How a Blockchain Works and What It Can Do (in plain english)

by
Ethan Mackay
ethan@jurisproject.io

Version 3 - 11/7/18

Introduction

In order to graduate from Columbia Law School, a student is required to write a "substantial and rigorous piece of legal writing" in the neighborhood of 7,000 words. In my final semester at the school, after procrastinating for months longer than I should have, I decided I would write a paper on how a blockchain works. At the time, I knew virtually nothing about how a blockchain worked. I just knew my brother, Adam, was convinced the technology would change the world and I thought I might want to get involved. Substantial and rigorous writing seemed like a high bar to clear since I was starting from near total ignorance and only had a few weeks. But I figured there was opportunity in my ignorance because it was a common deficiency. It seemed to me that there were mostly two kinds of people in the world in March 2018: those who knew an awe-inspiring amount about blockchain and wouldn't shut up about it (like Adam) and those who were ignorant, intimidated, and mildly annoyed to hear about it (like me). My hope was that this paper would be like a journal I kept as I passed from the latter group into the former, and that it might act as a bridge for other people to cross. The bridge is simple, and lacking in any computer science pyrotechnics, but hopefully it's sturdy enough to get you across (or just get you somewhere else). After I wrote it, I became so intrigued by the promise of the technology that I decided I wanted to get very involved. I changed my career path and joined a blockchain startup that Adam co-founded, called Juris. If you're curious, I'll tell you about the problem we're working on at the end.

This paper answers eight simple questions. (1) What is a blockchain? (2) How do you build a blockchain? (3) Why can't you cheat a blockchain? (4) How do you get people using a blockchain? (5) How is a blockchain better than what we already have? (6) What could we use a blockchain for today? (7) Why is a blockchain community worth building? (8) What is Juris doing?

While I'll describe most of these concepts in plain english, sometimes I'll delve into technical explanations that I think are really interesting, but that might make you sleepy (or want to die). If you ever feel fatigued, you can skip down to the next purple box to read a short, simple summary of what I just covered.

Part I: A blockchain is a list

Maybe you already read a brief explanation of what a blockchain is. The author might have used some obscure words like database, decentralization, ledger, trustless, or cryptography. Maybe those words were totally unfamiliar to you. Maybe you knew the individual terms, but when they were put together you felt like you only half-understood the idea. That's how I felt. In this paper, I'll try to avoid jargon-packed, dense sentences. I'll try to explain a blockchain in terms as basic as I felt were required for me to really understand it, i.e. very basic. I'll start by outlining some very obvious processes, but then things will accelerate.

Here's the punchline... a blockchain is a list. It's a way of storing information.

How we use lists

Today, I mowed my neighbor's lawn and made $20 in cash. I put it in my wallet, which was filled with old tickets and folded receipts and faded photos. I was worried I would forget I had it, so I took out my notepad and started a list. The $20 was my first entry:

What Do I Have?			
From	**To**	**Amount**	**For**
Neighbor	Me	$20.00	Mowing lawn

Then I went to the grocery store, where I decided to buy an apple ($0.50) and a bottled water ($1.00). I'm pretty dumb so I couldn't remember whether I had enough cash to buy them. I checked my list in my notepad and saw that I had enough, so I bought the items and put the change in my wallet. I wanted to remember what I had spent, so I made two new entries in my list. Now it looks like this:

What Do I Have?			
From	**To**	**Item/Amount**	**For**
Neighbor	Me	$20.00	Mowing lawn
Me	Grocery Store	-$0.50	Apple
Me	Grocery Store	-$1.00	Bottled Water

Then I painted my neighbor's house and he paid me $300. I put it in my wallet and made a new entry. Now my list looks like this:

What Do I Have?			
From	**To**	**Item/Amount**	**For**
Neighbor	Me	$20.00	Mowing lawn
Me	Grocery Store	-$0.50	Apple
Me	Grocery Store	-$1.00	Water
Neighbor	Me	$300.00	Painting house

At that point, my wallet was stuffed with cash. I was nervous about losing it so I opened a checking account at the bank and got a debit card. I deposited $300 of my cash into the account. When I opened my account, the bank created their own list. The first entry in their list was my $300 deposit:

Ethan's Account			
From	**To**	**Item/Amount**	**For**
Ethan	Checking Account	$300.00	Deposit

Note that this was a big decision for me to make. By giving the bank my money, I placed my trust in the bank. I trusted them (1) to keep track of how much of my money they had, (2) to give me my money back when I asked for it, (3) to not give my money to someone else, and (4) to not disclose to others the personal information they

collected from me (name, social security number, address, etc.). I'm accepting the risk that they might mess these things up. I think this risk is acceptable because (1) I can try to sue the bank if it messes up, (2) my deposits are FDIC-insured, (3) they're paying me interest, and (4) keeping the money in my wallet seems riskier.

Anyway, this afternoon I went to the department store and wanted to buy a shirt ($20.00) and some sunglasses ($60.00). I checked my list again to confirm that I had enough money to afford them. This time the payment process was more complicated than it was at the grocery store. I authorized the department store to contact the bank and take my money. The department store notified the bank that it was owed some of my money for a transaction. The bank checked its list to make sure I had enough money to pay for them. The bank saw that I did, so they told the department store that it could take some of my money. The bank made a new entry in their list to record that it sent some of my money to the department store:

Ethan's Account			
From	**To**	**Item/Amount**	**For**
Ethan	Checking Account	$300.00	Deposit
Checking Account	Department Store	-$80.00	Withdrawal (Department Store)

I'm still keeping my list in my notepad, so I also made a new entry. Now my list looks like this:

What Do I Have?			
From	**To**	**Item/Amount**	**For**
Neighbor	Me	$20.00	Mowing lawn
Me	Grocery Store	-$0.50	Apple
Me	Grocery Store	-$1.00	Water
Neighbor	Me	$300.00	Painting house
Me	Checking Account	$300.00	Deposit
Checking Account	Department Store	-$80.00	Shirt and Sunglasses

The importance of lists

Everything runs on lists like these. Individuals have their lists. Corporations have their lists. Governments have their lists. Lists are a key part of how our property system works. We can own something without having to constantly guard it because there's a record somewhere that we own it and that record will be enforced by other individuals, companies, and governments. And lists aren't just for tracking property. They are also how we keep track of associations. The Social Security Administration has a list with our names and our associated social security numbers. The hospital has a list with our names and associated medical records. The university has a list with our associated grades. The police station has a list of our associated arrest records (unfortunately). Lists are how we remember who we are, what we own, and what we have done. Lists are organized information. You already know all this.

A blockchain is list of information that is stored in a new way

Lists can be inaccurate. The two main causes of inaccuracies are (1) accident and (2) mischief. I drop my notepad in the toilet and lose a few pages of entries. I make a $300 cash deposit at the bank and some glitch in their software erases the entry from their list. My sister steals my list and adds a new entry saying that I owe her $100. Someone hacks into the bank's computer system and sets my balance to $0. Hopefully, I'll

prevent and discover any inaccuracies, but as a list grows, the prevention and discovery of inaccuracies becomes increasingly difficult.

If a list is inaccurate, it is less helpful, so ever since human beings have recorded information, we've developed and used methods to prevent inaccuracies. We've carved information into stone. We've copied information into books and stored them in towering libraries. We've triple-checked our entries. We've designed elaborate systems that use barcodes and redundancies. We've written computer programs that constantly check that the numbers add up. We've hidden our lists away from others with invisible ink and cyphers and firewalls and safes and more. While some of these methods are good, none are perfect. Sometimes erosion wipes the stone. Sometimes libraries burn down. Sometimes hackers breach our firewalls and alter our lists. All lists bear some risk of inaccuracy.

A blockchain is a list of information that guards against inaccuracies by a new method.[1] Instead of hiding your list, you use software to give it away. And you give it away to as many people as possible. The more the better. A blockchain exists on tens, or hundreds, or thousands, or millions of different computers simultaneously. These computers talk to each other to constantly reaffirm that their version matches the versions on the other computers. With each new entry made to a blockchain, the trustworthiness of past entries rises because with each new entry, more work has been put into checking the list one more time. Once you have this new way of storing information running, you can unlock some crazier, almost magical things.

But I'm getting ahead of myself. Describing a blockchain in this way is too abstract. Let's just build the thing so you can see how this works.

Modern society functions because human beings can record information into lists and rely on these lists to track who they are, what they own, and what they've done. But the ways we store information have serious flaws. A blockchain is a new way of storing information that might represent a significant improvement.

[1] In this explanation, I'll detail how the Bitcoin blockchain works, since it is the most famous blockchain, but you should know other blockchains can work quite differently.

Part II: Let's build a blockchain

Building a community

My notebook is drenched in toilet "water". My bank doesn't remember that I deposited $300 in cash. I'm pissed. I've learned (sort of) how a blockchain works, so I decide to organize my information with a blockchain instead of the old types of lists.[2]

The first thing I'll do is find some other people who also want to use this new type of list to record their information. The benefits of a blockchain are only unlocked when many people are involved with it, so the more list-keepers the better. I don't have many friends (partially because I talk too much about blockchains) so the most I can wrangle to participate is five: Justin, Sean, Michael, Rob, and Matthew. Together, we will form the community of participants that will use the list. Each will have a vested interest in the list's health, since we're all relying on it.

A false start

Now that I have my community, we dive right into using it. We six each create an empty spreadsheet on our computers and we create a group on our phones for text messaging each other. We each put $20.00 into a cash box to fund the list, which I'll keep at my house. The first entries on our spreadsheets account for this initial contribution, showing that the cashbox has funded our accounts:

[2] Again, this blockchain will be a simplified version of the Bitcoin blockchain, but you should know other blockchains work differently.

The Blockchain			
From	**To**	**Item/Amount**	**For**
Cashbox	Ethan	$20.00	Initial funding
Cashbox	Justin	$20.00	Initial funding
Cashbox	Sean	$20.00	Initial funding
Cashbox	Michael	$20.00	Initial funding
Cashbox	Rob	$20.00	Initial funding
Cashbox	Matthew	$20.00	Initial funding

Each of us record all of these entries into our spreadsheets. Now that there's some money in the list, we're ready to use the list to track our payments in the community. Michael and Sean go see a movie. Michael pays for Sean's ticket with cash and Sean want to pay him back, so he texts the blockchain group saying that he sends $10 to Michael for the movie ticket. We all see the text, so we open our spreadsheets and add this entry to our version of the list:

The Blockchain			
From	**To**	**Item/Amount**	**For**
Sean	Michael	$10.00	Movie Ticket

Now, if we want to see how much money Sean has according to the list, we look at all the entries in the list:

The Blockchain			
From	**To**	**Item/Amount**	**For**
Cashbox	Ethan	$20.00	Initial funding
Cashbox	Justin	$20.00	Initial funding
Cashbox	Sean	$20.00	Initial funding
Cashbox	Michael	$20.00	Initial funding
Cashbox	Rob	$20.00	Initial funding
Cashbox	Matthew	$20.00	Initial funding
Sean	Michael	$10.00	Movie Ticket

Adding the relevant entries up, we see that Sean had $20.00 from the initial funding, but then transferred $10.00 away, so he has $10.00 remaining. The entries also show Michael had $20.00 from the initial funding, then received $10.00 more from Sean, so he has $30.00 total.

Then, Michael buys Rob's saxophone from him, so Michael texts the group that he sends $20 to Rob. We all see the text and add this entry to our lists:

The Blockchain			
From	**To**	**Item/Amount**	**For**
Michael	Rob	$20.00	Saxophone

Now, on our spreadsheets, by looking at all the entries, we see that Michael has a balance of $10.00 and Rob has a balance of $30.00:

The Blockchain			
From	**To**	**Item/Amount**	**For**
Cashbox	Ethan	$20.00	Initial funding
Cashbox	Justin	$20.00	Initial funding
Cashbox	Sean	$20.00	Initial funding
Cashbox	Michael	$20.00	Initial funding
Cashbox	Rob	$20.00	Initial funding
Cashbox	Matthew	$20.00	Initial funding
Sean	Michael	$10.00	Movie Ticket
Michael	Rob	$20.00	Saxophone

Just as this system starts rolling along, I realize it won't work. I wake up in the middle of the night after having the kind of nightmare law school will give you, in which an evil actor messes up the whole scheme. Matthew has a wicked thought. He opens his spreadsheet and goes to the most recent entry and changes Rob's name to his name, so it looks like this:

Matthew's Blockchain			
From	**To**	**Item/Amount**	**For**
Cashbox	Ethan	$20.00	Initial funding
Cashbox	Justin	$20.00	Initial funding
Cashbox	Sean	$20.00	Initial funding
Cashbox	Michael	$20.00	Initial funding
Cashbox	Rob	$20.00	Initial funding
Cashbox	Matthew	$20.00	Initial funding
Sean	Michael	$10.00	Movie Ticket
Michael	Matthew	$20.00	Saxophone

Now, on Matthew's spreadsheet, it looks like he received the money that Michael intended to send to Rob.

If everyone else were honest this wouldn't be a big issue. Eventually we'd all compare our lists and see only Matthew's is wrong, so we'd assume his isn't trustworthy. But what if there are multiple wrongdoers? Maybe Matthew, Michael, and Rob all make changes like this so they get the money in the system. When we go to compare our lists, Matthew might say I paid him $100, Michael might say I paid him $84, and Rob might say I paid him $2,000. The blockchain is useless because I don't know which version to believe.

A blockchain can be used to record the movement of money by making an entry each time money moves. A blockchain won't work unless there is some way to tell which version of the list can be trusted.

What you should know about cryptography

The way I'll make the list trustworthy is by using encryption. The word "encryption" intimidates me because I know it's complicated and I know my technology relies on it and I know I don't fully understand it. So I'll introduce it briefly.

When you encrypt something, all you're doing is performing some process on some information in order to hide it. Let's say I want to send a message that reads "stay", but I don't want anyone else to understand it. The encryption process I'll use is to move all the individual letters up the alphabet by three places. So "stay" becomes "vwdb". After I'm done, the information "stay" is still contained in "vwdb" but it has been hidden. The information can be uncovered by someone who knows the process. This is a simple encryption process with just one step, known as a Caesar cipher. Other encryption processes are far more complex because they use computers that perform many steps to hide the information.

The United States National Security Agency (NSA) has been working on cryptography programs since the 1950s. It is very good at it. One of the programs the agency created is called Secure Hash Algorithm 256 (SHA-256). This program seems kind of magical. It's like a calculator. To operate a calculator, you punch in inputs and you choose a function. You might input 2 and + and 5. The inputs there are 2 and 5. The function is addition. And when you hit =, the calculator performs the addition function on the inputs and gives you an output of 7. SHA-256 takes one input in an empty text box like this:

You type in some stuff. Could be letters, numbers, spaces, punctuation, or symbols. Then you click a button that says something like "Generate", then SHA-256 takes that input and performs its only function, which is using a complex encryption process to turn that input into a garbled series of numbers and letters. This process is called "hashing" and the garbled series of numbers and letters is called a "hash".

Here I input "frog" and run SHA-256 on it:

→ 74fa5327cc0f4e947789dd5e989a61a8242986a596f170640ac90337b1da1ee4

This 74fa5… thing is the actual hash that's generated by "frog". If I input "phantom" instead I get:

phantom

→ 3bb68de60f56156655a2a70e606892edda3e2f2fc57b482bfe3ef1c5263db5b6

Right now, you can google "SHA-256 hash calculator", choose the top result, and play with the algorithm. It's pretty fun.

The first thing to know about this function is that the input can be any amount of words, letters, and symbols. It can be as short as your name. It can be as long as the full text of David Foster Wallace's *Infinite Jest* (including the endnotes).

The second rule is that the same input will always yield the same output. Hashing "frog" and "phantom" will get you the exact hashes above. Hashing all of *Infinite Jest* will always yield the same hash no matter how many times you redo the process. But if you made just one change to the input, like if you went to line 471 and changed "the" to "The", you would get a different hash.

The third thing to know is that this algorithm only works one-way. It isn't like the Caesar cipher, which is easy to crack. I can take any input and get its hash, but I cannot take a hash and get its input. The algorithm cannot be cracked.[3] It is the product of decades of work by people who have dedicated their lives to cryptography and it has been used for a long time by the U.S. military and U.S. corporations.[4] They trust that it cannot be cracked and you probably should too.[5]

---

[3] The current thinking by experts is that reversing this algorithm would require more computational power than you would have if every atom in the universe were a transistor and put toward breaking the encryption. Patrick O'Shaughnessy, "Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best" *The Investor's Field Guide*, published in 2017. http://investorfieldguide.com/hashpower/

[4] Here's an article with more detail on how the algorithm works. Antonio Madeira, "How does a hashing algorithm work?" *CryptoCompare*, published in 2018. https://www.cryptocompare.com/coins/guides/how-does-a-hashing-algorithm-work/

[5] This explanation is drawn largely from Charlie Noyce's explanation in this podcast starting around 31:20. Patrick O'Shaughnessy, "Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best" *The Investor's Field Guide*, published in 2017. http://investorfieldguide.com/hashpower/

Encryption programs are really cool. They take information and hide it by turning it into a specific series of numbers and letters. Encryption is part of what makes a blockchain work. SHA-256 is an encryption algorithm with some helpful properties. And it cannot be broken.

How I'll use cryptography to make my blockchain trustworthy

It's time to start my new blockchain. This process will be more complex and I have to hold off for a bit on explaining why it needs to be this way.

This time I'll add a game to the blockchain based on SHA-256. Here are the rules:

- When we first set up the blockchain, each of the six participants must contribute $5 to what I'll call the Encryption Fund. I'll store this along with the participants' cash in our cash box.
- Any of the six participants can compete to win the prize.
- The prize will be $1 from the Encryption Fund.
- The winner will be the first person to find something they can add to the entry, could be letters, numbers, or symbols, such that the SHA-256 hash of the whole thing starts with five zeros. These letters/numbers/symbols are called the "proof of work."

[entry] [proof of work]

→ 00000...

So if the first entry is "Sean sends $10 to Michael for the movie ticket", then the participants will race to find some proof of work that will cause the entry + the proof of work to generate a hash starting with five zeros. Someone might try adding "sdfjkh" to the next line as the proof of work, like this:

Sean sends $10 to Michael for the movie ticket. sdfjkh

→ 0a76b26b0a3325af1cc22ce465d39fc190695056d3e048f8d56695b47e36eaa9

This hash only has one zero at its start, so this proof of work won't win. But there is some proof of work out there that will win. When someone finally wins it could look like this:

Sean sends $10 to Michael for the movie ticket. 8d892838***(2839

→ 00000ab2dn6b0a29x5a22ce465diod81906958n1d3e048f8d56695b47e36eaa9

As I mentioned above, there's no way to start with a hash and reverse the function to get the hash's input. The only way to find a winning proof of work is by guess and check. We six want that prize, so we will guess and check until one of us wins. Over and over. Some of us might write computer programs that guess many random sequences per second. When one of us finally finds a solution, this serves as proof that they (or their computer program) did an intense amount of work to find the solution. Solving this cryptographic puzzle is called mining. The participants that do it are called miners, so named because they search for the correct proof of work in order to win the right to generate the next block, and get paid for their work. (This process also validates the list, but I'll get to that.)

Once someone finds a suitable proof of work, they broadcast their discovery to all the other participants. The other participants will check that the proof of work actually works. While the proof of work is very difficult to find, it is incredibly easy to verify. All the other participants have to do is hash the entry + proposed proof of work and see that the hash starts with five zeros.

This blockchain will feature an encryption competition using SHA-256 that miners have to win before they can verify a block of entries.

Chaining the blocks together

Now, I'll add a couple more pieces to my blockchain design. It takes time and energy for participants to mine, and I don't want that to slow down the pace of verifying new entries so I'll speed the process up by grouping entries together into blocks (blocks!). Instead of demanding a proof of work for every entry, I'll demand a proof of work for every three entries. So my first block will consist of three entries that Sean, Michael, and Justin texted to the group (I'm skipping the six initial funding entries since they're boring):

Block 1

Sean sends $10 to Michael for the movie ticket. Michael sends $20 to Rob for the saxophone. Justin sends Ethan $5 for the cd.

Now that this block has three entries, the encryption competition begins, and we six race to find a suitable proof of work. Rob finds it and texts the group what it is. Let's say it's "8493kjk" on the next line, so Block 1 with its proof of work is:

Block 1

Sean sends $10 to Michael for the movie ticket. Michael sends $20 to Rob for the saxophone. Justin sends Ethan $5 for the cd. 8493kjk

→ 00000djsk8474jdka918329204ajs24d1738494a243j3k2jh42j3dai8aik8

The other five check and see that it works so we agree that Rob will receive $1 from the Encryption Fund in Block 2. If someone's Block 1 hash does not start with five zeros,

then they must have made some inaccuracy in their list. They must discard their version and copy someone else's.

Matthew and Justin text the group about two new entries, so with Rob's award, our three entries for Block 2 are:

Block 2

Encryption Fund sends $1 to Rob. Matthew sends $1 to Michael. Justin sends $5 to Ethan.

The mining process is slightly different this time. Since we now have more than one block, we can chain (chain!) them together. The first block in a blockchain, called a "genesis block", is a weird one because there's no block before it to chain it to. After the genesis block all blocks can be chained, so the encryption competition changes. From here on out, the rule will be:[6]

- The winner will be the first person to find a proof of work that, when added to the *previous block's hash* and the entries, yields a hash that starts with five zeros.

[previous block's hash] [entry 1] [entry 2] [entry 3] [proof of work]

→ 00000...

So, this time when the encryption competition begins the miners will start with the hash of Block 2, which was 00000djsk8474jdka918329204ajs24d1738494a243j3k2jh42j3dai8aik8.

---

[6] This explanation is largely drawn from a primer written by Mohit Mamoria. Mohit Mamoria, "WTF is The Blockchain?" *Hacker Noon*, published in 2017. https://hackernoon.com/wtf-is-the-blockchain-1da89ba19348

In order to win, the miner needs to hash the Block 1 Hash, the entries, and some proof of work, such that the Block 2 Hash starts with five zeros.

Block 2

00000djsk8474jdka918329204ajs24d1738494a243j3k2jh42j3dai8aik8 Encryption Fund sends $1 to Rob. Matthew sends $1 to Michael. Justin sends $5 to Ethan. [proof of work]

→ 00000...

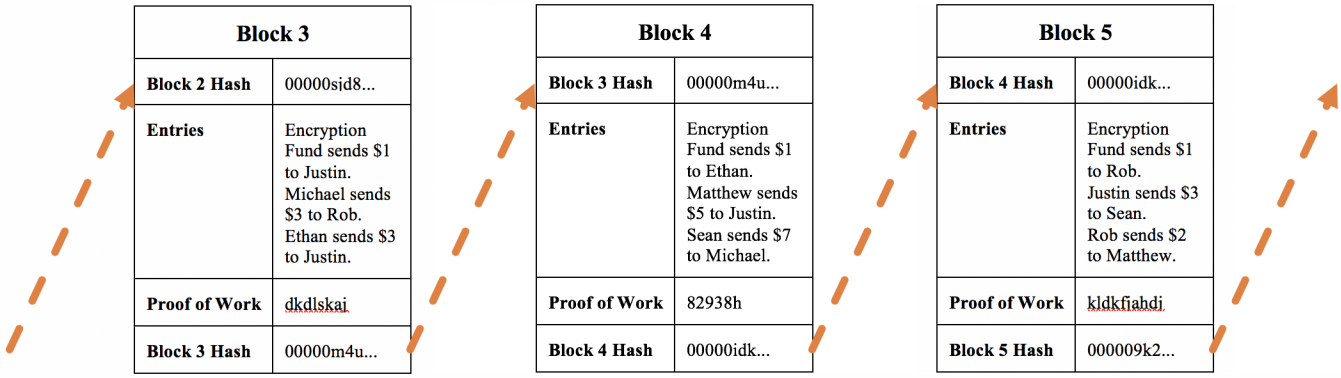
The miners race to find the proof of work. Justin finds it and broadcasts it to the other participants. They check to see that hashing all this creates a hash that starts with five zeros. Again, if any participant's Block 2 Hash does not start with five zeros then they did something wrong and everyone else will know their list is somehow wrong. The miners add this block to their blockchain. When they begin the competition for Block 3, its first entry awards Justin $1.00 from the Encryption Fund.

Block 3

[Block 2 Hash] Encryption Fund sends $1 to Justin. [entry 2] [entry 3] [proof of work]

→ 00000...

This is how the entry and encryption process will work for the remainder of the blockchain, with each block being linked to the one before it by including the previous block's hash. It'll look like this:



Block 3	
Block 2 Hash	00000sid8...
Entries	Encryption Fund sends $1 to Justin. Michael sends $3 to Rob. Ethan sends $3 to Justin.
Proof of Work	dkdlskaj
Block 3 Hash	00000m4u...

Block 4	
Block 3 Hash	00000m4u...
Entries	Encryption Fund sends $1 to Ethan. Matthew sends $5 to Justin. Sean sends $7 to Michael.
Proof of Work	82938h
Block 4 Hash	00000idk...

Block 5	
Block 4 Hash	00000idk...
Entries	Encryption Fund sends $1 to Rob. Justin sends $3 to Sean. Rob sends $2 to Matthew.
Proof of Work	kldkfjahdj
Block 5 Hash	000009k2...

Why the hell are we doing all this? This is a lot of work. Miners are spending time and money to win this competition. Participants are paying into the Encryption Fund to support the mining competitions. This seems like a lot more trouble than a bank account. To see why this might be worth doing, I'll turn now to the evil cheater. It turns out Matthew didn't ruin the first blockchain by accident. It turns out Matthew (who refuses to go by Matt) is a bit of a jerk, and he sets out to game this new blockchain too.

The encryption competition allows us to chain the blocks of our blockchain together so that each block contains elements of the blocks that precede it.

Part III: Why a blockchain is hard to beat

It is very hard to cheat a blockchain by double spending

Matthew first tries to hack the blockchain in the simplest way he can think of: by spending money he doesn't have. Matthew knows he only has $10 on the blockchain. He wants to buy Justin's guitar for $10 and Rob's aquarium for $10. He goes to their houses and retrieves the goods, promising to pay them on the blockchain. Then he texts the group saying he sends $10 to Justin. Then he texts the group saying he sends $10 to Rob. Matthew hopes his two proposed entries will be verified even though he has insufficient funds. This is called double-spending because Matthew is trying to give the same $10 to two different people.

Our text messages are time-stamped, so miners know exactly the chronological order that entries are proposed. Before a miner encrypts a block, they will check that all the entries contained in that block are possible given the funds each participant has. They'll check this because, if they find a proof of work based on an impossible entry, the other miners will reject this proof of work. They'll only be using possible entries, so the proof of work won't yield a five-zero hash for them.

When a miner goes to encrypt the block containing Matthew's first transfer, the one going to Justin, that miner will first check all Matthew's entries on the blockchain to that point to confirm that Matthew has $10 to spend. Seeing that Matthew does have $10, the miner will encrypt the block containing that entry and add it to the blockchain. When someone tries to encrypt the block containing Matthew's second transfer, to Rob, that miner will check Matthew's entries (including the transfer to Justin, since it was time-stamped earlier) and see that he does not have the funds to make the transfer. The miner will ignore that entry and move on to the others.

If Matthew were to somehow propose the two entries at the exact same time, the outcome would be tough to predict but eventually only one entry would persist. Some miners might choose to encrypt the transfer to Justin and ignore the transfer to Rob. Some might choose to encrypt the transfer to Rob and ignore the transfer to Justin. Either way, Matthew will not successfully spend the $10 twice because each miner will only verify one of the two transactions. For a while, there might be two different

versions of the blockchain among us six. Some with the Justin transfer. Some with the Rob transfer. Since the different versions produce different hashes, eventually we would have to reach a consensus on which version is correct. One version will be chosen and one will be forgotten.[7] For Justin and Rob, who are trying to determine whether they were paid, they wait a few blocks for a consensus to emerge.

Spending the same dollar twice on my blockchain isn't possible because proposed entries are time-stamped and miners check that entries are possible before incorporating them.

It is very hard and very expensive to cheat a blockchain by altering a past block

Time passes and our blockchain is thriving. It has 30 transactions grouped into 10 blocks so far. But Matthew decides to try to cheat the system in a new way. He opens his spreadsheet, and goes to Block 7:

Block 7	
Block 6 Hash:	00000ik2...
Entries:	Encryption Fund sends $1 to Michael. Justin sends $32 to Sean. Ethan sends $2 to Rob.
Proof of Work:	amdjfn8gf7aksk
Block 7 Hash:	00000dks...

Matthew changes "Sean" to "Matthew" so that it looks like Justin sent him $32. Now Matthew's Block 7 looks like this:

[7] Sudhir Khatwani, "What is Double Spending & How Does Bitcoin Handle It?" *CoinSutra*, published in 2018. https://coinsutra.com/bitcoin-double-spending/

Matthew's Altered Block 7	
Block 6 Hash:	00000ik2...
Entries:	Encryption fund sends $1 to Michael. Justin sends $32 to Matthew. Ethan sends $2 to Rob.
Proof of Work:	amdjfn8gf7aksk
Block 7 Hash:	0s7daskd78...

Matthew has a problem now. He altered the contents of Block 7. Now when he hashes the Block 6 Hash and the entries and the mined proof of work, he doesn't get a hash that starts with five zeros:

Altered Block 7

00000ik2...
Encryption fund sends $1 to Michael.
Justin sends $32 to Matthew.
Ethan sends $2 to Rob.
amdjfn8gf7aksk

→ 0s7daskd78926ee1a21d2cfcc8b64f0422c44a5b0f747c155104b054834ec

Remember, if anything in the SHA-256 input is changed, the hash will change. Matthew made changes to the input. If Matthew showed his altered Block 7 to any other participant, they would instantly know it was inaccurate because its hash doesn't start with five zeros.

Let's say Matthew anticipates this, so he works very hard and finds a new proof of work for his altered Block 7 that does yield a five zero hash:

Altered Block 7

00000ik2... Encryption fund sends $1 to Michael. Justin sends $32 to Matthew. Ethan sends $2 to Rob. 85j40j4kjkj

→ 000009slol2k4jf882h3h28hsdfjkdfgu834j829239823g2yu3g2u3h2u24u92

Now Matthew's Altered Block 7 looks okay to other miners. They would see he has a Block 7 Hash that starts with five zeros, which looks okay. But Matthew still has a problem. The blocks are chained together, so when he changes his Block 7 Hash, this change ripples into Block 8. Remember Block 8 consists of:

Block 8

[Block 7 Hash] [entry 1] [entry 2] [entry 3] [proof of work]

→ 00000…

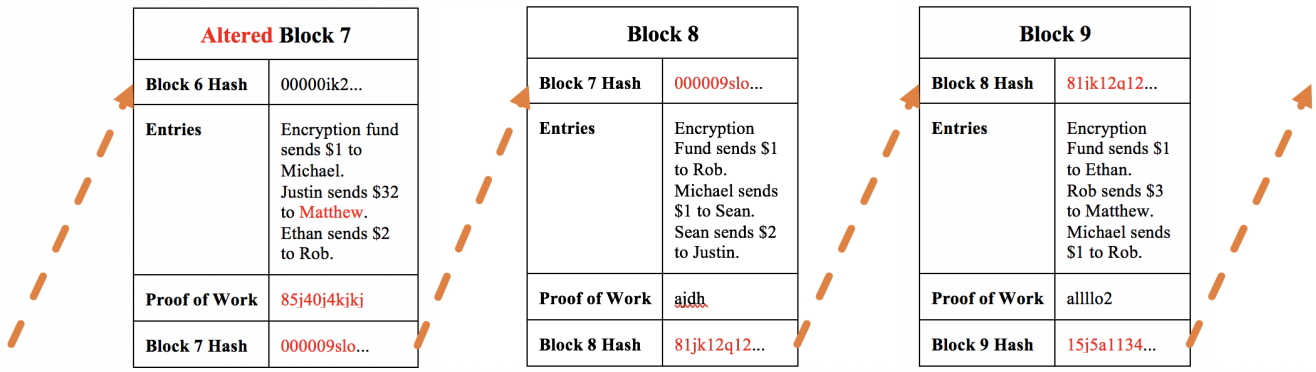
When Matthew found his new Block 7 proof of work, he generated a different Block 7 Hash. Even though it starts with five zeroes, it will still feed into Block 8 and cause a different hash for Block 8.

Altered Block 8

[Altered Block 7 Hash] [entry 1] [entry 2] [entry 3] [proof of work]

→ 81jk12q126637326c8247646583a192ab688d01cadeec4dafc

Block 8 no longer has a five-zero hash. And Block 8 feeds into Block 9. And Block 9 feeds into Block 10. So when Matthew makes one change in Block 7, every subsequent block loses its five-zero hash. Anyone looking at Matthew's blockchain version would see from any block after Block 7 that something was wrong.



Let's say Matthew anticipates the ripple effect, so he works *incredibly* hard and calculates new proofs of work for Blocks 7, 8, 9, and 10. Now his version is altered but each of its ten blocks has a five zero hash. Other participants looking at his altered blockchain and a true blockchain would not know which to trust. Matthew is exhausted but he has finally made his change. And since he's going through the trouble of finding all these new proofs of work, he might as well make a bunch of changes to all those blocks giving himself everything.

This scenario is highly unlikely to happen with a real blockchain because real blockchains have hundreds of miners competing to add blocks at a high rate. Mining is a competition, and competition is fierce. In order for Matthew to build this blockchain and pass it off as true, he would have to work so quickly that he could find proofs of work faster than all the other miners. Now you might better understand why I made my proof of work challenge so hard. If it were easy, Matthew might be able to catch up to the honest miners.

Now you may also see why an older block (i.e. a block with many blocks after it) is a more trustworthy one. The older the block, the more work a wrongdoer would have to do in order to alter it and then catch up to the true blockchain. Not only does this take the wrongdoer time, but it costs them lots of money, since they'll have to pay colossal

electricity bills to run their mining computers at a speed faster than other miners. If participants are confronted with multiple versions of a blockchain, all they have to do is use the longer one since the longest one had the most work put into it, and the work of bad actors is unlikely to ever add up to more than the work of honest actors.

Because of how blocks are chained together, it is very hard and very expensive to cheat a blockchain by altering a past block.

It is very hard to cheat a blockchain by altering a new block

Altering a past block is very difficult for Matthew, but what about cheating by altering a new block? When Block 11 needs to be encrypted, maybe Matthew alters the entries, and is the first to find a proof of work for this altered block. Matthew will broadcast his proof of work, but the other miners will hash Matthew's proof of work along with the unaltered entries they received and find that the hash does not start with five zeros. They will know that Matthew's proof of work doesn't work with the entries, so they will reject it and keep working to find a true proof of work.

Okay, so my blockchain is hard to cheat. It still seems easier to cheat than Bank of America. If Matthew just convinces Justin, Rob, and Sean to go along with his dastardly plan, they can cheat the blockchain since they'd have a majority of mining power. This still doesn't seem worth the trouble of gathering five friends, paying miners, and updating spreadsheets on computers. Things change when I add in a couple new rules that enable more people to join in.

Because of how the encryption competition works, it is very hard to cheat a blockchain by altering a new block.

Part IV: How blockchains get more participants

I'm going to restart my blockchain for a third and final time, and make some changes so it's even more useful. Many of the rules will be the same. I start with five friends who will keep the list and encrypt the entries. I group the entries into blocks of three. I require proposed entries to be time-stamped. I require proofs of work that generate block hashes with five zeros at their starts. But this time I don't collect $20 from each participant to fund their account. And I don't collect $5 from each participant to fund an Encryption Fund.

I don't like that system because it kind of makes me just as bad as the bank. Here are some annoying things about it:

1. I have to keep the money in the cash box safe (and resist the temptation to steal it).
2. I have to go through the hassle of cashing people out when they want to exchange their blockchain dollars for physical dollars.
3. The participants have to trust that I'll keep the cash safe (and not steal it).
4. Eventually, the Encryption Fund will run out and the participants will have to fund it again.
5. Worst of all, this part of the system is a big barrier to potential new participants on the blockchain.

This time around, the participants won't use text messages and their own spreadsheets to track the blockchain. Instead, I write my own computer program that will do everything. It will have a spreadsheet feature to store the blockchain and add new blocks. It will have a messaging feature to enable participants to broadcast proposed entries and proofs of work.

I'll also include a program that awards points to participants in the blockchain. I'll call these points "Tender". I'll design the program so that each time a miner wins the encryption competition, the program will automatically assign them 5 Tender. While I'm at it, I'll design the program to send 1 million Tender to me as the very first entry in

the blockchain, to reward myself for working so hard to set all this up.[8] These points can be transferred between different participants. Eventually, participants might believe these points have some value, and they'll start to think of the points like money. In order to encourage participants to think of these points like money, I'll make Tender scarce by designing the program so that it will only ever generate 10 million Tender.[9]

If participants agree with me that Tender has value, then I no longer need to collect U.S. dollars from those who want to participate. Participants no longer need to pay into an Encryption Fund to pay the miners to keep the blockchain true. Miners will mine so that they can earn Tender.

Wait, why would Tender have value? That question has a complicated answer that I can't fully cover without adding tens of pages to this thing. I'll just say that many people think cryptocurrencies have value and many think they don't have value. Value is kind of like Tinker Bell, from Peter Pan. If enough people believe that it exists, then it kind of does. And vice versa. This is an open question, the controversy of which is evident in the volatility of Bitcoin and Ether's market prices and the general public discourse.[10] For now, it seems like the believers are winning. On September 12, 2018, one Bitcoin was exchangeable for more than $6,000.[11] Moving on.

As I mentioned, the benefits of a blockchain are higher if more people are involved. In my past blockchain, there were just six of us so there weren't identity tracking issues. We used our real names and our permission to control our accounts was proven by the use of our phones. To get more people involved in my new blockchain, I build into the

---

[8] This appears to be how the creator of Bitcoin was compensated for his or her or their work. Rob Wile, "Bitcoin Creator Satoshi Nakamoto May Be Sitting on $5.8B" *Time*, published in 2017. http://time.com/money/5002378/bitcoin-creator-nakamoto-billionaire/

[9] Eventually, this version of the Encryption Fund will run out but by this point the blockchain will be so valuable to participants that they will pay miners to continue mining.

[10] Tae Kim, "Jamie Dimon says he regrets calling bitcoin a fraud and believes in the technology behind it" *CNBC*, published in 2018. https://www.cnbc.com/2018/01/09/jamie-dimon-says-he-regrets-calling-bitcoin-a-fraud.html. Kevin Costelloe, "Bitcoin 'Ought to Be Outlawed,' Nobel Prize Winner Stiglitz Says" *Bloomberg*, published in 2017. https://www.bloomberg.com/news/articles/2017-11-29/bitcoin-ought-to-be-outlawed-nobel-prize-winner-stiglitz-says-jal10hxd. John Kelleher, "Why do Bitcoins have value?" *Investopedia*, published in 2018. https://www.investopedia.com/ask/answers/100314/why-do-bitcoins-have-value.asp.

[11] https://www.coindesk.com/price/

computer program a username and password system. Mine will be very simple and far less secure than the username and password system real blockchains use, but it'll still give you an idea of how this works. When someone downloads the program, they will be instructed to think of a secret password, called a "private key". I dream up a private key that is the phrase "wicked lemons ingratiate Samson burger nickel". The computer program will hash this private key using SHA-256 in order to create a public key, which is my username:

Public Key Generation

wicked lemons ingratiate Samson burger nickel

→ 55875fa7ba5eba7948254c535b7ea5875ff90f41f09a8f8e80d9598ddd1a4f0d

This 55875fa… hash is my public key. When I want to make a new entry on my blockchain, I'll do it through my public key. Let's say I want to send 20 Tender to Michael. Michael told me his public key is 83928ed...:

The Tender Blockchain			
From	**To**	**Item/Amount**	**For**
55875fa...	83928ed...	20 Tender	Saxophone

When I propose this entry to my blockchain, the computer program running it will check that I'm allowed to control what public key 55875fa… does. To do this, it will ask me what the private key is for 55875fa…, and I'll put in "wicked lemons ingratiate Samson burger nickel". The computer program will hash this private key, see that the output is 55875fa…, determine I am allowed to control this public key, and allow me to broadcast the entry to the miners. This is called "signing" the entry.

If Matthew wanted control of my public key, he would need to steal my private key or perform guess and check with different inputs until he found one that generates my public key. Without such an input, he wouldn't be able to sign any entries. Remember,

hashes cannot be reverse engineered, so even though he can see my public key on my entries, he is incapable of guessing the private key needed to sign.

With these additions to the blockchain rules, anyone can download my computer program, generate a public key, and start mining to earn Tender. This means my blockchain can grow to a very large scale. As it does, its benefits are unlocked. And these benefits are, at least in theory, very powerful. I'm done explaining how to set up a blockchain. Now, I'll explain why it's worth setting up.

Cryptocurrencies and public keys allow a blockchain to become independent from existing currencies and existing identification methods, which means many participants can join the blockchain's community.

Part V: Why a blockchain is better than old lists

A blockchain can theoretically do a lot of cool things that a normal list cannot. I'll describe a few, roughly ordered from the more straightforward to the more exotic.

A blockchain retains information better than old lists

Lists can lose information. Recall when I dropped my notepad in the toilet and lost a few pages of entries. Recall when the bank's software glitch caused the bank to forget that I deposited $300. The damage of information loss can be much worse.

Sometimes information is lost because part of the list is destroyed or lost. I want to buy a house for $500,000 from Karen. One fact I should be certain of is that Karen actually owns the thing. I don't want to hand over $500,000, move all my stuff in, get kicked out three months later by the real owner, then struggle to track down the now-disappeared Karen. To avoid this nightmare, I pay someone to get me a list of everyone who has ever owned the house, stretching back to when it was originally granted to the first owner by the government.[12] If this list checks out and shows Karen as the final owner, I feel better but still not amazing. The problem is that this list will go back hundreds of years, over which some combination of fires, spilled coffees, trashed folders, data migrations, and fraudulent transactions might have caused the list to lose information.[13] The only way I can get comfortable with Karen's ownership is by purchasing title insurance, so that if some problem with Karen's ownership does arise, the insurance company will pay to resolve it. In 2014, people paid over $12 billion for insurance against information gaps in these lists.[14]

---

[12] "Chain of Title" *LII / Legal Information Institute*, published in 2014. https://www.law.cornell.edu/wex/chain_of_title

[13] This example might not be totally fair. Blockchain technology isn't competing with lists made in the 1800s with parchment and quills, it's competing with centralized, highly-protected and backed-up lists. But this example does show the pain information loss can cause, so I'm leaving it in.

[14] E.B. Solomont, "Special Report: Inside the title insurance cartel" *The Real Deal*, published in 2018. https://therealdeal.com/2016/03/24/inside-the-title-insurance-cartel/

A blockchain exists on hundreds or thousands or millions of devices.[15] For this reason, it is virtually impossible to destroy or lose its information. If one device explodes, there are many more with their own versions. As more users maintain the list, the power of this safety net grows. And as I covered above, it's very hard for a Matthew to purposely cause information loss.

Typical lists also lose information by overwriting. I have a spreadsheet with a ranking of my favorite five songs by Radiohead. A new album comes out, and I replace "15 Step" with "Little by Little" as my fifth favorite song. I have just lost the information that "15 Step" was once one of my five favorite Radiohead songs because I crossed out an entry and wrote in a new entry. Many lists work this way. Such a list does not contain a complete history of every entry ever made.

A blockchain does contain the complete history of every entry ever made. No information is lost by overwriting. Past entries in the list cannot be changed. One can only add new entries to the list. If something is entered, it won't be forgotten. In database design, this is called an "append only" architecture. In blockchain jargon, it is called immutability.

A blockchain retains information better than old lists because there are many versions of it stored throughout the world and it is never overwritten.

A blockchain is more secure than typical lists

Recall how, at the beginning of all this, I opened a bank account and deposited $300. I can withdraw my money by going to the bank and demonstrating that my name, birthday, and/or address match those of the person listed as the owner of the account. To verify that I have this name and information, I show a bank employee my face, along

[15] The sizes of entire blockchains today are not unwieldy, so they are easy to store. Right now, I could sign up to host the entire history of the Bitcoin blockchain from 2010 to 2017. It would take up around 150 gigabytes of my hard drive. "Bitcoin blockchain size 2010-2017" *Statista*, no publication date. https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/

with some government identification that matches a picture of my face with this information.

A debit card is meant to give me access to my account outside of the bank, which means I have to verify that I am the owner without facial verification. The bank could just say that it's enough proof to have possession of the debit card. This would be a bad system since a non-owner might use the card after I lose it or someone steals it. So the bank adds an extra layer of verification by issuing me a Personal Identification Number ("PIN"). As the owner, I keep the PIN a secret, so that only I can use it to prove that I am the owner of the bank account. Even if someone else gets my debit card, they still can't control my account because they can't prove that they are the owner without the PIN.

The bank has many customers, and they all expect this protection, so they all have been issued PINs. The customers only have to remember their own PIN, so they just memorize them. But the bank has to remember every single PIN of every single customer so that they can match the PINs to the bank accounts. The only way they can remember these PINs is by storing them somewhere. Think of it like a massive list of all the PINs (even though it's undoubtedly more complex). The bank protects this list with the many defenses I mentioned above. It's stored on a server buried six floors underground, behind thick, cement walls and doors that require retinal scans. It's protected by the most sophisticated firewalls money can buy, which are monitored by an army of programmers. It's backed up in many different places across the globe. But even these measures do not fully eliminate the risk that a Matthew will get the list. These measures were not enough to prevent the recent hacks of Equifax, Yahoo!, and JPMorgan.[16] In 2014, before it was hacked, JPMorgan reportedly spent $250 million on

[16] The JPMorgan hackers apparently stole data on 76 million people. Jessica Silver-Greenberg, Matthew Goldstein and Nicole Perlroth, "JPMorgan Chase Hacking Affects 76 Million Households" *The New York Times*, published in 2014. https://dealbook.nytimes.com/2014/10/02/jpmorgan-discovers-further-cyber-security-issues/. The recent Equifax hack might have affected as many as 145.5 million people. Donna Borak and Kathryn Vasel, "The Equifax hack could be worse than we thought" *CNNMoney*, published in 2018. http://money.cnn.com/2018/02/09/pf/equifax-hack-senate-disclosure/index.html. The recent Yahoo! hack might have affected 3 billion accounts. Selena Larson, "Every single Yahoo account was hacked" *CNNMoney*, published in 2017. http://money.cnn.com/2017/10/03/technology/business/yahoo-breach-3-billion-accounts/index.html

cybersecurity and employed more than 1,000 people to guard against this threat.[17] This apparently wasn't enough. And no matter how much an institution spends on protecting its list, a security weakness remains in the design: the institution holds a huge amount of information about its customers/users. When a Matthew gets to it, he gets it all at once. And this doesn't apply just to a bank. This applies to every centralized institution holding information. The Social Security Administration. The hospital. The police station. The university. Facebook. Google. Amazon. They all have lists of information on their users/customers. If their list is compromised, all that information is compromised at once.

A blockchain does not have this security weakness because it is not run by a central institution collecting information on the participants. Participants use a private key to prove ownership of a public key much like they use a PIN to prove ownership of a bank account, but there is no central institution with a list of everyone's private keys like there is with PINs. Verification of public key ownership is done by SHA-256 instead. If Matthew wanted to pretend to be Justin on a blockchain, Matthew would have to find out Justin's private key. The only person in the world that knows Justin's private key is Justin. And if Matthew wanted to pretend to be hundreds of other people, he would have to steal their private keys one-by-one. There can be no simultaneous theft of the information of thousands or millions of participants because there is no entity with a collection of the information of the participants.

A blockchain is more secure than old lists because its design avoids a central list that contains all the participants' private information.

A blockchain is cheaper to use than typical lists

Keeping a list can be costly. In the case of my notepad list, I pay for the notepad. I pay for the pen I use. I spend time making entries and checking that all is in order. I carry the heavy notepad around in my bag. In the case of a bank, it has to pay for its software.

---

[17] Erik Sherman, "Why $250M didn't protect JPMorgan from hackers" *CBS News*, published in 2014. https://www.cbsnews.com/news/why-250m-didnt-protect-jp-morgan-from-hackers/

It has to pay for the server on which it stores the list. It has to pay its army of programmers. Banks and other institutions pass these list-keeping costs on to their customers/users in the form of taxes, fees, and other obligations. If I want to get my driving record from the New York Department of Motor Vehicles, I have to pay them $7.[18] Part of this fee probably goes toward the costs of storing and maintaining my driving record. If I want to open a Chase Total Checking account, I have to keep $1,500 in the account to avoid a monthly service charge.[19] Chase uses my money to make more money, and part of that will help pay for the costs of storing and maintaining the list associated with my bank account.

A blockchain is designed to be low cost. Since the blockchain does not store valuable information in a centralized place, there are no costly security measures. Since the blockchain pays miners with cryptocurrency, participants do not have to pay them to maintain the list.

There is a cost that could be significant in the future. Sometimes a blockchain will get backed up as participants propose entries faster than miners can verify them. This means a participant that wants to send cryptocurrency to someone else will have to wait a long time, which is costly. One way to get around this is to add an incentive system wherein participants pay miners to prioritize their proposed entries. Such payments are called "gas," and they are a fundamental part of systems like Ethereum, another blockchain network. When a blockchain is severely backed up, it is conceivable that gas costs could be significant. It is an open question whether gas costs will prevent the scaling of some blockchains.

A blockchain is cheaper to run than old lists because there is no central list of participants' information that needs to be protected.

---

[18] https://dmv.ny.gov/dmv-records/get-my-own-driving-record-abstract

[19] https://personal.chase.com/personal/checking

A blockchain can automate transactions

So far, I've been talking mostly about lists that are used for remembering things. But many of the lists you know do far more than that. Imagine the software running on your smartphone. Imagine the United States Constitution. These can be viewed as lists that don't just have things to remember, but that contain the rules of a system. Your smartphone's first entry might be a rule saying that when you push the on/off button, the smartphone will start booting up. The Constitution's first entry says that Congress will have the power to create laws. A blockchain's entries can also contain rules. And what makes this even more interesting is that these rules can be automated by computer code.

Rob is a big believer in the car company Tesla and he bet me 20 Tender that Tesla stock will be above $400 on January 1, 2019. I like Tesla's mission and products, but I have concerns about its future, so I agree to the bet. The bet concerns a condition and can result in two possible outcomes. The condition is that Tesla stock is above $400 on January 1, 2019. If this condition is true, I will owe Rob 20 Tender. If this condition is false, Rob will owe me 20 Tender.

I write a computer program that will create a new public key to hold 20 of my Tender and 20 of Rob's Tender, this is the "pool" created by our bet. I first program an initial condition: the smart contract will only become active if it receives the necessary funds from both me and Rob within three days from proposing the smart contract. If it does not, then it will send back whatever funds it received, then shut down. If it determines we have sent the funds within three days, it will move on the main condition. I program it to check the condition at the end of the day on January 1, 2019 by navigating on the internet to Yahoo! Finance. The program will then determine whether the condition is true or false by reading the code of that website. If the condition is determined to be true, the program will send the pool to Rob and, if false, it will send it to me.

I then propose this computer program entry to the miners of my blockchain and the entry is verified and added to the blockchain. Then I send 20 Tender to the program's public key and tell Rob to send his 20 Tender to the program's public key to satisfy the initial condition. We now have a program stored in the blockchain that will automatically run. Rob and I don't have to do anything else for our bet to be executed,

it's all programmed. Without the program, I would have to trust Rob to show up and pay me after he knows he's lost. If I had a standard contract, all I would have is the right to sue him if he didn't show up.[20]

Such an automated program is called a "smart contract". Hey, my legal education actually equipped me with something to say about this name. This name is a little confusing. As defined by legal scholars, a contract is "a promise or a set of promises for the breach of which the law gives a remedy, or the performance of which the law in some way recognizes as a duty."[21] In other words, a contract is (1) a promise that (2) the government will enforce.

Regarding (1), a smart contract isn't really a promise. It's the instructions that carry out a promise. The promises were made when Rob and I agreed to pay each other depending on what Tesla's stock price did, not when we wrote the computer program to execute those promises. While one could deduce the most important parts of our promises by looking at our smart contract, I don't think it's right to say that the smart contract is the promise itself. Maybe an analogy would help. I promise to Venmo my friend $10 because he bought me a beer. I punch the figures into Venmo, and a bunch of computer code that Venmo programmers wrote moves the money. I think it would be wrong to call the Venmo code the promise itself. The code is just how the promise is effectuated. The code does not spell out why I wanted to send my friend $10. A judge couldn't look at the code to determine what I wanted to happen. Maybe the code had a bug causing it to send $100. The judge would have no way of knowing my intent by looking just at the code.

Regarding (2), it is currently unclear whether governments will enforce smart contracts as legally binding agreements. And even if they were legally enforceable, that would be a difficult lawsuit to bring. The parties might be from opposite sides of the world and the suing party would have to prove that the court has jurisdiction over the other party. The other party might be anonymous, and impossible to even find, which happens in blockchain transactions. The transaction might have been an exchange of $1.00 for a

---

[20] Ameer Rosic, "What Are Smart Contracts? A Beginner's Guide to Smart Contracts" *Blockgeeks*, no publication date. https://blockgeeks.com/guides/smart-contracts/

[21] "Second Restatement of Contracts" *JSTOR*, originally published in 1981. https://www.jstor.org/stable/27876768?seq=1#page_scan_tab_contents

baseball card. Transactions for small amounts is a very popular use case for blockchains. Who would pay the cost and spend the time to take this to court? If the court in your jurisdiction won't enforce the smart contract, then it is only as useful as far as the parties trust each other to honor it, and one of the major reasons to use a blockchain is so that users don't have to trust each other. (By the way, this is the problem that my company, Juris, is trying to solve. I'll get into that briefly in Part VII.) Anyway, despite the confusion this name causes, it has stuck, so you should know it.

Another limitation of smart contracts today is that they only really work when a condition can be determined by a computer to be true. Some things are easily shown to be true.[22] My Tesla bet relies on Yahoo! Finance to determine the condition, and that seems like a pretty reliable source. In blockchain jargon, Yahoo! is the "oracle" I'm using to determine the truth. But even this condition can be a tricky one to determine. Yahoo! might change its website such that my program no longer knows where to look to check the price. Rob might hack Yahoo! and change the price to be above $400 on January 1, 2019. Imagine a smart contract meant to send an mp3 file to a user upon receiving a certain amount of cryptocurrency. What if the buyer accidentally signed the entry. The oracle would say that the money came in, so the condition is true, but if the buyer lacked the intent to send the money, was the money really sent? These complexities arise from a seemingly easy condition to prove as true. Imagine more complex conditions. How can you write code that can determine whether it is true that a party has made "reasonable best efforts," a term contained in a huge number of modern contracts?[23]

Okay, so those are some reasons why a blockchain might be a better way of storing information. Now, I'll get into a specific application that tries to unlock those benefits.

Blockchain entries can feature little computer programs that automate transactions between users.

---

[22] This analogy comes from this article. Ameer Rosic, "What Are Smart Contracts? A Beginner's Guide to Smart Contracts" *Blockgeeks*, no publication date. https://blockgeeks.com/guides/smart-contracts/

[23] Perhaps in the future we'll have technology that predicts with high accuracy what a majority of human beings would consider reasonable. But we aren't there yet.

A blockchain can be designed to reward participants with cryptocurrency for improving the design of the blockchain

A list of information is helpful. But when a list can pay participants to improve the design of the list, it becomes something far more special. It becomes a self-improving community/machine that will innovate at a rapid pace.

When I build my blockchain computer program I could also add another part to it. This program would act as a forum, in which participants can post comments. This program would also allow for voting on comments. Maybe I make it a rule that participants get one vote for every one Tender they hold. I could make it a rule that if a proposal receives "yes" votes from the holders of the majority of issued Tender, the proposal goes into effect and the rules of the blockchain's computer program are altered accordingly. I could also add a rule saying that the participant who proposes a change which is approved by a majority shall be rewarded with 20 Tender.

This feature of a blockchain is promising because it means that the participants are incentivized to improve the blockchain. The pace of this innovation might outstrip that of corporations and governments. Within a corporation, employees are incentivized to improve the company's product with money and promotions. On a blockchain, the participants can be incentivized to improve the service with the currency that the blockchain uses. The participants don't have to get a job to make these improvements. Anyone can add anything, and if its valuable, then they'll receive compensation. This comparison of a corporation to a blockchain community can be described as a permission-based solution versus a permission-less solution. As Naval Ravikant, a prominent blockchain commentator puts it, "Permission-less networks out-innovate permission networks by a factor of a hundred or a thousand. It's not even close."[24] Think Encyclopedia Britannica versus Wikipedia. Which is more useful?

A blockchain community can be incentivized with cryptocurrency to improve the blockchain's design, so it might improve at a very rapid pace.

[24] This comes from Naval Ravikant on a podcast starting around 13:00. Patrick O'Shaughnessy, "Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best" *The Investor's Field Guide*, published in 2017. http://investorfieldguide.com/hashpower/

Part VI - What a blockchain can be used for today

Over the course of this paper, I've spent most of the time describing a blockchain that's used for the storing, sending, and receiving of payments of money. This is what the Bitcoin blockchain does, and it is one of the more obvious uses for a blockchain. It's an alternative to a bank because all a bank really has anyway is a list of transactions. But there are other, more interesting uses for a blockchain. I'll detail a couple now.

A blockchain can be used to track ownership of digital property

In the United States today, it is relatively easy to determine who owns what physical property, and this ownership is enforced by the government. If Marc owns farmland and Pete moves onto it, Marc can call the police, show them his state-certified document proving ownership, and the police will forcibly eject Pete from the property. Similarly, if Marc grows corn on his property and Pete steals some of Marc's corn, Marc can call the police, have Pete charged with theft, then sue for the value of that corn by proving Pete stole it. Millions of people want that corn. The United States government excludes all of them, except Marc, from having it.

In the United States and many other countries, you can exclude others from using the things you own because (1) lists keep track of who owns what and (2) the government will physically enforce what these lists say if necessary. You know this already. The reason it's worth outlining is because this system, which is the one human beings have used for millennia to organize their societies, doesn't work so well with digital property.[25]

Consider a recording of a song. This recording of a song is owned by an owner who wants to control that recording. Millions of people want that recording, but the United States government excludes all of them, except the owner, from having it. In the early 20th century, a duplicate recording of a song was expensive to make, expensive to copy, and expensive to distribute. As a result, the number of copies was low. It was as easy to track the copies as it was husks of corn. It was easy for the owner to determine when it

[25] Of course, other systems of ownership exist. Systems for ownership of intellectual property have existed for centuries. But these systems also cannot be simply applied to digital property for reasons I will give.

was stolen, to prove ownership to the government, and to have the government enforce the owner's rights.

Then recording technology improved. Eventually, these costs came down, which raised the volume of available copies, which made it more difficult for owners to track who was using the recording and to get the government to enforce their rights. This trend crescendoed with the cassette tape, then CD-ROM, then mp3. Mp3 technology made duplicating a recording virtually costless, and lossless, meaning the copy and the original are 100% identical in every way.[26] One only had to copy the 1s and 0s in the right order to create the recording, just as they would copy the right letters in the right order to produce a book. The property was digitized. Then Napster and its progeny made distribution of that recording virtually costless.[27] The owners scrambled to create laws that protected their rights but the government was unable to enforce these laws against the vast majority of participants. Participants multiplied and disseminated the digitized property too quickly and too broadly. The participants were content to break these laws because they knew they were unlikely to get caught and they were tired of paying $12 for a CD with only one good song on it. Today, it is estimated that more than one-third of internet users consume unlicensed music content.[28]

This breakdown of the traditional property system has occurred for many forms of digitizable property: books, movies, photographs, software. Anything reducible to a series of 1s and 0s can be copied and distributed at no cost. While the owners of these properties were upset by the failure, the consumers rejoiced. Now everyone (sort of) can have everything (sort of) for free (sort of). Which I think is mostly a really great thing. The problem is that the makers of digital property still want to get paid for what they make, which I think is fair. This problem is solvable because many consumers agree that it's fair and are willing to pay a fair price.

---

[26] "Home Taping Is Killing Music" *Wikipedia*. https://en.wikipedia.org/wiki/Home_Taping_Is_Killing_Music

[27] This explanation is derived in part from Jeremiah Lowin's in this podcast starting around 5:35. Patrick O'Shaughnessy, "Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best" *The Investor's Field Guide*, published in 2017. http://investorfieldguide.com/hashpower/

[28] "The recording industry's ability to develop the digital marketplace is undermined by piracy" *IFPI*, no publication date. http://www.ifpi.org/music-piracy.php

Blockchain technology can't be used to stop the illegal copying and distributing of digital property. As far as I can tell, consumers will always be able to do this. What we can do with blockchain technology, however, is reduce this activity by creating a better product. Because of transactional efficiency, a blockchain can be used to sell legal copies of digital property at a price much lower than the one sellers are currently offering and make sure more of that money ends up in the right place (wherever we agree that might be).

The existing property system doesn't work so well when applied to digitized property like music and books.

The popular ways to prevent digital property theft are flawed

The entities most interested in preventing music piracy are those companies that enable the exchange of music and money between musicians and consumers. Musicians probably want to prevent piracy but they mostly just want to make music and make a living. Consumers probably want to pay musicians for their work but mostly just want to listen to music. The intermediaries are the worried ones because all they do is sell the music and take a cut. They can only make money if they can establish who owns what digital property and can get consumers to pay these owners.

The dominant music service today in the United States is Spotify. Spotify's founder, Daniel Ek, has stated that his service "was designed from the ground up to combat piracy" by building "a service which was better than piracy".[29] Ek recognized that piracy could not be stopped, but it could be outperformed. His theory was that people would be willing to pay money (or tolerate advertisements) for a service if it dependably and quickly delivered recordings of the highest quality, along with other features like social networking and playlist building. His theory seems proven out.

[29] Andy, "Spotify Was Designed from the Ground Up to Combat Piracy" *TorrentFreak*, published in 2013. https://torrentfreak.com/spotify-was-designed-from-the-ground-up-to-combat-piracy-131204/

Still, the music industry is troubled. Musicians like Taylor Swift are unhappy with their payments, stating "valuable things should be paid for."[30] If they are underpaid, it seems like intermediaries are to blame. Record labels want their cut. Licensing agencies want their cut. Performance rights organizations want their cut. These intermediaries create costs that are passed on to users and that come out of musicians' compensation. The model also has the weaknesses of a centralized system. Spotify gathers users' data and stores it in a centralized place that could be hacked. Spotify sells users' data to advertisers, perhaps in violation of their desires for privacy.

A blockchain might be used to create a new music distribution service that outperforms Spotify.

Many companies have tried to find new ways to enforce property rights over digitized property, but their solutions are flawed.

A system using a blockchain might be a better way to sell digital property

Here's how this might work. Sarah Stevens is a musician and she wants to sell a recording of her song, called "Blackbird", to others. There exists a blockchain-based music service called Musichain, which runs using a cryptocurrency called Note. Sarah signs up for the service by creating a public key. She generates her public key by hashing with SHA-256 the private key she came up with, "sold my cold knot a heavy stone".

Musichain Public Key Generator

sold my cold knot a heavy stone

→ 58c747fd240f8ffff4a5a14b24949983514f3c2e98a1dde4f223b2e1c6d7ea01

---

[30] Taylor Swift, "For Taylor Swift, the Future of Music Is a Love Story" *The Wall Street Journal*, published in 2014. https://www.wsj.com/articles/for-taylor-swift-the-future-of-music-is-a-love-story-1404763219

This is her public key on Musichain. Now she is ready to propose an entry to the Note miners. With her first entry, she will claim her artist name on Musichain. She proposes an entry like this and signs it with her private key:

Musichain	
Public Key	**Action**
58c747fd240f8ffff4a5a1...	Claiming Artist Name: Sarah Stevens

The miners check all the Musichain entries to verify that no public key has ever tried to claim the artist name of “Sarah Stevens”. They see that no one has, so they include the entry in their next block. Now Sarah’s public key controls the artist name “Sarah Stevens”. If anyone else ever tries to claim that name on the Musichain, the miners will see Sarah’s public key has claimed the name, and will ignore the proposed entry.

Now Sarah proposes a new entry that contains an audio file of her recording of Blackbird. The entry claims that her public key is the owner of the recording:

Musichain	
Public Key	**Action**
58c747fd240f8ffff4a5a1...	Uploading and Claiming Recording: blackbird.mp3

This time, the miners check all the Musichain entries to verify that no public key has ever tried to claim ownership of this audio file. They confirm no one has, so they include her entry in their next block. Now Sarah’s public key is recorded as the owner of the recording of Blackbird. If anyone else ever tries to upload and claim this recording, their entry will be ignored by the miners.

Now Sarah proposes a new entry which announces that she is selling infinite copies of her recording to all Musichain users. This entry contains a smart contract, with its own

public key, to automatically do so. This entry sends the Blackbird audio file to that smart contract's public key. The smart contract is programmed to check all the new verified entries on Musichain every second. The smart contract is looking for new entries sending 5 Notes to Sarah's public key and indicating that the sender wishes to buy a copy of the Blackbird audio file. If those conditions are satisfied, then the smart contract sends a copy of the Blackbird audio file to the buyer's public key. This entry looks like this:

Musichain	
Public Key	**Action**
58c747fd240f8ffff4a5a1...	Announcement: Selling Blackbird for 5 Tender Send: Blackbird.mp3 to d0665bba31dcc96a4c03… (this is the smart contract's public key) Smart contract: if any public key sends 5 Notes to 58c747fd240f8ffff4a5a1… and includes the message "for one copy of Blackbird", then send blackbird.mp3 to that public key

At this point, we have a way to upload, claim ownership of, offer, and sell audio files. There would have to also be a program that works like Spotify, that actually allows users to navigate through the musicians' songs, decide what to play, and play the songs. I'll call this the Musichain Player. This Player will operate by looking to the Musichain to see what users are authorized to do.

James is a music listener on Musichain. He has a public key and has authorized the Player to make certain entries to Musichain on his behalf. He saw a Sarah Stevens show recently and liked the music, so he finds Blackbird in the Player, sees that one copy costs 5 Notes, and determines to buy a copy. When he clicks the "buy" button, a whole process occurs on Musichain and the Player. First, the Player generates and proposes a new entry to the Musichain using James' public key.

Musichain	
Public Key	**Action**
9345a35a6fdf174dff721...	Purchase: One copy of Blackbird Send: 5 Notes to 58c747fd240f8ffff4a5a1...

Sarah's smart contract is scanning the new Musichain entries and it notices James' entry, which satisfies its condition. It generates the following entry:

Musichain	
Public Key	**Action**
d0665bba31dcc96a4c03...	Sale: One copy of Blackbird Send: blackbird.mp3 to 9345a35a6fdf174dff721...

Now, Musichain shows that James' public key owns one copy of the Blackbird preview. James will see in the Player that he is authorized to listen to Blackbird once. When James hits the "play" button, the Player starts playing the audio file for James and generates a new proposed entry on behalf of James' public key.

Musichain	
Public Key	**Action**
9345a35a6fdf174dff721...	Send: blackbird.mp3 to 992jjk32l2jk3l2j3h1k3…

This part is weird (this is all weird), but this entry effectively causes James to throw away his copy of Blackbird. The address he is sending his copy of Blackbird to is a public key that's controlled by no one. After he does this, Musichain will reflect that he no longer owns a copy of Blackbird. If he tried to play it again in the Player, it would check Musichain, see he was no longer authorized to play the song, and would not allow it. Each time James buys a copy and plays the song, this process is repeated. And it all happens in under a second.

So should we use a blockchain for this? The answer to this question is really hard to determine but I'll try to sketch it out a bit with some simple math.

Musichain might be cheaper for consumers than Spotify. One Forbes article says that, in 2017, Americans listened to an average of 32 hours of music per month.[31] If I assume a song is 3 minutes long, that's 640 songs per month. Spotify Premium costs $10 per month.[32] If I divide $10 by 640 songs, that's $0.0156 per stream. So let's say an average consumer is paying Spotify $0.0156 per stream.[33] Let's say Musichain artists charge $0.01 per stream. That would make Musichain 36% cheaper for consumers than Spotify. But does this $0.01 price work for the artists?

Musichain might be more lucrative for musicians than Spotify. In 2015, Spotify's average "per stream" payout to musicians was somewhere between $0.006 and $0.0084.[34] If an artist gets the low end of that scale from Spotify, they'd make 66% more charging $0.01 on Musichain. If an artist gets the high end of that scale from Spotify, they'd make 19% more charging $0.01 per stream on Musichain. The total payouts to artists would also be higher than these figures since the artist wouldn't have to pay intermediaries to get their music on Spotify.

Even if Musichain isn't cheaper than Spotify and more lucrative for artists, it has other benefits. Musichain might make consumers happier than Spotify since they'll know their money is going directly to the musicians instead of being split among intermediaries. Musichain might also innovate far faster than Spotify, since all participants could have the ability to improve it as I mentioned in Part IV. As one

---

[31] Hugh McIntyre, "Americans Are Spending More Time Listening to Music Than Ever Before" *Forbes*, published in 2017. https://www.forbes.com/sites/hughmcintyre/2017/11/09/americans-are-spending-more-time-listening-to-music-than-ever-before/#5ea6101e2f7f

[32] https://www.spotify.com/us/premium/

[33] I could add that Spotify isn't a profitable business yet, so this price might be unsustainably low. It lost 394 million euro in 2018 Q2. https://www.sec.gov/Archives/edgar/data/1639920/000156459018017613/ck0001639920-6k_20180726.htm#Item_1_Financial_Statements

[34] Lizzie Plaugic, "Spotify's Year in Music shows just how little we pay artists for their music" *The Verge*, published in 2015. https://www.theverge.com/2015/12/7/9861372/spotify-year-in-review-artist-payment-royalties.

example, if someone created a strong playlist on the Musichain Player, they could receive “tips” from other participants.

A startup called Voise is currently building a blockchain like the one detailed above, though it plans to use a downloading model instead of a streaming model.[35] Musicoin is also working on such a blockchain.[36]

While I focused on a blockchain for music distribution in this example, this same structure could be used for other digital property: books, movies, video games, television, etc. The costs of third-party intermediaries could be eliminated for all these products, possibly making them cheaper for consumers and more lucrative for producers.

A blockchain might be used to sell digital music in a way that is cheaper for consumers and more lucrative for music producers.

A blockchain might be a better system for controlling physical locks

Digital property used for entertainment is a tricky example due to the piracy dynamic. Everyone wants to see/hear the digital property and can only do so if they get a copy of it and once they have a copy it's hard to keep them from making copies for free. Ownership enforcement is hard and blockchain technology doesn’t change that.

I’ll quickly detail a more straightforward use for a blockchain. This blockchain will be used to enable control over the locks on people’s front doors. I’ll call it bLockchain and it will use a cryptocurrency called Pin. To participate, you buy a physical lock for your house’s front door that can connect to the Internet and that has a keypad. You then download the bLockchain computer program. You create a public key for the bLockchain by hashing a strictly numerical private key, like “473827”. You then create a new entry in the bLockchain, assigning this physical lock an identifier code and

[35] https://www.voise.com/.

[36] https://musicoin.org/for-listeners.

associating your public key with it as the owner. Miners verify that no previous entry has claimed to be the owner of this physical lock. They are rewarded with Pins for their verification.

You set your door to lock when it's closed, then you close it. To unlock it, you input your private key using the keypad on the lock. The Internet-connected lock has a program that hashes your private key to verify that you are allowed to control your public key. The program then checks the bLockchain to verify that your public key is allowed to unlock the door. The program sees that your public key is the owner of the lock, so the door unlocks. If anyone input anything other than your private key, the door would not unlock because the lock's program would check the bLockchain and see that public key was not authorized to control it. This can get more interesting.

You want your mother-in-law to be able to unlock the door. You don't need to give her your private key, and all the powers of ownership that you have. Instead, you tell her to create her own public key for the bLockchain using a numerical private key. Then you make a new entry to the bLockchain associating her public key with the physical lock as an "approved user". And maybe you limit her use a little. Maybe you want to prevent weekend, unannounced drop-ins so you make a rule that she can only unlock the door during the week. If she tries to unlock the door on a Saturday at 8:00AM, the program will check the bLockchain, see she is only allowed access during the week, check the date and time, and refuse her entry.

You want to rent out your basement to make some more money. The basement only has one door and it leads outside. You install a bLockchain lock on the door, make an entry establishing your public key as its owner. Your tenant creates a public key and you make a new entry associating him as an "approved user". In the tenant's lease agreement you have a provision stating that if he fails to pay rent for a month, you can evict him by force. The tenant fails to pay rent for a month and you go into the bLockchain and make a new entry disassociating him from the lock, so he has no ability to unlock the door.

You want to put a smart lock on the door to your office. You want a smart lock to unlock your car. You don't need to memorize new passwords for these new locks. Instead, you make two new entries on bLockchain claiming ownership of these locks by

your public key. Now you can unlock both of them using the same private key you used on your front door. You can buy 1,000 different locks and you only ever need one password, your private key. Of course, the drawback here is that if someone steals your private key, they now have access to every smart lock to which you have access.

Anyway, this kind of product already exists and it operates without using a blockchain. The reasons this solution might be worth creating with a blockchain are those I've already covered. A blockchain solution might be cheaper. A blockchain solution might be more secure, since there is no central lock company with a list of everyone's passcodes that could be hacked.[37] A blockchain solution might be better at retaining information so it never forgets who is allowed to control what. And the service could innovate at a rate far greater than that of a private lock company with employees.

A blockchain could be used to track ownership and use of digital keys to physical locks. This could be cheaper and more efficient than current methods.

[37] The idea for such a key was mentioned by Jeremiah Lowin on this podcast starting around 5:35. I have drawn out this idea into how that product might look. Patrick O'Shaughnessy, "Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best" The Investor's Field Guide, published in 2017. http://investorfieldguide.com/hashpower/

Part VII: Why blockchain communities might be better than firms

Now, I'll briefly zoom up to a really high level and pose an interesting idea: blockchain communities might be better at accomplishing some things than firms are.

People want good and services and they want to buy them with money. People make and provide goods and services and they want to sell them for money. Consumers rarely organize into groups but producers almost always do. Today, the predominant structure by which these producers organize is a firm, whether that's a corporation, partnership, LLC, whatever. We view it as a normal thing that producers are organized into firms but this isn't some mandated thing. Mark Zuckerberg didn't have to form a corporation and sell shares and hire employees. He could have just made some limited agreements with some workers to build his platform. He could have made some limited agreements with financiers to give him money to pay the workers. Instead he created a legal entity called a firm. The firm hired employees and employees made the website. And the advertisers pay for space on the website, and the advertising money goes to the firm. And the firm pays the employees their salaries and gives out the profits to the shareholders. Why is it done this way?

According to economist Ronald Coase, organizing into a firm is more efficient than making a bunch of limited agreements. In his famous 1937 paper, "The Nature of the Firm," he theorizes that firms exist because hiring employees is cheaper in terms of transaction costs (costs of search, bargaining, and cooperating between transacting parties) than using contractors.[38] Employees are cheaper than contractors because they can be trusted at lower cost. But the firm also has inherent design flaws. There are principal-agent costs that firms pay, i.e. the costs that arise when employees don't act in the best interest of the firm's owners. These costs can be very high.

---

[38] Jeremy Liu, "Blockchain, Decentralisation, and the 'Theory of the Firm'" *Medium*, published in 2017. https://medium.com/the-pointy-end/blockchain-decentralisation-and-the-theory-of-the-firm-92649c62350d

The firm is probably one of the most important inventions of humanity. It's a great way for producers to get together to build something. But a blockchain is also a way of organizing people to build something, which begs the question: is it a better way?[39]

Think again about Facebook.[40] Facebook is a corporation. It connects users and advertisers. Users pay for the service with their attention and advertisers pay for the service with their money. The design of Facebook is limited by the innovativeness of its employees, since non-employees cannot change how the service works. Many Facebook shareholders don't get to meaningfully participate on governance of the corporation. When Facebook is used for bad purposes, users cannot do anything about it but protest and leave. And users cannot control what Facebook does with their data.

Imagine a blockchain-based Facebook. It operates via open source code and new entries are verified by miners. These miners make sure, for example, that users cannot change a profile that is not their own. Users create and update their own profiles. Other users can view others' profiles. The innovation would be limitless, since users can propose changes to the rules of the blockchain. Payment for the service could be far more efficient. Instead of being bothered by advertisements, users could pay micro-amounts to miners each time they use the service to consume others' posts. Users could also earn micro-amounts each time their post adds value to the service.

As I mentioned in Part IV, computer programs can be thought of as very long lists of information and rules. In the case of an iPhone, when you turn it on, you are telling it to start reading the information and rules it contains and to execute them. You might think of all organizational structures this way, as a list of information and rules that the organization executes. The promise of blockchain technology is that it will allow the users of products and services, and maybe even the citizens of governments, to write the

---

[39] Jeremy Liu, "Blockchain, Decentralisation, and the 'Theory of the Firm'" *Medium*, published in 2017. https://medium.com/the-pointy-end/blockchain-decentralisation-and-the-theory-of-the-firm-92649c62350d

[40] This final application is inspired by a new book called "New Power," by Jeremy Heimans and Henry Timms, in which they discuss Facebook's structure. I haven't read the book. I only read this article about it and used their quotations as a jumping off point. Kevin Roose, "Can Social Media Be Saved?" *The New York Times*, published in 2018. https://www.nytimes.com/2018/03/28/technology/social-media-privacy.html

rules themselves instead of putting up with those of centralized intermediary entities that charge them too much and care about them too little.

A blockchain might be a better way than a firm to organize a group of people to accomplish a task. It might be cheaper, more innovative, more private, and more fair.

Part VIII: The problem Juris is solving

Here, the paper ends with a brief explanation of what my company is doing to improve blockchains. I first wrote this paper so I could graduate from law school and learn about blockchains. I improved it and lengthened it so I could get readers interested in helping out with Juris' mission. So here's what Adam, Konstantin, Saul, John, and I are working on.

Many smart people think that blockchain technology will change the world for the better and in many ways. They predict that in five or ten or twenty years, it will be used to make things cheaper, faster, more secure, and more fair. But this potential won't be unlocked until a few different problems are solved, one of which I mentioned in Part V: the enforceability of agreements on blockchains. The legal systems we have today are not suited for the speed, volume, and low cost of blockchains. They are slow. They are expensive. They are disjointed, with their jurisdictions determined by their governments' borders. In many developing countries, they are corrupt and unfair. Millions, if not billions, of people don't have access to a legal system that they can use to enforce their agreements. And if an agreement can't be enforced, it just won't work.

Juris is building a new legal system that is more open, more accessible, more affordable, and more fair than any the world has seen. We're building a legal system that will make agreements on blockchains enforceable. Our users are people that want to make agreements and people that want to help structure and support those agreements. A person that wants to make an agreement goes to Juris, finds the appropriate legal agreement template, finds a smart contract to automate it, customizes it with the help of other users, signs it, deploys it to a blockchain, monitors it through its life, and if any dispute arise, has it resolved by a neutral arbitrator on Juris.

This is the problem we see and the solution we're building. If you're interested in how we're doing this, you should sign up at jurisproject.io. There you can find our white paper, detailing how we propose to do this. You can find information on our team and our roadmap and our values. You can join a community of like-minded people that also want to build a new legal system that makes enforceable agreements as accessible as the internet.

Conclusion

If you read this far, you should be proud. Especially if you were new to blockchain technology. We've been storing information the same way for so long that it's difficult to learn a new way. It's harder still to learn a new way of organizing human activity.

Hopefully, reading this paper is just one, early step in your blockchain education. There's still so much more to learn and to discover. Blockchain technology is very young. And we really don't know exactly what it's good for. We know that the Bitcoin blockchain is enabling payments across the world, and that on October 11, 2018, all the Bitcoin had a total value of over $100 billion. We know that the Ethereum blockchain is operating a decentralized, world computer, and on that same day, all the Ether had a total value of over $20 billion. We also know that, in recent years, blockchain technology has captured the attention of many of the smartest and most passionate people on Earth, united by a fervor like that of the early internet pioneers.

We really don't know exactly what this technology is good for, but if more smart people get involved, we'll figure it out faster. And it's my hope that by figuring it out, we can help hundreds of millions of people achieve a higher standard of living. More knowledge. More prosperity. More privacy. More fairness. More freedom.

Contact

Maybe you think there's a gaping hole in this paper. Maybe I got something totally wrong. Maybe you just want to let me know I suck. Maybe you have something nice to say. Either way, email me at ethan@jurisproject.io. I want this thing to get better and I'd appreciate your help. Thanks!

Sources

Andy “Spotify Was Designed from the Ground Up to Combat Piracy” *TorrentFreak*, published in 2013.
https://torrentfreak.com/spotify-was-designed-from-the-ground-up-to-combat-piracy-131204/

Tyler Babich, “Atlanta's Government Is Shutdown By Ransomware” MDCHHS, published in 2018.
https://www.mdchhs.com/atlantas-government-is-shutdown-by-ransomware/

Conrad Barsky & Chris Wilmer, “The Blockchain Lottery: How Miners Are Rewarded” *CoinDesk*, published in 2014.
https://www.coindesk.com/blockchain-lottery-miners-rewarded/

Donna Borak and Kathryn Vasel, “The Equifax hack could be worse than we thought” *CNNMoney*, published in 2018.
http://money.cnn.com/2018/02/09/pf/equifax-hack-senate-disclosure/index.html

Kaj Burchardi & Nicolas Harle, “The blockchain will disrupt the music business and beyond” *Wired*, published in 2018.
https://www.wired.co.uk/article/blockchain-disrupting-music-mycelia

Kevin Costelloe, “Bitcoin 'Ought to Be Outlawed,' Nobel Prize Winner Stiglitz Says” *Bloomberg*, published in 2017.
https://www.bloomberg.com/news/articles/2017-11-29/bitcoin-ought-to-be-outlawed-nobel-prize-winner-stiglitz-says-jal10hxd

Euny Hong, “How Does Bitcoin Mining Work?” *Investopedia*, published in 2018.
https://www.investopedia.com/tech/how-does-bitcoin-mining-work/

Preethi Kasireddy, “How does Ethereum work, anyway?” *Medium*, published in 2017.
https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369

John Kelleher, “Why do Bitcoins have value?” *Investopedia*, published in 2018.
https://www.investopedia.com/ask/answers/100314/why-do-bitcoins-have-value.asp

Adam J. Kerpelman, Ramsay Brown, Akash Desai, Matt Mayberry, Jake Dodd, Vince Enachescu and T. Dalton Combs, “Juris Adjudication Protocol: Human-Powered Dispute Resolution for Smart Contracts” published in 2018.
https://www.gitbook.com/book/juris/whitepaper/details

Sudhir Khatwani, “What is Double Spending & How Does Bitcoin Handle It?” *CoinSutra*, published in 2018.
https://coinsutra.com/bitcoin-double-spending/

Tae Kim, “Jamie Dimon says he regrets calling bitcoin a fraud and believes in the technology behind it” *CNBC*, published in 2018.
https://www.cnbc.com/2018/01/09/jamie-dimon-says-he-regrets-calling-bitcoin-a-fraud.html

Matthew Leising, “The Ether Thief” *Bloomberg*, published in 2017.
https://www.bloomberg.com/features/2017-the-ether-thief/

Selena Larson, “Every single Yahoo account was hacked” *CNNMoney*, published in 2017.
http://money.cnn.com/2017/10/03/technology/business/yahoo-breach-3-billion-accounts/index.html

Jeremy Liu, “Blockchain, Decentralisation, and the 'Theory of the Firm'” *Medium*, published in 2017.
https://medium.com/the-pointy-end/blockchain-decentralisation-and-the-theory-of-the-firm-92649c62350d

Antonio Madeira, “How does a hashing algorithm work?” *CryptoCompare*, published in 2018.
https://www.cryptocompare.com/coins/guides/how-does-a-hashing-algorithm-work/

Mohit Mamoria, “WTF is The Blockchain?” *Hacker Noon*, published in 2017.

https://hackernoon.com/wtf-is-the-blockchain-1da89ba19348

Hugh McIntyre, “Americans Are Spending More Time Listening to Music Than Ever Before” Forbes, published in 2017.
https://www.forbes.com/sites/hughmcintyre/2017/11/09/americans-are-spending-more-time-listening-to-music-than-ever-before/#5ea6101e2f7f

Hugh McIntyre, “Why Did Taylor Swift Really Rejoin Spotify?” *Forbes*, published in 2017.
https://www.forbes.com/sites/hughmcintyre/2017/06/27/why-did-taylor-swift-really-rejoin-spotify/#383a05a8373d

Patrick O’Shaughnessy, “Hash Power – Episode 1 – Understanding Blockchains Invest Like the Best” *The Investor’s Field Guide*, published in 2017.
http://investorfieldguide.com/hashpower/

Lizzie Plaugic, “Spotify's Year in Music shows just how little we pay artists for their music” *The Verge*, published in 2015.
https://www.theverge.com/2015/12/7/9861372/spotify-year-in-review-artist-payment-royalties

Ramon Recuero, “The Decentralized Future Series” *Y Combinator*, published in 2018.
https://blog.ycombinator.com/the-decentralized-future-series/

Kevin Roose, “Can Social Media Be Saved?” *The New York Times*, published in 2018.
https://www.nytimes.com/2018/03/28/technology/social-media-privacy.html

Ameer Rosic, “What Are Addresses on Blockchains? What Are Addresses on Blockchains?” *Blockgeeks*, published in 2017.
https://blockgeeks.com/guides/blockchain-address-101/

Ameer Rosic, “What Are Smart Contracts? A Beginner's Guide to Smart Contracts” *Blockgeeks*, no publication date.
https://blockgeeks.com/guides/smart-contracts/

Michael Scott, “Blockchain and the Road Ahead for Digital Property Rights” *BTCMANAGER*, published in 2017. https://btcmanager.com/blockchain-road-ahead-digital-property-rights/

Jessica Silver-Greenberg, Matthew Goldstein and Nicole Perlroth, “JPMorgan Chase Hacking Affects 76 Million Households” *The New York Times*, published in 2014. https://dealbook.nytimes.com/2014/10/02/jpmorgan-discovers-further-cyber-security-issues/

Erik Sherman, “Why $250M didn’t protect JPMorgan from hackers” *CBS News*, published in 2014. https://www.cbsnews.com/news/why-250m-didnt-protect-jp-morgan-from-hackers/

E.B. Solomont, “Special Report: Inside the title insurance cartel” *The Real Deal*, published in 2018. https://therealdeal.com/2016/03/24/inside-the-title-insurance-cartel/

Jimmy Song, “Bitcoin Diamond/Super Bitcoin/BitCore: What You Need To Know” *Medium*, published in 2018. https://medium.com/@jimmysong/bitcoin-diamond-super-bitcoin-bitcore-what-you-need-to-know-f49c35688a39

Taylor Swift, “For Taylor Swift, the Future of Music Is a Love Story” The Wall Street Journal, published in 2014. https://www.wsj.com/articles/for-taylor-swift-the-future-of-music-is-a-love-story-1404763219

Nick Sullivan, “A (relatively easy to understand) primer on elliptical curve cryptography” *Ars Technica*, published in 2013. https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Rob Wile, “Bitcoin Creator Satoshi Nakamoto May Be Sitting on $5.8B” *Time*, published in 2017. http://time.com/money/5002378/bitcoin-creator-nakamoto-billionaire/

"$2 Billion and Counting" *Spotify*, published in 2014.
https://news.spotify.com/us/2014/11/11/2-billion-and-counting/

"Bitcoin blockchain size 2010-2017" *Statista*, no publication date.
https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/

"Chain of Title" *LII / Legal Information Institute*, published in 2014.
https://www.law.cornell.edu/wex/chain_of_title

"Genesis block" *Bitcoin Wiki*.
https://en.bitcoin.it/wiki/Genesis_block

"Home Taping Is Killing Music" *Wikipedia*.
https://en.wikipedia.org/wiki/Home_Taping_Is_Killing_Music

"Second Restatement of Contracts" *JSTOR*, originally published in 1981.
https://www.jstor.org/stable/27876768?seq=1#page_scan_tab_contents

"The recording industry's ability to develop the digital marketplace is undermined by piracy" *IFPI*, no publication date.
http://www.ifpi.org/music-piracy.php

"What is open source?" *Opensource.com*, no publication date.
https://opensource.com/resources/what-open-source