SMART CONTRACTS *in* TRUSTLESS NETWORKS

__Peter Brogden__ considers the functions of blockchain technology, and how the establishment of trustless networks could impact the legal sector.

Introduction

Humans are a trusting species. We trust friends not to lie to us. We trust chefs not to poison us. We trust banks to keep our money safe and secure. Trust is often built on consequences and consistency: the bank has always kept my money safe, and the chef knows I can sue him.

Trust of institutions runs deep within society. We trust that the credit card network has not been compromised. We assume that the bank has its security protocols up to date.

Trust underpins the financial decisions we make. When was the last time you invested directly in an Argentinian winemaker? An Indian start-up? What about the Filipino engineer who needs to patent his idea? These might all be great investments, but most people avoid them because they do not know or trust the participants, and the costs of enforcing on a bad deal are disproportionate. You are no expert in Argentine law, so who knows if you will recover your investment. The people who do make these investments tend to do so through layers of banks, investment companies and consultants – each taking their cut along the way.

Blockchain technology claims that it can change all of this by creating a secure trustless world network running "smart contracts". The claims are bold: whereas Uber made everyone a taxi driver, eBay made everyone an auctioneer, and AirBnb made everyone a hotelier ... blockchain technology lets anyone build a legal system. This paper explores how the blockchain works, and what it means for lawyers.

The Origins of the Blockchain

Understanding the blockchain requires a look back in time, to the heady early days of the internet. In 1999, bored college students discovered that they could compress the music on their CDs and share it (illegally) with others over the internet. By uploading files to a central depository (Napster was the largest), they could send them on to anyone who asked. The weakness is obvious in hindsight: Napster was the single point of failure and soon enough, the lawyers came knocking.

But the lawyers did not come soon enough. At its peak, Napster had over 80 million users, now disgruntled and looking for an alternative. By 2002, Bram Cohen at Buffalo University had invented BitTorrent: a decentralised file-sharing system with no single point of failure. Whilst the early versions still required a central tracker (not to store the files, but to say who had them), it was soon followed by Distributed Hash Table (DHT) technology which sent that look-up information to hundreds of peers across the network, removing any single point of failure from the network. It worked: today BitTorrent has over 250 million users and accounts for about a quarter of all internet traffic.

> *"Whereas Uber made everyone a taxi driver, eBay made everyone an auctioneer, and AirBnb made everyone a hotelier ... blockchain technology lets anyone build a legal system."*

Decentralisation gave people ideas. Napster's servers worked like banks, who store everyone's money and keep track of who owns what. The 2008 crash taught us that banks are not impregnable. What if we could decentralise the money system?

In January 2009, an anonymous software engineer using the pseudonym Satoshi Nakamoto released the first build of a new product called Bitcoin Core. Bitcoin brought together decentralised technology with cryptographic techniques that had already been developed elsewhere.

Cryptography in the Blockchain

Cryptography has been around for thousands of years, but until recently suffered from a singular problem: the sender and the receiver had to agree on a secret key before any messages could be sent. This required a meeting, or at the very least some form of unsecure communication before secure communication could begin. That was a problem.

The solution came in the 1970s. Public-key cryptography uses *pairs* of keys – a public and a private one – to encrypt and decrypt messages. A public key can be widely promulgated. Anyone with your public key can run it through a special one-way algorithm to produce an encrypted message that can only be decrypted with the corresponding private key (which the recipient keeps secret). For the first time, two people could exchange encrypted messages from the start, without intermediaries or non-secure communication.

Around the same time, cryptographers needed a way to check that a message had not been corrupted or tampered with during transmission. Various cryptographic 'hash functions' were invented to map a set of data of arbitrary size (the input) to a string of fixed size (the hash). The best hash functions would radically alter the hash, giving very minor variations in the input.

Given the message and the hash, it was possible to verify whether the message received was exactly the same as the message sent.

The blocks were beginning to fall into place. It was possible to send encrypted messages to anyone along a coordinated but decentralised network.

HOW A HASH FUNCTION WORKS

A hash function maps a set of data of arbitrary size (the key) to a string of fixed size (the hash). Changing the key slightly should comprehensively alter the hash. An example using the MD5 hashing algorithm:

The fox jumps over the dog	MD5 HASH →	01d83900271f3120f48b9848c372a063
The foxes jump over the dogs	MD5 HASH →	d79281ca91a8b35480794caf6294241b

What is the Blockchain?

Bitcoin is one implementation of blockchain technology, and a good example. It works like this:

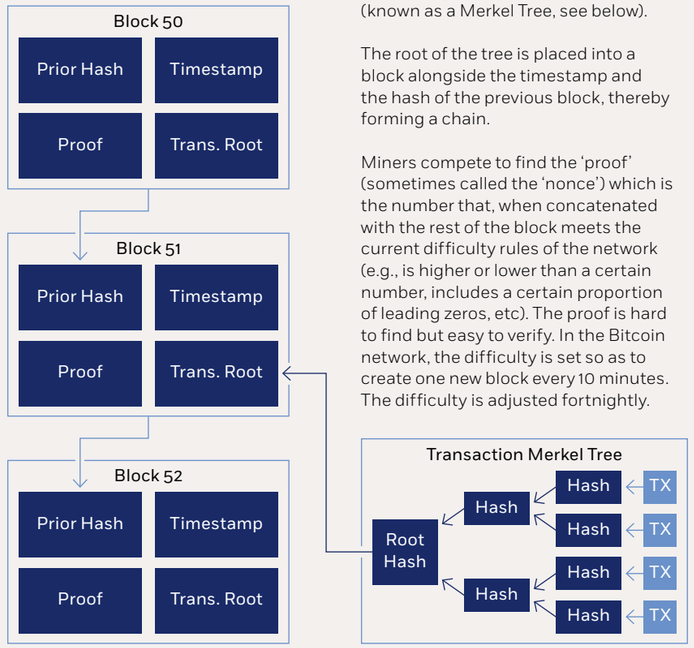
- Anyone can join the Bitcoin network by creating themselves a public/private key pair and connecting to local nodes through a Bitcoin client. Let's assume Alice has done this, and she has 5 Bitcoins. She wants to send one to Bob. Alice broadcasts a message to the decentralised network, with Bob's public key and the amount she wants to send. She hashes the message and signs it with her private key so every knows that it's her making the broadcast. The nodes around her check a global ledger to make sure that she's got enough Bitcoins, and they check her signature is valid. When enough nodes validate the transaction, the ledger is amended to reduce Alice's account to 4 Bitcoins, and increase Bob's account by 1. The ledger is public, and everyone holds a copy.

- Every 10 minutes, certain participants in the network (called 'miners') collect all validated transactions into a block. In order for a block to be accepted into the network, the miner must create proof-of-work. Proof-of-work regulates the supply of Bitcoins, which is essential to preserve value in any money system. Proof-of-work is achieved by solving a complex cryptographic problem, designed to be hard to find but easy to verify. The proof-of-work requires miners to find a number called the 'proof' (or 'nonce'), such that when the block content is hashed along with the proof, the result is numerically smaller than the network's current difficulty target. Every couple of weeks, the difficulty target is automatically adjusted to keep the mining time to about 10 minutes. Miners compete to solve each block, and the winning miner is rewarded with 12.5 Bitcoins (currently about £72,000).

- A solved Bitcoin block contains four things: a timestamp, a hash representing all transactions in that block, the proof found by the miner and – importantly – the hash of the *previous* block, thereby forming a chain.

- The chain is important because it prevents attacks on the network, and attacks must be expected on any money system. If an attacker controlled enough nodes, they could authorise a fraudulent transaction and publish it to the ledger. There would then be two ledgers: one with the fraudulent transaction, and another created by honest users who invalidated the fraudulent transaction because it didn't follow the rules. For the fraudster's ledger to be accepted, he would need to solve the block faster than the honest users, so would need to control more than half of the computing power in the whole network for at least 10 minutes. Even if he managed that, he would need to keep solving blocks faster than anyone else, every 10 minutes, to keep ahead of all the other nodes that contradict his blockchain history. To historically alter the blockchain is even harder – the attacker would need to fork the blockchain by solving each and every cryptographic challenge in the network for as far back as he wanted to go – requiring orders of magnitude more computing power than the rest of the network put together. There comes a point where controlling that much computing power stops being worth the reward.

HOW BLOCKCHAINS WORK



Transactions are transmitted to the network, where they are collated and hashed together into a tree of hashes (known as a Merkel Tree, see below).

The root of the tree is placed into a block alongside the timestamp and the hash of the previous block, thereby forming a chain.

Miners compete to find the 'proof' (sometimes called the 'nonce') which is the number that, when concatenated with the rest of the block meets the current difficulty rules of the network (e.g., is higher or lower than a certain number, includes a certain proportion of leading zeros, etc). The proof is hard to find but easy to verify. In the Bitcoin network, the difficulty is set so as to create one new block every 10 minutes. The difficulty is adjusted fortnightly.

Bitcoin has been wildly successful. It is accepted by many online retailers as it offers zero transaction fees when compared to the 2-3% levied by credit card companies. It is increasingly popular in China and other jurisdictions that strictly control their national currencies. At the time of writing, the value of the world Bitcoin supply is about £100 billion.

Ethereum

Whilst Bitcoin is important, it is the underlying blockchain technology that is the real prize. Just as decentralisation got people thinking about blockchains, so blockchains have people thinking about other kinds of trustless networks. Contracts are the obvious candidate.

The state creates money to support commerce. It imposes two rules on using money, which may be so obvious that we do not even recognise them as rules. They are: (1) you cannot spend more than you have; and (2) you do not still have the money that you have spent.

The Bitcoin network applies these rules programmatically when it validates a transaction.

But what if we extended the platform to execute *any* rules we wanted?

When two or more people write down private rules for their conduct, we call that a contract. We put our trust in contracts because we know that the court system can step in when contracts are broken. But courts can be slow, expensive and occasionally unpredictable. They might work well nationally, but nobody starts an international arbitration over a £50 debt. Smart contracts provide the answer.

Smart contracts are a way to reduce obligations to executable code and have it executed by a cryptographically secure worldwide network. Let us imagine that a bank enters into a smart contract car loan. Whilst the loan is outstanding, the borrower can drive the car but the bank retains the right to stop the borrower selling it whilst the loan is outstanding. If the borrower defaults, the contract rescinds access to the car and grants control back to the bank. If the loan is repaid, the bank's rights to the car are deleted and the borrower assumes complete control.

The ability to reduce contracts to code has existed for decades, but has never gained traction because we have never before had a secure trustless network, outside the control of either contracting party, which we know will execute the contract in accordance with its terms.

In July 2015, a young Russian programmer named Vitalik Buterin designed a blockchain-based system called the Ethereum Virtual Machine (EVM). It has generated an enormous level of excitement in the technology industry, and its currency, 'Ether', is already second in value to Bitcoin worldwide. The EVM is a Turing-complete computer capable of executing scripts on an international network of public nodes. It is similar to Bitcoin but extended to run any kind of contract, effectively as a cryptographically-secure "world computer".

How does it do this? The Bitcoin network and the EVM network both have a ledger that records which Accounts hold currency (Bitcoins on the Bitcoin network; Ether on the EVM). In addition to Accounts, however, the Ethereum network also holds Contracts (as compiled code) and records the machine state of each Contract on the network. Users pay tiny amounts of money (called 'Gas') to have the network run cycles of their contract and move money around the network. The amounts really are tiny, especially when compared to the 2-3% fees charged by credit cards: the average transaction today costs about half a penny, irrespective of value.

The big advantage of Ethereum – and smart contracts – is that they are automatically executable. If you have a stock option that is not honoured, you have to go to court, secure an injunction and call the bailiffs. With Ethereum, that option automatically executes on the network when its conditions are met, moving money between accounts without user input.

Ethereum and Real-Life Law

Does this mean the end of lawyers? In short, no. Law is flexible; it requires interpretation and judgement but is corruptible and sometimes uncertain. Machine code is rigid, inflexible and absolute. There are roles for both solutions.

Forming the junction between life and code does not necessarily require human judgement, but often it will. In the car loan example above, the question of whether a loan has been paid is a binary one, capable of being rationalised by a machine. By contrast, a relatively simple contract to paint a house might require the subjective evaluation of a human where the quality of the workmanship is in dispute.

Smart contracts might deal with subjectivity by incorporating call-out functions. If the painter is not paid, the owner might be required to raise and specify the dispute within a certain time, failing which the painter is paid automatically. Once the dispute is raised, the smart contract code might then define its parameters ('Is this workmanship adequate?') and transmit that question to a third-party arbitrator. The jurisdiction and scope of the dispute is pre-defined, reducing the potential for satellite litigation. A bid/offer system might allow the owner to offer less than the contract value, putting pressure on the painter to accept something less than the price in an effort to avoid the cost of an arbitrator's intervention (similar to CPR Part 36 in England). By making the losing party liable for the arbitrator's costs, we can disincentivise the raising of unmeritorious disputes.

> *"Low-cost, high-trust transacting unlocks the economic potential of new markets and new parts of old ones."*

Ultimately, the subjectivity call-out functions could themselves be contracted out to the network. Let us say an aggrieved party submits evidence of their grievance to 100 human 'judges' across the network, who vote on the outcome. The 'judges' can themselves gain trust and respect from the network by consistently voting in line with (what transpires to be) the consensus, such that vote weight can be adjusted in favour of those who have demonstrated competence and impartiality in the past.

Of course, not everything can be reduced to written evidence and there will always be a role for inspection, cross-examination and advocacy. Smart contracts are, and probably only ever will be, a way to reduce basic and binary disputes to a simpler, cheaper and more certain means of dispute resolution.

What's the Point?

Smart contracts reduce transaction costs and improve trust in those transactions. By implementing what would, in effect, be a global legal system for private law, Ethereum allows individuals to cut out the middleman and invest directly in places they might otherwise ignore.

To return to my opening examples, why wouldn't you invest in that Filipino engineer if you knew (with cryptographic certainty) that you would recover your investment if his patent was rejected? Why not buy equity in that winemaker if you could be sure that you would automatically receive a share of his profit? Low-cost, high-trust transacting unlocks the economic potential of new markets and new parts of old ones.

Decentralisation also has social utility. A decentralised information network cannot easily be censored. A decentralised money system takes control away from governments (see, e.g., Bitcoin's current popularity in China). Right now, a decentralised microblogging platform called Eth-Tweet prevents anyone but the original poster removing their post.

Whilst this talk of cryptography and blockchains might sound very abstract, there are already real-world blockchain applications. New start-up Slock.it creates Wi-Fi connected locks for bikes, lockers and apartments – designed to interface directly with smart contracts. WeiFund is an Ethereum-based crowdfunding platform, which creates individual smart contracts between backers and pitchers. KYC-Chain is positioning itself as a trusted gatekeeper for consensus-based, and KYC regulation compliant, digital identities.

Blockchain technology has yet to reveal its full potential. Right now, it is in its ascendency. The next few years will be an interesting time for lawyers and inventors.