submit a new transfer statement can then simply circulate a copy of it to everyone in the community. Under this 'distributed' system, everyone is expected to check each new submission against her own ledger to judge whether it is valid or invalid. If a submission is valid it goes in; if not, not.

But inevitably, not everyone will be a perfect administrator of her own ledger. Someone will lose a statement that was circulated to her, or will judge a statement to be valid when it is really invalid, and so on. And this will lead to problems.

Suppose Bob wants to impress another antiquarian, David, by proving to him that he has the Gutenberg Bible. Short of displaying the book itself, what can he do? He can ask David to consult his ledger, but David might have lost, or never received, the statement that says that Alice transferred the Gutenberg to Bob. In that case his ledger will not show that Bob has the Gutenberg. Bob can also show David his own ledger, but David may not trust Bob to have properly administered it.

Bob might instead try to appeal to the community at large. He might ask each member of the community to show her ledger to David. Because the community may contain delinquents who have not properly administered their ledgers, it may be that not *every* ledger will show that Bob possesses the Gutenberg. Still, Bob might think, the *majority* of ledgers will show this. Surely, he might argue, most members of the community are responsible, and so whatever the majority of ledgers show will reflect whatever transfer statements have in fact been circulated.

But even if most members of the community are responsible, that does not mean that the majority of ledgers will reflect the statements that have been circulated. After all, the community might contain malicious actors like Alice. Suppose that, as before, Alice alters her ledger to include a fraudulent statement to the effect that she transferred the Gutenberg to Carol. She might then create many copies of this fraudulent ledger—so many copies that the majority of ledgers in the community end up being fraudulent.

Her scheme could be blocked *if* there were a way to enforce a 'one person, one ledger' rule. But this might not be possible, especially if the community interacts largely online. After all, Alice could simply create many accounts, each one claiming to be an independent member of the community—one whose ledger conveniently agrees with Alice's own.

In a way, this system is even worse than the previous one. The earlier system required the community to trust one person to properly administer the ledger. This new system no longer requires *that* kind of trust, but the community must now trust each of its members not to run a scheme like Alice's—something that was impossible under the old system.

There is, however, a way to remove the need for this trust. We start with the distributed ledger system just described. But we now require every ledger to satisfy two additional conditions.

1. If any statement in a ledger is altered, that automatically destroys all the subsequent statements.
2. Adding a statement to a ledger is likely to take a long time.

These two conditions lie at the core of blockchain.

But how could these conditions ever be enforced? Set this question aside for the moment, and just suppose that the conditions hold, so that we can see why anyone would want to impose them in the first place.

Imagine that Alice and Bob have identical ledgers consisting of 100 transfer statements. Suppose statement number 50 says that Alice transferred the Gutenberg Bible to Bob. And suppose that, as before, Alice alters her ledger by replacing this statement with a fraudulent statement which says she transferred the Gutenberg to Carol.

Given condition (1), this action automatically destroys statements 51 through 100 in her ledger. So now Alice's fraudulent ledger is half the size of Bob's legitimate one. This disparity in size is a smoking gun: it shows that Alice has altered her ledger. If not for her devious deed, her ledger would have been just as large as Bob's.

Condition (2) now makes the gun hard to dispose of. Suppose Alice tries to conceal her fraud by enlarging her ledger so that it is once again the same size as Bob's. To do this she will somehow have to add 50 statements to her ledger. But given condition (2), this is likely to take a long time. And during this time Bob is likely to receive statements circulated by other members of the community, to check each of them, and to include in his ledger those that are valid. So his ledger will likely continue to grow beyond its current stock of 100 statements. Of course, condition (2) means that adding statements will likely be slow for him too. But because he has a head start on Alice, it will be hard for her to catch up.

What this shows is that anyone who tampers with her own ledger is likely to end up with a smaller ledger than someone who does not. And *that* means that it is likely that the largest

ledger in the community—the one with the most statements—has *not* been tampered with.[3] Let us say that a ledger has *integrity* when it is likely that it has not been tampered with. The core insight behind blockchain is that the largest ledger in a blockchain-using community will have integrity.

Let us now return to the question of how our two conditions could actually be enforced. Take condition (1) first. If statement number 50 is altered, we need not literally destroy statements 51 through 100. It would be enough to somehow force there to be a clear mismatch between statement 50 and all subsequent statements. Then it would be obvious to everyone, or everyone who checked, that the ledger had been tampered with. The only way for the tamperer to avoid suspicion would be to remove statements 51 through 100 from the ledger. So altering statement 50 would—effectively, if not literally—destroy statements 51 through 100.

Here is a simple way to achieve this. Insist that, in order for a ledger to be above suspicion, each statement must include an exact copy of every previous statement. Statements 51 through 100, then, will each contain a copy of statement 50. So if the *original* statement 50 is altered, the mismatch with the later statements will be obvious to everyone who looks. (Note that this simple method uses a huge amount of storage space, and so for that reason it is not employed in practice. Actual blockchains employ sophisticated cryptographic techniques to achieve the same end in far less space.)

Now take condition (2). How can we ensure that adding a statement to the ledger is likely to take a long time? A straightforward way is to require that, whenever a new statement is added, one must add something alongside that statement whose creation is likely to be time-consuming. This thing could in principle take a number of forms. It could be a ship in a bottle, for instance,

[3] How likely exactly? Quantitative matters lie beyond the scope of this paper, but for one relevant discussion see Nakamoto (2008, §11).

or an original pen-and-ink illustration, or a sweater knitted by hand. But if we want to implement our blockchain system digitally, we will need something digital. Here, in essence, is what actual blockchains do.

A *one-way function* $f(X)$ is a certain kind of mathematical function. Given X, it is easy to calculate $f(X)$. But given $f(X)$, it is very hard to work backward and find X. A good example is multiplication, if we require X to be a set of prime numbers. It is easy to find the product of several prime numbers, but it is much harder to find a number's prime factors.

Now suppose we have some particular way of digitally (and thus mathematically) encoding ledger statements. We can then define a one-way function f that takes as input a ledger statement together with an integer called a *nonce*. The function yields as output another integer. And when a new statement is added to the ledger, we require that, alongside that statement, a nonce is added which satisfies the following condition: the function f, applied to the new ledger statement together with the nonce, must yield as output an integer lying in some specified small range. It is not easy to find a nonce that works. Since f is a one-way function, one cannot easily work backward from the desired range of output to find a nonce that will yield an integer in that range. The only practical way to find a working nonce is by trial and error. And although one *could* hit upon a solution right away, it is much more likely that this will take a long time. (But once a working nonce is found, it is easy for anyone to verify that it works.)

This nonce—the one that yields the right value for f—is exactly what we need in order to enforce condition (2). Like a ship in a bottle, this is a 'product' whose creation is likely to be time-consuming. But unlike the ship, it takes a digital form. A ledger will be above suspicion only if it includes a working nonce for each of its statements.

By enforcing conditions (1) and (2) in these ways, a community can make it likely that the largest ledger existing in the community will have integrity.[4]

The community can make this even more likely by working together. Let us suppose that when a new valid transfer statement *S* is circulated throughout the community, each member of the community begins working on adding *S* to her ledger. This will require finding a working nonce (assuming that the community is implementing the blockchain method digitally). But let us suppose that when someone does manage to find one, she promptly circulates her updated ledger, with the newly added statement *S* and its nonce, throughout the community. And let us suppose that the other members of the community, if they see that her ledger is larger than theirs, simply adopt hers as their own. (Or at least, they adopt it after checking to make sure that all its statements are valid, that (condition 1) each statement matches every previous statement, and that (condition 2) all its nonces work.) If they do this, then they can stop searching for a nonce for *S*, since one has already been found, and start searching for a nonce for the *next* circulated transfer statement.

Let's call the members of the community who work together in this way the *honest* members. Thanks to their teamwork, the honest members' ledgers will grow faster than Alice's fraudulent ledger. It will be hard for her to overtake them.

To be clear, honesty thus characterized is no great virtue. To call someone honest is not to say she is in any way morally upstanding; it is only to say that she is one of the people working together in the way just described.

---

[4] I have here described the so-called 'proof of work' method used in Nakamoto (2008). Because finding a nonce (or constructing a ship in a bottle) is likely to take a lot of work, adding a new statement to the ledger is likely to take a long time. Other methods for ensuring the integrity of ledgers have been proposed but have been less widely adopted; I cannot discuss them here.

Faced with the honest members' teamwork, Alice's best strategy will be to mimic them. In just the way that the honest members work together to find working nonces for new ledger statements, Alice will have to rely on the help of some accomplices to generate working nonces for her fraudulent statements. And Alice's syndicate will have to do this faster than the honest members. But if most members of the community are honest, then it will be very hard for the syndicate, even working together, to outpace them.

The syndicate might think to buy a very powerful computer, one that can test many, many nonces each second. In fact, since the members of the syndicate, by working together, are in effect pooling their computing power, they might *each* try to buy powerful computers. These acquisitions would indeed help them grow their fraudulent ledger faster. But the honest members are also pooling their computing power. If most members of the community are honest, then it will be very hard for the syndicate to acquire more computing power than all of the honest members put together.

Thus by working together, the community can protect itself against malicious actors like Alice. And this kind of teamwork also brings a further benefit: it increases the extent to which any two ledgers agree. Since the honest members of the community are constantly circulating ledgers among themselves, their ledgers will tend to agree, or very nearly agree. The largest ledger, then, will not only have integrity, it will also be widely shared.

We will call this method, or system, the *blockchain system*. The largest ledger in such a system we will call the *blockchain*.[5]

---

[5] The blockchain system just described is a simplified version of the system in Nakamoto (2008). That system contains a further mechanism for adjusting the difficulty of finding a nonce. This mechanism was later discovered to undermine the claim that the largest ledger will have integrity. Nakamoto included the difficulty-adjustment mechanism because he expected that computers would grow faster and that the ranks of honest community members might expand,

We have so far described the blockchain system as a system for preserving data about rare book holdings. But in principle, this system could be used to preserve data about a range of things. We will consider some other possibilities below.

We are now able to understand part of blockchain's relationship to trust. Let us say that a *community* is honest when most of its computing power is controlled by the honest members of the community. Then it is only if a community is honest that its blockchain will have integrity. We therefore see that a certain level of trust is a prerequisite for the blockchain system.[6] One need not trust anyone in *particular* to be responsible or to avoid attempting fraud. But one does need to trust that the community as a whole is honest.[7]

But although in this way the blockchain system requires trust, there is also a way in which it creates trust. If the community is honest, then the blockchain will have integrity, and so one can trust that it has not been tampered with.

---

thereby reducing the time it takes to find a nonce. However, the inclusion of the adjustment mechanism leaves the system open to a certain attack. Once computers have grown faster and the nonce difficulty has increased, an attacker could quickly add many fraudulent statements to her ledger under the *old* difficulty level and thereby end up with the largest ledger. In the presence of such an attacker, it is no longer the largest ledger which has integrity; it is rather the ledger which has likely taken the most computational work to produce. The bitcoin code was updated in 2016 in response to this discovery; its users now adopt a ledger as their own, not if it is larger, but if it has likely taken more work to produce.

[6] Lipman (2023, 801) argues that 'there is a clear sense in which bitcoin is indeed aptly described as trust-free'. But his point is that bitcoin users need not trust the community to accept certain rules governing the construction of social objects. The sort of trust being discussed here is different.

[7] If one implements the blockchain system by running a computer program, one may choose to trust the author of that program to have written it properly. But one need not: one can always use open source software and inspect the code oneself.

So the blockchain system is a kind of trust transformer. It takes a generalized trust in the community as a whole and transforms it into a specific trust in the integrity of a particular blockchain. As long as there is the right kind of trust in the community as a whole, the system guarantees the existence of a blockchain whose integrity can be trusted.

§2 Accuracy

The fact that the integrity of the blockchain can be trusted goes only so far. Just because the blockchain has *integrity* does not mean it is *accurate*.

To see why, consider again the rare book community. Suppose Alice circulates the statement 'Alice transfers the Gutenberg Bible to Bob'. And suppose this statement is judged valid and is admitted to the blockchain.

Now imagine that years pass. Thousands of other books change hands. The blockchain grows. Assuming the community is honest (I won't always bother to make this assumption explicit from now on) the blockchain will have integrity. So even years later, we can be confident that it will still contain the statement 'Alice transfers the Gutenberg Bible to Bob'.

But in spite of all that, it might be that Alice in fact never physically transferred the Gutenberg to Bob and only circulated a statement saying she did. The book might still be sitting, carefully preserved, in Alice's storage facility. If so, then the blockchain contains a false statement. It represents Bob as possessing the Gutenberg when in fact he does not. Nothing in the blockchain method prevents this kind of misrepresentation. Although the method ensures that the blockchain has integrity, it does nothing to ensure that it is accurate.

This point might seem obvious, but it is often overlooked. Why is that? I suspect it is because the original and best-known application of the blockchain system is one in which the

blockchain is unlikely to contain falsehoods. This is the application to cryptocurrency. But to see why falsehood is unlikely there, and where else it might be unlikely, we need to consider another case first.

§2.1 Vaulted gold

Imagine a community that uses gold coins as money. For simplicity, imagine that every coin has the same weight and purity and that every coin bears a unique serial number.

As many philosophers have argued, this currency system requires the community to *recognize* these gold coins as money, in the way that Americans recognize dollars as money and Swiss recognize francs as money.[8] I will not try to give a precise account of recognition here. I will however assume that recognition can be an implicit matter. There need not be anything like an explicit agreement among the members of the community in order for the community to count as recognizing the coins as money.

But not only must the community recognize the coins as money, it must also recognize what counts as *holding* a coin—as having the power to dispose of that coin in various ways, including by transferring it to someone else as part of a transaction.

There are a number of 'holding criteria' the community might (perhaps implicitly) adopt. Here is a straightforward one. The community might recognize someone as holding a coin if, and only if, she physically possesses it. There could be a community, for instance, in which everyone carries around a coin purse for everyday transactions. When a transaction occurs, the buyer takes some of the coins in her purse and gives them to the seller, who places them in her own purse and hands over the purchase.

---

[8] See, for example, Thomasson (2003), Hindriks (2006), and Passinsky (2020).

But this is not the only holding criterion a community might recognize. Imagine that the members of a community hate having to always carry coins around. Suppose they hold a meeting to designate a trusted member of the community as their *banker*. The community has the banker write down each person's name together with the serial numbers of the coins in her purse. Then everyone's coins are stored in an underground vault. Once the coins are in the vault, there is no longer anyone who physically possesses any of them: everyone's purse is empty. So under the old, straightforward criterion, no one holds any coins. But the community could recognize a new holding criterion: one holds a coin if, and only if, the coin's serial number is next to one's name in the banker's book. Under this new criterion, a transaction no longer involves the physical movement of coins from the buyer's purse to the seller's. Instead, the buyer tells the banker to cross out certain serial numbers next to her name and to rewrite them next to the seller's name.

This example illustrates a concept that will be crucial for understanding what makes a blockchain accurate. Consider the relationship between these two conditions:

1. You hold a certain gold coin.
2. That coin's serial number appears next to your name in the banker's book.

Under this vaulted-gold currency system, these two conditions will always go together. If you hold a coin, then its serial number will be next to your name, and vice versa. And it is no accident that they always go together: they *have* to go together. The reason is that under this currency system, you hold a given coin *because* its number is next to your name, and it is impossible for you to hold a coin without its being the case that you hold it because its number is next to your name.

More precisely, the relationship between conditions (1) and (2) involves a connection of metaphysical ground.[9] The facts about the banker's book, together with some facts about the vaulted-gold currency system, ground the fact that you hold coin n, and necessarily, if this currency system is in effect, then if you hold coin n, your holding coin n is grounded in this way. We may therefore say that under the vaulted-gold currency system, the banker's book *dictates* the facts about which coins you hold.

Now let us see how this sheds light on blockchain. Let us imagine that the banker rewrites her book so that it is in the format of the rare book ledger of §1. Suppose that at the initial meeting, the banker writes down sentences like 'Carol initially holds #1234' to record the fact that at that time Carol physically possessed the coin with serial number 1234. Then the coins are all vaulted away. After that, if Alice wants to transfer one of her coins, say coin 9876, to Bob, she writes down 'Alice transfers #9876 to Bob' on a slip of paper and gives this to the banker. The banker checks this submission against her book to see whether (a) the book has the entry 'Alice initially holds #9876' and does not have an entry of the form 'Alice transfers #9876 to X', or (b) the book has an entry of the form 'X transfers #9876 to Alice' and has no later entry of the form 'Alice transfers #9876 to Y'. If one of these conditions is satisfied, then the transfer statement is said to be *valid* and the banker adds it to her book. If not, not.

As in §1, the community can then avoid the need to trust the banker by implementing the blockchain method. Instead of having the banker administer the ledger, each member of the community can administer her own. When Alice wants to transfer a coin to Bob, she simply circulates a signed transfer statement to the community as a whole. And the community, we may suppose, will now recognize Alice as holding a given coin if, and only if, the blockchain—the

[9] The loci classici for the notion of metaphysical ground are Rosen (2010) and Fine (2012).

largest ledger—satisfies either condition (a) or condition (b) with respect to that coin. Now it is no longer the banker's book but the blockchain which dictates the facts about who holds which coins.[10]

This blockchain currency system has a remarkable feature: when a statement is added to the blockchain, it is guaranteed to be true. Suppose, for example, that the statement 'Alice transfers #9876 to Bob' is added to the blockchain. Given the definition of validity it must then be that, just *before* the statement is added, the blockchain satisfies either condition (a) or condition (b). But that is exactly what it takes for Alice to hold coin 9876, and so just before the statement is added, Alice holds coin 9876. Now just *after* the statement is added, the blockchain will have the entry 'Alice transfers #9876 to Bob' and no later entry of the form 'Bob transfers #9876 to *Y*'. But that is enough to make it the case that Bob holds coin 9876, and so just after the statement is added, Bob holds coin 9876. Since the coin was first held by Alice and then later, upon the addition of the transfer statement, held by Bob, the transfer statement is true!
The underlying explanation for its truth is that the blockchain dictates the facts about who holds which coins. That is why we can be sure that just before the statement 'Alice transfers #9876 to Bob' is added, Alice holds the coin. And that is why we can be sure that just after the statement is added, Bob holds the coin.

So far we have argued that a statement is guaranteed to be true at the time it is added to the blockchain.[11] If the blockchain is tampered with later, its statements may change from truths

[10] We should perhaps acknowledge that there is a sense in which the dictation is less direct than in the original banker's book case. After all, the blockchain principally contains transfer statements, and so most of the facts about who holds which coins will be grounded in the blockchain only by way of the facts about who transferred which coins to whom. But this distinction between more and less direct forms of dictation will not matter here.

to falsehoods. For example, suppose the statement 'Alice transfers #9876 to Bob' was fraudulently replaced years later with the statement 'Alice transfers #9876 to Carol'. That would not change the fact that the coin was actually transferred to Bob. Tampering with the blockchain cannot change the past. And so the blockchain would contain a false statement, namely 'Alice transfers #9876 to Carol'.

The fact that the blockchain dictates the facts about who holds which coins, then, does not entail that it dictates the facts about who *held* which coins. Of course, it is true that a community, by recognizing this or that holding criterion, has some measure of freedom to determine who *now* holds which coins. And it is also true that, in the past, that same community may have had some measure of freedom to determine who *then* held which coins. But the community cannot now determine retroactively who held which coins in the past.

What, then, would be the effect of the above act of tampering? Suppose that the blockchain contains no later statement about coin 9876. The tampering would not change the fact that Alice transferred coin 9876 to Bob. But given the community's criterion for what counts as holding a coin, the tampering *would* change the facts about who now holds coin 9876. Before the tampering, Bob holds it. After the tampering, Carol holds it. The tampering effects a transfer from Bob to Carol—one which the blockchain fraudulently conceals.

However, as we saw in §1, in an honest community this scenario is unlikely to occur. The blockchain will have integrity: it will be likely that it has not been tampered with. And since each of its statements was true when it was added, even some time later the blockchain is likely to

---

[11] The blockchain contains not only transfer statements but also statements about who *initially* holds various coins. But these are also guaranteed to be true at the time they are added. After all, at that time, there are no other statements in the ledger, so condition (a) is trivially satisfied. And this is enough to make these statements true.

contain only true statements. Let us say that a ledger is *accurate* when it is likely to contain only true statements. Then the vaulted-gold blockchain is accurate.

A qualification. There are various reasons why a ledger might be likely to contain only true statements. For example, perhaps the community consists entirely of ultra-virtuous saints who would never dream of submitting a false statement to the ledger. We will call a ledger accurate only when the reason for the likely truth of its statements is 'internal', having to do with the blockchain method itself or the content of the ledger's statements. This is indeed the case with the vaulted-gold blockchain.

§2.2 Cryptocurrency

We can now show that cryptocurrency blockchains are also accurate.

To see why, start by imagining that one day the vault becomes inaccessible—a landslide blocks the entrance, say. If the community is using the blockchain system of §2.1, this development would not affect it at all. It could simply carry on using the coins in daily transactions by updating the blockchain to reflect each new transfer from buyer to seller. The coins' physical inaccessibility would not matter.

Recall that we are assuming that each coin bears a unique serial number. For simplicity, let us suppose the serial numbers range from 1 to 1,000,000. Once the physical coins are inaccessible, a certain shift might well occur—perhaps gradually and implicitly, perhaps suddenly and explicitly. The community might come to recognize these *numbers*—the integers from 1 to 1,000,000—as valuable instead of the coins.

If the community changes what it recognizes as valuable in this way, then it will also have to change how it understands the entries in the blockchain. When the blockchain says

‘Alice transfers #9876 to Bob’, for instance, that can no longer be understood to mean that Alice transfers the coin whose serial number is 9876. Instead, it must be understand to mean that Alice transfers the number 9876 itself. Under this new system, people hold numbers, not coins.

This new number-based system is in one way very different from the old coin-based system. In the new system, it is numbers, not coins, which serve as money. But the two systems are in another way very similar. They have the same structure; they work the same way. Since it is clearly possible for a community to adopt the coin-based currency system, it is possible for one to adopt the number-based system too. What this shows is that it is possible to have a currency system in which the currency itself is not a material thing but rather a kind of abstract object.

We are now ready to talk about cryptocurrency systems, such as the bitcoin system or the ethereum system. Like the number-based system, cryptocurrency systems are abstracta-based currency systems. The currencies in cryptocurrency systems, such as bitcoin (in the bitcoin system) or ether (in the ethereum system), are not material things but rather abstracta of a certain kind.[12]

Take the case of bitcoin, the best-known cryptocurrency. Simplifying matters, the bitcoin blockchain can be viewed as consisting of chains of transfer statements. Here is an example of such a chain:

1. Alice initially holds a cache of 3.125 bitcoin.
2. Alice transfers to Bob the cache of bitcoin she holds according to statement 1.
3. Bob transfers to Carol the cache of bitcoin he holds according to statement 2.

[12] Lipman (2023) and Warmke (2024) can be seen as views on which bitcoin is abstract.

Assuming the blockchain contains no later statement of the form 'Carol transfers to *X* the cache of bitcoin she holds according to statement 3', this chain memorializes the complete transfer history of a certain cache of bitcoin. If Carol wants to transfer that cache to David, she (and she alone) can do this by adding to the blockchain the signed statement 'Carol transfers to David the cache of bitcoin she holds according to statement 3'. In general, the holder of a given cache of bitcoin is the recipient of the final transfer in the corresponding chain.[13] Only that recipient has the power to transfer the cache to someone else.

It seems clear that, in this example, Carol holds this cache of bitcoin *because* she is the recipient of the final transfer in the corresponding chain. And it also seems clear that there is no way for one to hold a cache of bitcoin without holding it because one is the recipient of the final transfer in the corresponding chain (or by initially holding it, as in the first statement). Evidently, the bitcoin blockchain dictates the facts about who holds various caches of bitcoin.[14]

Just as in the case of the vaulted-gold blockchain we discussed earlier, the statements contained in cryptocurrency blockchains are likely to be true. The bitcoin blockchain, for instance, contains statements saying that a certain cache of bitcoin is transferred from person *A* to person *B*. If such a statement is a valid addition to the blockchain, it must be that, before the addition of the statement, *A* holds the cache of bitcoin in question. And once the statement is added, *B* becomes the recipient of the last transfer in the corresponding chain of statements, thereby ensuring that *B* holds the cache of bitcoin. So *A* really does transfer the cache of bitcoin

[13] I ignore the complications that arise from the possibility of splitting and combining caches of bitcoin. See Warmke (2022) for more detail.

[14] This point is made by Glazier (2021) and Bailey et al. (2024, 24).

to B, and so the statement is true at the time it is added to the blockchain.[15] As in the vaulted-gold case, integrity then ensures that the blockchain will likely contain only true statements and so that it will be accurate.

2.3 Where accuracy comes from

We have now seen several examples of accurate blockchains. Cryptocurrency blockchains are among these. What is it in general about the accurate blockchains that *makes* them accurate?

Our examples show that accuracy flows from two features.

1. The blockchain dictates the very facts that its statements concern.
2. The blockchain has integrity.

The first feature ensures that, when a statement is added to the blockchain, it is true at the time it is added. For example, the statements of the vaulted-gold blockchain concern who holds which coins, and the blockchain also dictates those facts. (Recall that the blockchain principally contains transfer statements; I count these as concerning who holds which coins.) The second feature ensures that the blockchain cannot easily be tampered with later and so will likely continue to contain only true statements.

These two features of accurate blockchains are not wholly independent. A blockchain may possess the first feature because of what the community recognizes. But the community's recognition may in turn depend on the blockchain's possession of the second feature. The vaulted-gold blockchain, for example, dictates the facts about who holds which coins precisely

[15] The bitcoin blockchain also contains statements like 'A initially holds a cache of r bitcoin'. The bitcoin community will recognize A as holding this cache of bitcoin if the blockchain contains no later statement saying that A transferred the cache to someone else. So these 'initial holding' statements are also true at the time they are added to the blockchain.

because the community recognizes someone as holding a certain coin if the blockchain says she does. But it only makes sense for the community to do this if the blockchain has integrity. Who in her right mind would be willing to take the blockchain to be the final word on her assets if it could easily be tampered with?

We have also seen some examples of *non-accurate* blockchains. These are blockchains for which there is no reason—no 'internal' reason at least—to think its statements are likely to be true. The rare book blockchain is one of these. Non-accurate blockchains, like all blockchains, have integrity (assuming an honest community). So they have the second of the two features listed above. But they lack the first. The rare book blockchain, for example, is difficult to tamper with, but its statements concern who has which books. And the facts about who has which books are not dictated by any blockchain. One cannot make a book have a certain location merely by updating a database, no matter how sophisticated that database might be.

So what accounts for the difference between the accurate blockchains and the non-accurate blockchains? Not integrity; all blockchains have that. What accounts for the difference is that only the accurate blockchains dictate the facts that their statements concern. And so given our understanding of dictation, the difference between accurate and non-accurate blockchains is ultimately a difference in how their statements (or the corresponding facts) are metaphysically grounded.

We are now in a position to say more about blockchain's relationship to trust. We saw in §1 that the blockchain system is a trust transformer. It takes as 'input' a certain generalized trust in the community as a whole, and creates as 'output' trust in the integrity of a particular blockchain. We can now see that, when a blockchain also has the first feature listed above, it

creates a further sort of trust. It creates trust not only in the integrity of the blockchain but in its accuracy too.

§3 Conclusion

Call a potential application of blockchain technology *accurate* if it would involve an accurate blockchain; otherwise call it *non-accurate*. Many applications currently being explored are non-accurate, including applications to supply chain management, to health records, and to certificates of birth, marriage, and death. No blockchain can dictate the facts about where a certain widget is, or what medical condition someone has, or when someone died.

Although I cannot discuss this here, I believe that this focus on non-accurate applications is a mistake.[16] For the most part, the benefits of these applications of blockchain can be had more easily without blockchain, and the tamper-resistant nature of blockchains makes it hard to correct any inaccuracies that may arise.[17] Only the accurate applications fully exploit blockchain's potential as a trust transformer. It is these applications which should be our focus as we continue to explore the technology.[18]

---

[16] I briefly develop this line of thought in Glazier (2021).

[17] I am grateful to Jon Simon for discussion of this last point.

[18] This paper was originally drafted in 2019 after a characteristically thought-provoking conversation with Asya Passinsky; I am grateful to her for sparking my interest in this topic. I would also like to thank Andrew Bailey, Ira Bolychevsky, Natasha Chamilakis, Lewis Cohen, Jonathan Glazier, Tina Kourmpetis, Gary Miller, Bradley Rettler, Jon Simon, Craig Warmke, and audiences in Hamburg and Kirchberg am Wechsel for helpful comments and for discussion of these issues. I acknowledge the support of the German Research Foundation (grant KR 4516/2-1) and of the Swiss National Science Foundation (grant 197172).

References

Bailey, Andrew M., Bradley Rettler, and Craig Warmke. 2024. *Resistance Money: A Philosophical Case for Bitcoin*. Routledge.

Fine, Kit. 2012. "Guide to Ground." In *Metaphysical Grounding: Understanding the Structure of Reality*, edited by Fabrice Correia and Benjamin Schnieder. Cambridge University Press.

Glazier, Martin. 2021. "Enterprise Blockchain Doesn't Work Because It's about the Real World." *CoinDesk*, March 31. https://www.coindesk.com/markets/2021/03/31/enterprise-blockchain-doesnt-work-because-its-about-the-real-world.

Hawley, Katherine. 2017. "Trustworthy Groups and Organizations." In *The Philosophy of Trust*, edited by Paul Faulkner and Thomas Simpson. Oxford University Press.

Hindriks, F. 2006. "Acceptance-Dependence: A Social Kind of Response-Dependence." *Pacific Philosophical Quarterly* 87: 481–98.

Lipman, Martin A. 2023. "On Bitcoin: A Study in Applied Metaphysics." *Philosophical Quarterly* 73 (3): 783–802.

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.

Passinsky, Asya. 2020. "Social Objects, Response-Dependence, and Realism." *Journal of the American Philosophical Association* 6 (4): 431–43.

Rosen, Gideon. 2010. "Metaphysical Dependence: Grounding and Reduction." In *Modality: Metaphysics, Logic, and Epistemology*, edited by Bob Hale and Aviv Hoffmann. Oxford University Press.

Sorkin, Andrew Ross. 2018. "Demystifying the Blockchain." *New York Times*, June 27. https://www.nytimes.com/2018/06/27/business/dealbook/blockchain-technology.html.

Thomasson, Amie. 2003. “Foundations for a Social Ontology.” *ProtoSociology* 18-19: 269–90.

Warmke, Craig. 2022. “Electronic Coins.” *Cryptoeconomic Systems* 2 (1).

———. 2024. “What Is Bitcoin?” *Inquiry* 67 (1): 25–67.