WHAT IS AND WHAT ISN'T A 'BLOCKCHAIN'?

There are a few friends in my network whose opinions I respect and Professor Michael Mainelli, Chairman of Z/Yen Group is one of those. He has kindly given me a really insightful paper into blockchain technologies and terminologies. For the uninitiated, it will enlighten and for the initiated, it will offer something new. Enjoy …

Terminology Wars – I Record Therefore I Ledger; I Block Therefore I Chain?

What is, and what isn't, a 'blockchain'? The Bitcoin cryptocurrency uses a data structure that I have often termed as part of a class of 'mutual distributed ledgers'. Let me set out the terms as I understand them:

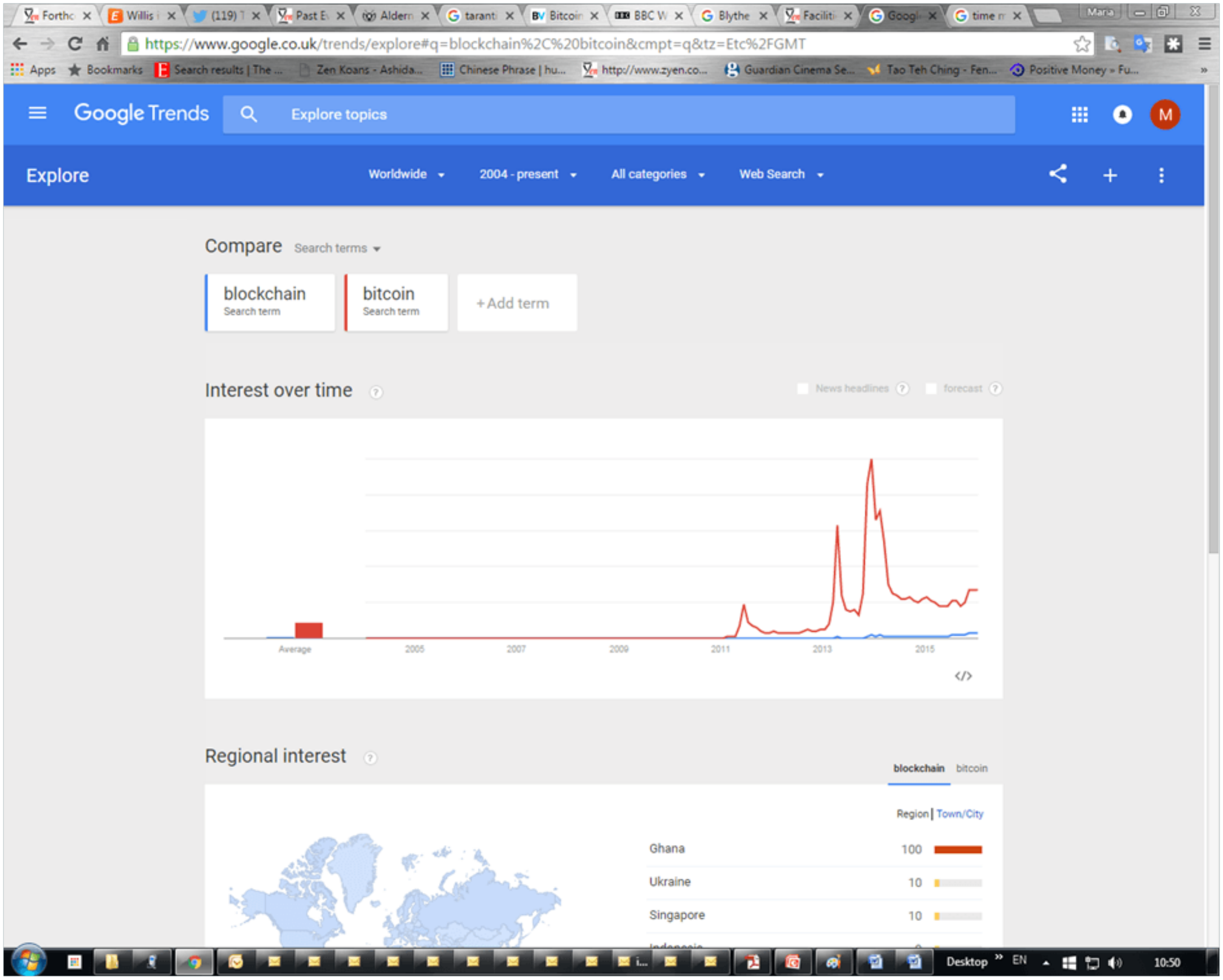
- **ledger** – a record of transactions;
- **distributed** – divided among several or many, in multiple locations;
- **mutual** – shared in common, or owned by a community;
- **mutual distributed ledger (MDL)** – a record of transactions shared in common and stored in multiple locations;
- **mutual distributed ledger technology** – a technology that provides an immutable record of transactions shared in common and stored in multiple locations.

The Bitcoin Wiki defines blockchain as "a transaction database shared by all nodes participating in a system based on the Bitcoin protocol". So, 'blockchain' is, strictly, only the Bitcoin protocol transaction database. A Bitcoin purist might maintain "there is only one blockchain, **the** Bitcoin blockchain!" Yet Ripple (predating Bitcoin and blockchain terminology, from 2004) supposedly has a blockchain, Ethereum (2015) claims to have a blockchain, and about a thousand other cryptocurrencies have a blockchain, though most are copies of the Bitcoin protocol.

Wikipedia is, etymologically, a rather authoritative source on the current shared understanding of words we use to communicate. Wikipedia looks at blockchains more widely – "a block chain or blockchain is a permissionless distributed database based on the bitcoin protocol that maintains a continuously growing list of data records hardened against tampering and revision, even by operators of the data store's nodes. The initial and most widely known application of the block chain technology is the public ledger of transactions for bitcoin which has been the inspiration for similar implementations often known as altchains." Wikipedia's definition permits databases similar to Bitcoin's blockchain to be called 'blockchains'.

Interestingly, the 2008 Satoshi Nakamoto paper that preceded the 1 January 2009 launch of the Bitcoin protocol does not use the term 'blockchain' or 'block chain'. It does refer to 'blocks'. It does refer to 'chains'. It does refer to 'blocks' being 'chained' and also a 'proof-of-work chain'. The paper's conclusion echoes a MDL – "we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power." [Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, bitcoin.org (2008)]

I have been unable to find the person who coined the term 'block chain' or 'blockchain' [contributions welcome!] The term 'blockchain' only makes it into Google Trends in March 2012, over three years from the launch of the Bitcoin protocol.



Worldwide · 2004 - present · All categories · Web Search

blockchain
Search term

bitcoin
Search term

Ghana	100
Ukraine	10
Singapore	10

And the tide may be turning. In July 2015 the States of Jersey issued a consultation document on regulation of virtual currencies and referred to 'distributed ledger technology'. In January 2016, the UK Government Office of Science fixed on 'distributed ledger technology', as does the Financial Conduct Authority and the Bank of England. Etymological evolution is not over.

Ledger Challenge

Wuz we first? Back in 1995, our firm, Z/Yen, faced a technical problem. We were building a highly secure case management system that would be used in the field by case officers on personal computers. Case officers would enter confidential details on the development and progress of their work. We needed to run a large concurrent database over numerous machines. We could not count on case officers 'out on the road' 'dialling in' or using internet connections. Given the highly sensitive nature of the cases, security was paramount and we couldn't even trust the case officers overly much, so a full audit trail was required.

We took advantage of our clients' 'four eyes' policy. Case officers never worked alone. Case officers worked on all cases together with someone else, and not on all cases with the same person. Case officers had to jointly agree on a final version of a case file. We could count on them (mostly) running into sufficient other case officers over a reasonable period of time and use their encounters to transmit data on all cases. So we built a decentralised system where every computer had a copy of everything, but encrypted so case officers could only view their own work, oblivious to the many other records on their machine. When case officers met each other their machines would 'openly' swap their joint files over a cable or floppy disk, but 'confidentially' swap everyone else's encrypted files behind the scenes too. Even back at headquarters four servers treated each other as peers rather than having a 'master' central database. If a case officer failed to 'bump into' enough people, then he or she would be called and asked to dial in or meet someone or drop by headquarters to synchronise. This was, in practice, rarely required.

We called these decentralised chains of data 'stacks'. We encrypted all of the files on the machines, permitting case officers to share keys only for their shared cases. We encrypted a hash of every record within each subsequent record, a process we called 'sleeving'. We wound up with a highly successful system that had a continuous chain of sequentially encrypted records across multiple machines treating each other as peers. We had some problems with synchronising a concurrent database, but they were surmounted. At that time, I also held a senior post in the Ministry of Defence in charge of technology commercialisation, including ITSEC. I commented to colleagues that the most secure system Z/Yen had built had not been for the military. Their immediate reaction was our sleeved stacks were too complex and insecure because the data was not centrally held.

Whether it's Newton or Leibniz, Von Neumann or Turing, Diffie-Hellman-Merkle/Rivest-Shamir-Adleman/Ellis-Cocks-Williamson, simultaneous invention is common. People facing similar problems reach for the contemporary tools available and frequently craft similar solutions. By the time we approached our problem, people had been working on concurrent and distributed databases for about two decades. They might say many of their ideas were in our software. Around the time of our work, there were other attempts to do similar highly secure distributed transaction databases, e.g. Ian Grigg's and Ricardo on payments, Stanford University and LOCKSS and CLOCKSS for academic archiving. Some people might point out that we weren't probably truly peer-to-peer, reserving that accolade for Gnutella in 2000. Whatever. We may have been bright, perhaps even first, but were not alone.

Ledger Choices

I saw my first database in the late 1970s. Yes, there was a time when they were the new new thing. In 1970, ‘Ted’ Codd publicly released some internal IBM work from 1969. I was too young to notice. It was ok. Codd’s work took some time to get recognised. An explosion of interest both within IBM and outside in firms such as Ingres and Oracle only started in the late 1970s. In fact, I saw my first database in Cincom’s Total, rather coincidentally and somewhat confusingly at the French oil firm Total, around this time.

Imagine two scenes.

A – I come back to tell people about the database idea. “Look at this, holds oil data.” “Michael, that’s great but we’re working in cartography. Can it hold mapping data?” “Sure, we just change the name of an oil well to the name of an appropriate river, or the depth of the well to the height of the contour…” Too many people have discovered MDLs via cryptocurrency blockchains, so they think all MDLs work the same way. There are a lot of choices as we’ll see below.

B – I come back to tell people about the database idea, including my boss. “Hey boss, look at this. I have a sample database of everyone in the company’s salaries, and see, here, I can change the amount of my salary and it retotals the entire wage bill automatically without us having to write it out to magnetic tape or punch cards or paper tape.” In my excitement I may overlook the boss’s tone when he or she says, “Michael, perhaps you could make sure that people can’t change that without authorisation? Perhaps you could even add an audit trail that records all transactions?” “Boss, sure I could, but that would add a lot of processing time and storage costs…”

In a strict sense, MDLs are ‘bad’ databases.   They wastefully store information about every single alteration or addition, and never delete. In another sense, MDLs are great databases. In a world of connectivity and cheap storage it can be a good engineering choice to record everything ‘forever’. MDLs make great ‘central databases’, logically central but physically distributed. This means that they eliminate a lot of messaging. Rather than sending you a file to edit, which you edit, send back a copy to me, then send a further copy on to someone else for more processing, all of us can access a central copy with a full audit trail of all changes. The more people involved in the messaging, the more ‘mutual’ participation, the more efficient this approach becomes.

Trillions Of Choices

Perhaps the most significant announcement of 2015 was in January from IBM and Samsung. They announced their intention to work together on mutual distributed ledgers (aka blockchain technology) for the Internet-of-Things. ADEPT (Autonomous Decentralized Peer-to-Peer Telemetry) is a jointly developed system for distributed networks of devices.

In summer 2015 a North American energy insurer raised an interesting problem with us. They were looking at insuring US energy companies about to offer reduced electricity rates to clients who allowed them to turn appliances on-and-off, for example a freezer. Now freezers in America can hold substantial and valuable quantities of foodstuffs, often several thousand dollars. Obviously, the insurer was worried about correctly pricing a policy for the electricity firm in case there was some enormous cyber-attack or network disturbance.

Take for example coming home to find your freezer off and several thousands of dollars worth of defrosted mush in your freezer. You ring your home & contents insurer who notes that you have one of those new-fangled electricity contracts. It was probably the electricity company. Go claim from them. You ring the electricity company. In a fit of customer service, they deny they had anything to do with turning off your machine, but, if anything, it was probably the freezer manufacturer who is at fault. The freezer manufacturer knows for a fact that there is nothing wrong except that you and the electricity company must have installed things improperly. Of course, you may not be all you seem to be. Perhaps you unplugged the freezer to vacuum your house and forgot to reconnect things. Or perhaps you were a bit tight on funds and thought you could turn your frozen food into ‘liquid assets’.

I believe IBM and Samsung foresee, correctly, ten billion people with hundreds of ledgers each, a trillion distributed ledgers. My freezer-electricity-control-ledger, my entertainment system, home security system, heating-and-cooling systems, telephone, autonomous automobile, local area network, etc. In future, machines will make decisions and send buy-and-sell signals to each other that have large financial consequences. Somewhat coyly, we pointed out to our North American insurer that they should perhaps be telling the electricity company which freezers to shut off first, starting with the low value contents ones.

A trillion or so ledgers will not run through a single one. The idea behind cryptocurrencies is ‘permissionless’ participation, any of the billions of people on the planet can participate. Another way of looking at this is that all of the billions of people on the planet are ‘permissioned’ to participate in the Bitcoin protocol for payments. The problem is that they will not be continuous participants. They will dip in and out. And if there is an opportunity for a temporary majority of cheats to defraud the system, they will. But for the other trillion-minus-1 ledgers, we know the participants who need to participate, us, the electricity company, the insurer, our bank, our freezer installer.

Some obvious implementation choices are: public versus private – is reading the ledger open to all or just to defined members of a limited community? Permissioned versus permissionless – are only people with permission allowed to add transactions, or can anyone attempt to add a transaction? True peer-to-peer or merely decentralized – are all nodes equal and performing the same tasks, or do some nodes have more power and additional tasks? People also need to decide if they want to use an existing ledger service (e.g. Bitcoin, Ethereum, Ripple), copy a ledger 'off-the-shelf', or build their own. Building your own is not easy, but it's not impossible. People have enough trouble implementing a single database, so a welter of distributed databases is more complex, sure. However, if my firm can implement a couple of hundred with numerous variations, then it is not impossible for others.

The Coin Is Not The Chain

Another sticking point of terminology is adding new transactions. There are numerous validation mechanisms for authorizing new transactions, e.g. proof-of-work, proof-of-stake, consensus or identity mechanisms. I divide these into 'proof-of-work', i.e. 'mining', and consider all others various forms of 'voting' to agree. Sometimes one person has all the votes. Sometimes a group. Sometimes more complicated voting structures are built to reflect the power and economic environment in which the MDL operates. As Stalin noted, "I consider it completely unimportant who in the party will vote, or how; but what is extraordinarily important is this — who will count the votes, and how."

As the various definitions above show, the blockchain is the data structure, the mechanism for recording transactions, not the mechanism for authorising new transactions. So the taxonomy starts with a MDL or shared ledger; one kind of MDL is a permissionless shared ledger; and one form of permissionless shared ledger is a blockchain.

Last year, Z/Yen created a new timestamping service, MetroGnomo, with the States of Alderney. We used a mutual distributed ledger technology, i.e. a technology that provides an immutable record of transactions shared in common and stored in multiple locations. However, we did not use 'mining' to authorise new transactions. Because the incentive to 'cheat' appears irrelevant here, we used an approach called 'agnostic woven' broadcasting from 'transmitters' to 'receivers'. To paraphrase Douglas Hofstadter, an Eternal Golden Braid. Because we are writing single transactions against a UTC clock, our blocks are UTC time. We don't strictly have blocks of grouped transactions.

So is MetroGnomo based on a blockchain? I say that MetroGnomo uses a MDL, part of a wider family that includes the Bitcoin blockchain along with others that claim use technologies similar to the Bitcoin blockchain. I believe that the mechanism for adding new transactions is novel (probably). For me it is a moot point if we 'block' a group of transactions or write them out singly (blocksize = 1).

Yes, I struggle with 'blockchain'. When people talk to me about 'blockchain' it's as if they're trying to talk about databases yet keep referring to "The Ingres" or "The Oracle". They presume the technological solution, "I think I need an Oracle" (sic), before specifying the generic technology, "I think I need a database". Yet, I also struggle with MDL. It may be strictly correct, but it is long and boring. Blockchain, or even 'Chains' or 'ChainZ', is cuter.

We have tested alternative terms such as "replicated authoritative immutable ledger", "persistent, pervasive, and permanent ledger", and even the louche "consensual ledger". My favourite might be ChainLedgers. Or Distributed ChainLedgers. Or LedgerChains. Who cares about strictly correctness? Let's try to work harder on a common term. All suggestions welcome!

Thanks – I would like to thank Dave Birch of Consult Hyperion and David Bannister of Ovum for kicking off, and kicking around, this thinking.