What dreams may come: Bitcoin a revolutionary technology or passing fad?

Justin Anderson, Equity Analyst
June 2021

MAWER
Be Boring. Make Money.™

Trust: Humanity's philosopher stone

"Can I write you a check for the other 1k?" asked the would-be buyer of my 2005 Volkswagen Jetta. Admittedly, there was a slight risk that the check might bounce, but I knew the buyer's home address having stopped by the day before to pick up the deposit. The risk seemed slight and the cost of not accepting meant returning the cash deposit and continuing the hunt for a buyer. "Sure," I answered.

Was it the right decision? More on that later.

Stories along these lines have repeated since the dawn of civilization. When archeologists recover papyrus dating from ancient times hoping to learn about a past civilization they often find instead, lists of merchant records laying out transactions or specifying ownership. When the ancients chose to invest such scarce resources in writing, they usually did so not to help future historians understand the key elements of their culture, but for the practical business of securing trust.

Trust animates everything around us—however accustomed we've become to swimming in it. Only with a certain degree of trust can we park our natural existential anxieties and move on to the business of creating, improving, and trading something.

Nobel Prize winning economist Douglas North makes the case that low transaction costs, which are highly correlated with the level of general societal trust, help explain disparate economic performance globally. More trust = lower transaction costs = stronger economic performance.

But trust is expensive. The enormous relative effort that the ancients poured into manufacturing papyrus presaged the herculean effort that future societies would pour into institutions aimed at building and maintaining societal trust. Academics, bankers, accountants, politicians, judges, actuaries, police, bureaucrats and so forth can attribute much of their existence to lubricating the wheels of societal trust.

But even with so many resources brought to bear, trust remains vulnerable. After all, these institutions have evolved from and depend on fallible *humans*.

What if there was a way to build and maintain trust that did not depend on the integrity of certain institutions? Hopeful answers (and fantasies?) to this question is what animates proponents of decentralized blockchains. Elements of this new technology's emerging story seem to offer a positive answer to this question. Other aspects are more speculative, perhaps even unsolvable. Investors are left wondering if this is akin to nuclear fusion, to be suspended in futurist dreaming purgatory, or are we closer to something that is—or could be even more—revolutionary than the Internet?

Let's buy some Bitcoin

Understanding how Bitcoin works is critical for assembling the mental toolkit that will allow us to comprehend the more general role of blockchains—i.e., the role they may have in ushering in that lofty vision that reaches beyond a mere trustless currency/asset. By trustless, I mean a system that does not depend on the integrity of a central (trust-building) authority/institution. Bitcoin is currently the most successful expression of this trustless vision offered by blockchain technology as it approaches an incorruptible system—trust is built into the architecture with integrity depending on a majority of the network (distributed) rather than on a central authority.

Our journey into Bitcoin begins with signing up. How does one sign up? Well, remember there is no central authority or central entry point. We can "sign up" by picking a **private key**: any 256-bit number of our choosing be it via a random number generator or any other approach. Here is a sample key that I generated using this webpage: www.bitaddress.org:

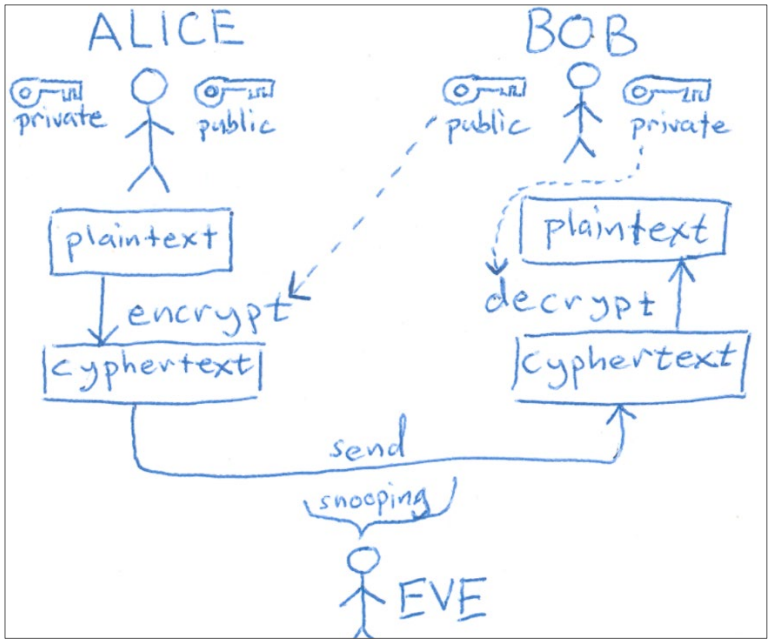
Private Key:
B38865B72CE6D1659C9C5958859CA75B1D1C1AE0F0134A14557B9A8193CA1604

Next, we plug our private key into an Elliptic Curve Digital Signature Algorithm (ECDSA) which will generate our **public key**. Note that we will use this public key to encrypt messages that only the associated private key holder can unlock. Here is the public key that is generated when we input our private key into ECDSA:

Public Key:
043F4A0125D51EEC68E2FDB4350F2D5575435E64B3B959A4D6D681E472A3BD4B64F84E5
DB562B1F5ECDFE571B4FCA495D27BEC1BE0CCADA8C67936E1C63E60AF1C

This public key is then put through two SHA-256 hashing functions—hashes are one-way functions i.e., trivial to produce the output from the input, impossible to derive input from output. As an instructive mental model, think of how 10 mod 3 = 1. In this expression, we cannot derive the inputs (10 and 3) from the output (remainder 1 when modding these two numbers). The best we can do is guess inputs to see if it gives us the desired output. The hashed hash of the public key becomes our bitcoin address. The address can be expressed in several formats, here is the base58 format which eventually gets added to the blockchain as our publicly visible **address**:

Address: **1N3h89TAgnL3TrKFR5zmM4SjmCopuSH95N**



Now comes the magical moment; we find someone who has some bitcoin and offer them something to trade. One of the first such trades happened on May 22, 2010, in Jacksonville, Florida when someone paid 10k bitcoin for two pizzas...today that bitcoin is worth roughly $600 million USD.

To make the transaction happen, we must generate a **UTXO** or an "unspent transaction output." The UTXO includes the details of the transaction including:

1. the *amount* of bitcoin to send,
2. the *address* the bitcoin is being sent to, and
3. a *locking script* that only the private key associated with the target address can unlock.

The entire blockchain is a linked list of blocks each which contain a list of transactions. In our example, the pizza buyer starts with a UTXO addressed to them, applies their private key to unlock that UTXO and then generates a new UTXO (with the amount, address, and locking script).

The locking script can only be unlocked by providing the private key associated with the public address used to generate the script. In our example, only we (or whoever we share it with) will be able to unlock this script unless we lose our private key in which case the UTXO can never be unlocked. Once unlocked, we are free to repeat the process for any UTXO addressed to us.

Suppose I wanted to try to hack a UTXO addressed to someone else. The only possible way to do it is via brute-force, namely I would have to guess a private key, generate a digital signature (the hash created by plugging the guessed private key + transaction data into the SHA-256 hashing function) and plug that digital signature into the UTXO's locking script to see if it generates a hash that matches the hash generated when separately plugging the transaction data into the SHA-256 hashing function. The nature of SHA-256 means on average we'd need to guess the key 2^{256} times before we guessed right. (The reader may want to Google "how secure is SHA-256?" to better appreciate the futility of this undertaking.)

While Bitcoin's security is a critical feature of the protocol, it also introduces a challenge. Forget your private key and the bitcoin associated with the address is lost forever. Estimates vary, but ~10% of bitcoin's current stock is effectively gone forever due to lost private keys. After all, there was a time when bitcoin was worth nothing so it's easy to understand why many private keys went missing. Perhaps, in a few decades treasure hunters will shift focus from scouring the deep sea to the personal effects of suspected lost key holders.

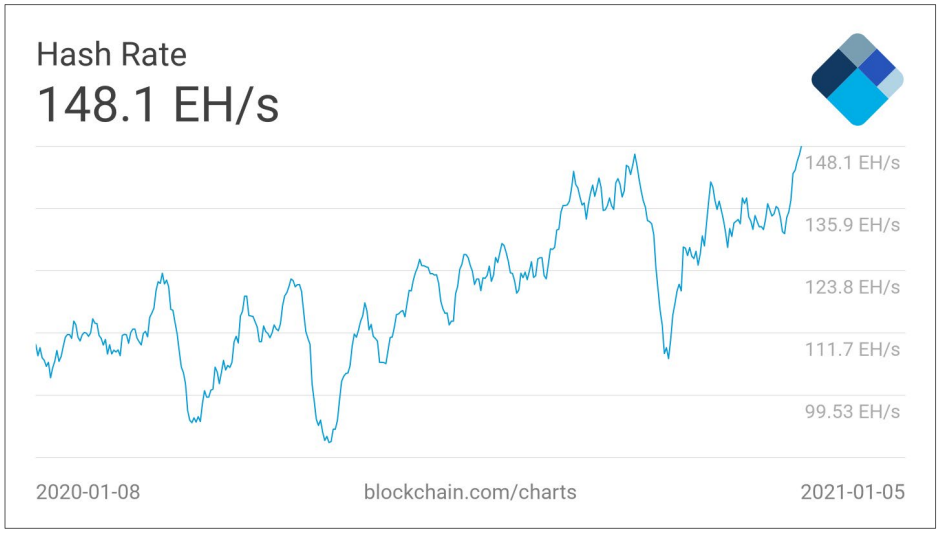
The process we have described so far is the "crypto" piece of cryptocurrency, namely it is using well established cryptography principles to send/receive data in a way that validates that the sending address also knows the sending address private key. But we have a problem. How do we keep the integrity of our linked list using a *decentralized* architecture?

Centralized authorities would ensure integrity via activities such as validating the UTXO signatures, checking UTXO inputs add up to the associated outputs, guarding against double-spending (i.e., someone trying to send the same UTXO twice) and determining which blockchain is the valid one. However, we need a system that incentivizes network participants to engage in these validation checks and one that prevents rogue actor(s) from corrupting the database *without relying on a governing central authority*. Solving this problem (validating the transactions without a central authority) is at the core of building a trustless architecture.

Let's mine some Bitcoin

"Proof of Work" is Bitcoin architect's (alias Satoshi Nakamoto) answer to this problem. Running the validation checks associated with a block of new transactions is a negligible computational task which would make corrupting it trivial. To make it a less trivial, Bitcoin's architecture uses that SHA-256 hashing function yet again.

A bitcoin miner's purpose is to create a new block to add to the chain. The new block must meet a set of conditions which indicate that the miner is proposing a legitimate block. "Legitimacy" is checked by the rest of the miners on the network who will add the proposed block to their own local copy of the blockchain if it meets the validation criteria. The legitimacy checks amount to ensuring that each transaction passes the validation checks *and* that the new block contains a nonce (a randomly generated input) which when plugged into SHA-256 (with the rest of the block data) generates a hash that is lower than the previous block's target hash. This "lower than" condition means that the difficulty of the brute force search can be adjusted. For Bitcoin, every 2016 blocks the target hash adjusts its difficulty up or down depending on the average block generation time for the previous 2016 blocks. The adjustment tries to move the average mining time to 10 minutes per block. Keeping the target time per block fixed means that the processing power increases as the value of bitcoin rises as there is more value offered to the miners. The "Hash Rate" (EH/s = exahashes, or quintillion hashes generated per second) is a proxy measure of the computing power being trained on generating new blocks.



Source: Cointelegraph.com

Miners are willing to bring all this processing power to bear for two reasons. First, when a new block is successfully created (meeting the validation checks of the network peers and containing an unlocking nonce) the miner is permitted to mint new coins into their own address (~6.25 coins in April 2021). Every 4.5 years, the amount of coins minted per new block is halved eventually to reach exactly 21 million coins in circulation.

Second, miners also receive a fee which is added to each transaction by the sending address when it is created. If no fee is included, then there is a risk that no miner will care enough to add the transaction to the block they are mining which means the transaction will not be processed (i.e., it won't be added to the blockchain). The fee is designed to be the only financial incentive for miners once the minting mechanism splits itself into obscurity.

What emerges from this architecture is a (nearly) un-hackable system without relying on a central authority. How can it be compromised? The answer is central authority—otherwise known as a 51% attack. This vulnerability depends on 51%+ of the processing power of the network coming under the control of a single actor. Going forward, this actor could act like any central authority; we're back to our fallible human institutions. Still, even under this scenario, a powerful feature of the blockchain is that re-creating a chain with different transactions and putting it forward as the legitimate chain is difficult: you'd have to repeat the process of brute-force finding a nonce for each block that satisfies the new (corrupted) transactions.

Competing chains is not just a concern for a 51% attack scenario. It also arises naturally. For instance, two miners might simultaneously find a nonce that meets the target criteria. Each miner would thus add the block to the chain leading to a fork: two blockchains each with a different most-recent block. Bitcoin solves this conflict by the principle of "longest chain = the valid chain." In our example, when the following block is mined and added to one of the two chains, that chain would then become the "valid" chain, orphaning the other. If there are 2+ equally long chains, this process would repeat until a single chain emerges as the longest. It is rare for multiple equally long chains to co-exist but it does happen. Because of this issue, your transaction is only truly "processed" after it is several blocks deep in the longest chain.

Beyond Bitcoin

Blockchain dreamers are thinking beyond merely disrupting the career plans of central bankers. The institutions of trust are wide-ranging, from lawyers and judges enforcing laws to financial exchanges settling stock trades to name a couple.

Ethereum, the second most popular blockchain protocol, aims to generalize blockchain technology from a specific application (a currency) to a general purpose protocol. Like Einstein's ambition to work out general relativity (starting from special relativity), Ethereum's ambition brings with it more mountains to climb with some peaks still not visible—

as well as heaps of possibility and hope. Ethereum powers ~3,000 different blockchain applications (decentralized apps or "Daaps") including cryptocurrencies, smart contracts, and NFTs or "non-fungible tokens" to name a few. Just as ledgers have been around since financial assets were traded, the tokens in NFTs are nothing special. A token is a unique key that indicates ownership of something, typically a digital asset such as a work of digital art or a skin used in a video game. The "non fungible" piece means that NFTs are not directly exchangeable with each other as no two NFTs are identical (unlike bitcoin). NFTs are not divisible (unlike one bitcoin which can be sliced into 100 million "Satoshis").

The digital asset the token is pointing to is not unique and can be copied easily, however, the token itself is unique (to its blockchain) and impossible to destroy, remove, or replicate. NFTs are tied into the blockchain that spawned it and typically that same blockchain enables the transfer of the token leveraging similar cryptography concepts as Bitcoin.

Communities are an additional aspect of NFTs, i.e., each token class has a sponsoring community. For instance, Cryptoart.io displays art by artists who upload their work to the "SuperRare" protocol. SuperRare, leverages Ethereum's blockchain to power the transactions, but also carries meta-data unique to the SuperRare protocol. Interoperability (with other Ethereum-based protocols) may or may not be built into the meta-data (those decisions are up to the relevant community architects).

Some of the top NFT communities today (by transaction value) include GodsUnchained (collectible game cards), OpenSea (art), MyCryptoHeroes (game icons), ENSdomains (domain names), decentraland (virtual worlds), CryptoKitties (art) and SuperRare (art). Nothing technically stops an artist on SuperRare from generating a token that points to art already registered on say CryptoKitties—ultimately the authority/validity starts and stops with the integrity and enforcement mechanisms available to the community itself. These various communities are analogous to how Bitcoin is "competing" with other would-be cryptocurrencies such as Bitcoin Cash, Litecoin, or Tether.

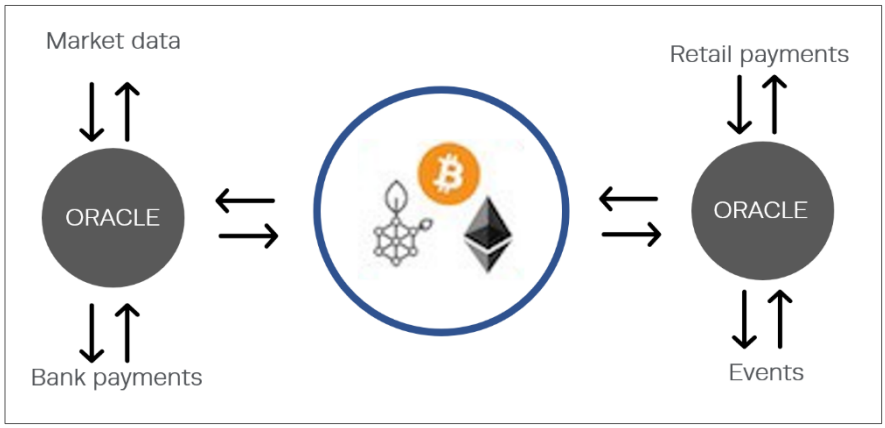
Automating management

Imagine we have a new business idea. Let's create a marketplace that lets people place sporting bets and then allows them to later transact on those placed bets. If we do it right, our NFT will not depend on the intervention (and hence longevity) of our organization. In other words, no central authority is required to run or validate the bets or subsequent trades—these duties are left to the participating distributed nodes.

This kind of self-automated system excites proponents of Ethereum who see cutting management overhead and replacing it with an assortment of finely tuned blockchain protocols as pursuing their founding purpose be it to enable sports betting, real estate buying, and so forth.

But there are challenges.

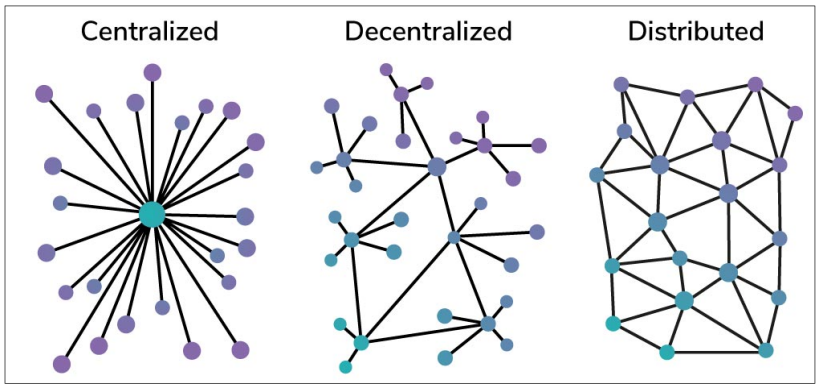
The Oracle problem



A challenge facing many blockchain protocols creeps into the picture when the protocol reaches outside of the chain to something external to arbitrate. Bitcoin does not face this challenge as all of the relevant data (i.e., coin ownership) and the mechanism for introducing transactions is fully encapsulated by the architecture. It's all included on the blockchain. In our sports-betting NFT example, we would have to rely on a third party (referred to as an Oracle) to rule on who won a particular game. This starts to sound a lot like a central authority where the community, regulations, and enforcement mechanisms ultimately drive the success of our venture rather than the technology itself orchestrating a trustless outcome. People are trying to find solutions to these problems such as chain.link which is working at creating a kind of crowdsourcing of truth solution to mitigate the Oracle Problem when reaching off-chain for data that will impact the blockchain. When you hear "smart contracts" you should immediately wonder about how they propose to handle the Oracle Problem.

Investor take-aways

Investors should think about the current state of blockchain technology at a high level, as a tech enabler of decentralized and distributed architectures. Such architectures bring advantages and disadvantages analogous to the trade-off between democracy (distributed power, inefficient, robust) and autocracy (centralized power, efficient, fragile). The cost of distributed/decentralized architectures in many contexts may be reduced dramatically via blockchains. That in turn should lead to more decentralized/distributed architectures bringing their benefits to bear such as pushing up general levels of trust.



As for Bitcoin specifically, its *currency* ambitions remain elusive. Bitcoin's architecture uses 1 MB sized blocks which effectively limits the number of transactions possible per block such that it would not suffice as a global currency. Resistance to changing the block size to 8 MB blocks led to an offshoot protocol called "Bitcoin Cash." The story of Bitcoin Cash illustrates how hard it is to make changes to the protocol. The beauty of distributed security brings with it the challenges of distributed governance: transactions are not reversible complicating dispute resolution (an important feature of incumbent payment networks), and the price of bitcoin itself is volatile, another strike against its envisioned currency role.

Bitcoin's *"digital gold"* ambitions, however, have never looked better. Bitcoin is a textbook description of what makes for a good store of value. Some of that value accrues due to its digital nature. The cost of not having to store or protect it contrasts favorably against physical stores of value such as real estate and gold. No material closing or shipping costs. Physical stores degrade with time. Other features make Bitcoin unique vs. other digital stores. Transactions, while not free due to proof of work, are at the core tamper-proof. Built-in scarcity contrasts with fiat authorities and their "flexible" monetary policies. Distributed governance protects it from the corruption that these authorities, given enough time, inevitably succumb to. The entire history is available to everyone making auditing trivial. Most importantly for any store of value, the number of people who accept it as inherently valuable continues to grow. Increasing government intervention in the economy is likely to only expand interest in store-of-value safe havens.

Other blockchains are more speculative passion projects. NFTs today resemble the world of art collection. This is not to dismiss these experiments. Bitcoin was once such an experiment. Disruptive technology tends to start as a

fringe experiment before evolving into something world changing. Others (most) will be passing fads akin to a sports card collection craze.

Pay attention to the application context. Some of the more promising contexts in my view:

- Decentralized blockchains seem well suited for maintaining namespaces. As an example, it's easy to imagine that the map of DNS to IP addresses for the internet would most easily be maintained in a decentralized blockchain. This context has the added benefit of encapsulating the space, meaning all the relevant info could be contained on the chain (i.e., avoiding the Oracle Problem)
- Powering secondary markets. The potential in blockchain in many contexts might be less about the trustless feature that is appealing in Bitcoin due to the Oracle Problem. However, the opportunity to drive down the cost of managing/settling trading of assets at low cost within niche markets is promising. More generally, the possibility of decentralizing management to the protocol carries world-changing potential—if still distant and impossible to predict.

Overall, Bitcoin, and blockchains more generally, are unlikely to usher in a trustless utopia, however, in some contexts, they should deliver improved trust/cost trade-offs disrupting traditional trust-building institutions.

Finally...was I right to take the check?

Turns out it was the wrong decision to accept the check. It bounced and all attempts at communication were ignored. I appealed to a credit agency (another trust institution) and registered the swindle against the buyer's credit, but ultimately never received any further word about it. Unfortunately, the trust institutions let me down.