Contents List

Section 1 - A Ledger for the 21st Century

Section 2 - Blockchain Fundamentals

Section 3 - The Cloud

Section 4 - Smart Contracts

Section 5 - Oracles

Section 6 - The Internet of Things

Section 7 - Cryptocurrency

Section 8 - The Regulatory Lens

Section 9 - Conclusion

Addendum

PREFACE

Blockchain scares some, and infuriates others. But to many, it may well provide the fuel required to drive the next-generation of technological innovations, and in turnmaking the management of assets safer.

I, for one, was somewhere between cynicism and frustration about the hype. With over 30 years of global investment banking and finance technology behind me, I've seen my share of bubbles and emperor's new clothes.

Having said that, there is a problem to be solved - the data challenges that my peers and I faced in financial services seemed so insurmountable that I was prepared to invest some time in order to explore this supposed silver bullet.

And it's not just the financial services world that can be transformed. Finance is central to any business entity, as is the need for trustworthy and transparent record keeping and transaction management. Think ethical data mining, the ability to address GDPR, and sharing of your medical records. Blockchain is so much more than bitcoin and ICOs!

So it seemed that this blockchain revolution was definitely worth investigating and understanding. Not least because it has the potential to fundamentally transform the way that we manage or transfer data and assets.

So, in 2015 I took the plunge, and delved deep into the world of blockchain, distributed ledger technology, and the much berated cryptocurrency landscape. With emotions ranging from incredulity to suspicion at what I had thought was a fairly nascent technology movement, I've come out the other side as a believer in the power of technology to help transform our financial landscape at whatever speed we choose to accept it.

The mission of this book is to enable you the reader to gain a basic understanding of the purpose and value of this emerging technology, itself intended to facilitate trust and transparency in transactions and data. In particular, we want you to have a reasonable chance of forming your own opinion beyond the hysteria of Bitcoin.

We have approached this from a high level foundational standpoint, in order to enable the basic concepts of this technological

revolution to be understood to their fullest, and then embraced. In order to achieve this, we include a few use cases that show how the technology will change our business models, investment profiles, operational paradigms, and general view of trust. (Not quite sure what is meant by 'general view of trust', could be phrased more clearly – CM).

Enjoy the ride!

Section 1

A Ledger for the 21st Century

What's the problem that blockchain and distributed ledger are trying to solve? Let's start with financial data, and build from there.

A ledger records transactions. The process (protocol) for adding a financial transaction to the master ledger, usually by way of a manual or automated journal, is unique to each organisation, and, often, even to each individual department.

The problem in finance - the process and security for adding, amending or viewing the data entry are bespoke to each individual ledger. This is exacerbated further by the geographical and functional fragmentation of organisations, corporate consolidations and multiple technology platforms, among other issues. Finance has got very complicated!

Additionally, the business is *only* recording and managing their *own internal* version of the truth. We don't collaborate with our ecosystem very well - not with our suppliers, vendors, clients, nor counterparts. This leaves us vulnerable to human error, and accidental/deliberate failings in controls, leading to potentially disastrous outcomes - think Enron, UBS and Barings to name but a few. So, although Luca Pacioli's codification of the double-entry system for ledgers holds true since the 15th century, the who-what-when-how-and-why needs significant modernisation in our digital world.

We need a ledger for the 21st century!
We need to ensure Trust, Transparency and Timeliness!

This would be one shared version of the truth, that all parties, internally and externally, can rely on as they have agreed to the process and protocol for updating. Enabling providence and immutability to ledger updates. Sounds like Nirvana to me!

The solution - Blockchain and the Distributed Ledger!

The really exciting part is that not only can this technology solve many of the challenges that we face in the recording and management of financial transactions, but the data being recorded and managed and shared in the ledger can also be anything that matters. The authentication of artwork, validation of source of natural minerals, supply

chain transformation, government and other data record keeping, management and sharing.

The list of use cases is endless, if we combine some key concepts, such as the protocol for adding to the ledger in a safe and secure way through encryption techniques, enabling validation and authenticity to be asserted through consensus, and maintaining a perfect audit trail and providing immutability. This is revolutionary, not just for banks and financial services and finance functions in all firms, but for every sector.

The dream begins with imagining that a protocol for adding data to the ledger is agreed by all stakeholders, and they actually share and work from that same single ledger. Neutral parties are providing the consensus, and ensuring that the encryption is robust. They could actually trust the record, and control who gets to see and change it. They can control the who-how-what-when-and-why.

Dream on, combined with other (emerging) technologies like the Public Cloud, the Internet of Things (IoT), Artificial Intelligence (AI), and the synergystic opportunities enabled by these converging technologies, gives rise to endless possibilities. Imagine being able to track the fish from source to table, and record at all times the storage temperature, in order to ensure its being delivered in healthy condition. In short, the use cases are restricted only by our imaginations.

The power of blockchain and DLT is far beyond mere cryptocurrency use case. The possibilities areexciting and endless ...

…tomorrow's world is here today!

Silver bullet for everyone?

Of course not. There are a number of common sense tests to apply when determining the right technology to solve a business problem. Blockchain and DLT are no different.

For example:
Do you really need collaboration?
Do you really need real-time updates of the dataset?
Does the distribution really need to be widespread; not just hundreds, but thousands, and really tens of thousands?
Are you really disrupting the current business practices?

If the answer to some of the above is "no", then re-engineering current business processes and contracts, combined with developed technology, should suffice. Furthermore, many pure blockchain use cases today achieve consensus, with lots of powerful computers racing to be the first to validate the authenticity of the update. These are anonymous (see later chapters). In the finance world, we don't like anonymity - thus we need administrators, intermediaries, and others that force transparency. And don't forget the old adage - rubbish in, rubbish out. If your data is fragmented today, unfortunately this won't solve that problem. Similarly, if your data taxonomy is broken - this won't solve it.

So, yes, it is powerful, but it's not a miracle!

Remember...distributed ledger technology refers to the technological infrastructure and protocols that allows simultaneous access, validation and record updating in an immutable manner across a network spread that refers tomultiple entities or locations.

Lets have a look at a few use cases alive today...

Marco Polo.

Marco Polo network is a joint venture with technology firm TradeIX and software firm R3 (Corda), along with a number of major banks and their corporate clients.

It was conceived in order to facilitate trade finance between banks, their corporate clients, and the wider trade ecosystem, hopefully in the processsolving the $8 trillion paper-based current operating model.

So lets ask ourselves the blockchain DLT use-case questions?

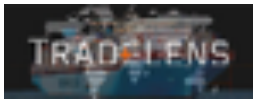
Do you really need collaboration?
Yes. There are many participants in the trade finance ecosystem, all of whom need to use the same data set across geographies.

Do you really need real-time updates of the dataset?
Yes, credit and operational risk is high in this business model.

Does the distribution really need to be widespread; not just hundreds, but thousands, and really tens of thousands?
Yes, the trade finance ecosystem is very large.

Are you really disrupting the current business practices? Absolutely. The old paper, email and need to physically connect between participants was inefficient, insecure, and not as timely as credit providers and payment instructors would like!

Note: MarcoPolo are very transparent that their new solution leverages multiple technologies from Cloud to a specialised DLT, and only then when combined with smart contracts, revamped, and re-engineered business practises, have workflow, logic and rules provided the transformative solution. DLT was merely a component. Collaboration, agreeing protocols, policies, practises and rules was as critical as the technology! And that's the truth and secret of success for any technology transformation!

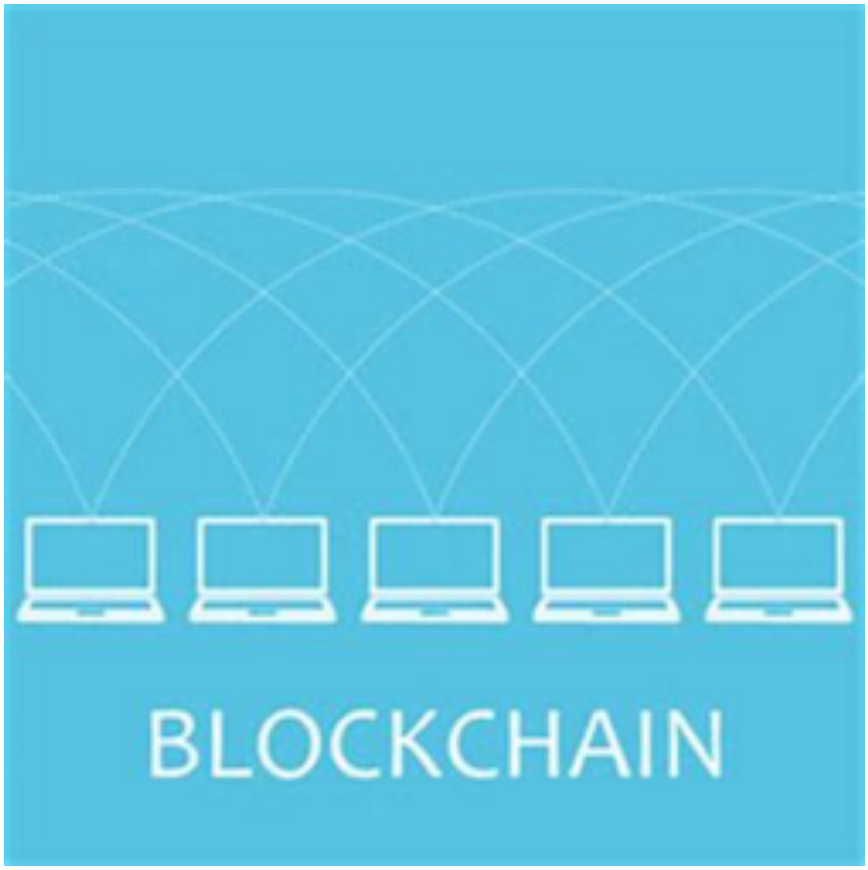


TradeLens

TradeLens is a blockchain-enabled shipping solution. It was jointly developed by IBM and shipping giant Maersk, with the purpose of improving the world's global freight supply, by leveraging blockchain and DLT.

TradeLens uses IBM Blockchain technology as the foundation for digital supply chains, empowering multiple trading partners to collaborate by establishing a single shared view of transactions, without compromising details, privacy or confidentiality.

Shippers, shipping lines, freight forwarders, port and terminal operators, inland transportation, and customs authorities can interact more efficiently through real-time access to shipping data and shipping documents, including IoT and sensor data ranging from temperature control to container weights



Meets the use case tests? Absolutely!

Section 2

Distributed Ledger & Blockchain Fundamentals

Chapters

2.1 Distributed Ledger

2.2 Cryptography

2.3 Cryptographic Hashing

2.4 Public Key Cryptology

2.5 Authentication and Digital Signatures

2.6 Blockchain

2.7 Blockchain structure

2.8 Consensus Mechanism

2.9 Types of Blockchain

This chapter will discuss the technical ideas underpinning blockchain and distributed ledger.

We'll introduce distributed ledgers and what the c*rypto*, or cryptography, in c*ryptocurrency* means, including a few key cryptographic concepts. We'll also cover how a blockchain is structured, and why it is useful.

So, in order to understand a ***distributed*** ledger, we first need to understand what is meant by a centralised ledger. Let's remind ourselves of the purpose of a ledger. Essentially, a ledger is a book or file for recording monetary transactions, with debits and credits recording movements in separate columns. A ledger begins and ends with a monetary value.

Chapter 2.1 Distributed Ledger

What is a distributed ledger, and why is it important for more than Bitcoin?

In this context we'll be talking about a ledger as a store of data, or a **database**. In Bitcoin, it is a store of transactions, but it could also be other types of records.

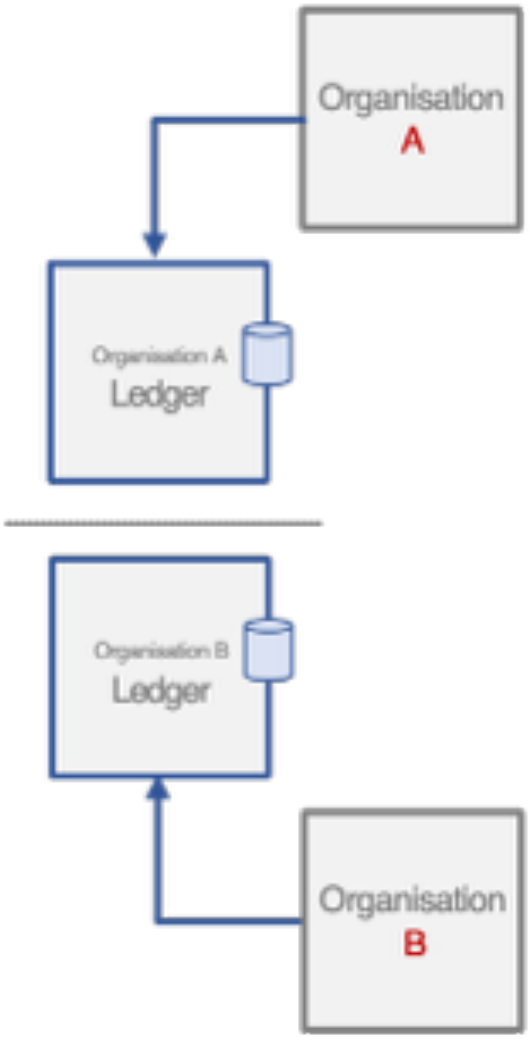
Today, most data stored in an organisation will be contained within relational databases, big data solutions, data lakes, or equivalents. It is all managed within the boundaries of an organisation, and not designed to be shared. If data needs to be shared then it inevitably becomes difficult to do so. You need to solve problems, such as who owns the data, and who can access and update it. Even if youare able to share, sharing data between two parties is very different to sharing data between thousands!

Data held within most organisationsis fragmented because there isn't necessarily a single source of the truth. The data could be split between several databases, or replicated in a way that makes consistency difficult, and multiple reconciliations necessary.

So next we'll look at different ways of sharing this data; namely, separate ledgers with reconciliation, centralised ledgers, and distributed ledgers.

Separate Ledgers

Separate Ledgers are where two parties maintain their own ledgers, each responsible for their own information.



It is the responsibility of the organisations to maintain their own data, so there need to be processes and reconciliations put in place, in order to ensure that the two ledgers are synchronised. This must be managed very

carefully, or else the ledgers could drift apart and become inaccurate. If such a problem is found then it may be unclear which party has the accurate record, and who is responsible for correcting the issue, and in which ledger this should be recorded

And it's not just data that needs to be synchronised. Permissions and validation could be different in different ledgers, meaning one ledger could deem entries to be valid that are not valid elsewhere.

And if more parties become involved then the amount of communication channels increases, meaning that reconciliations become a lot more complicated.

Centralised Ledger

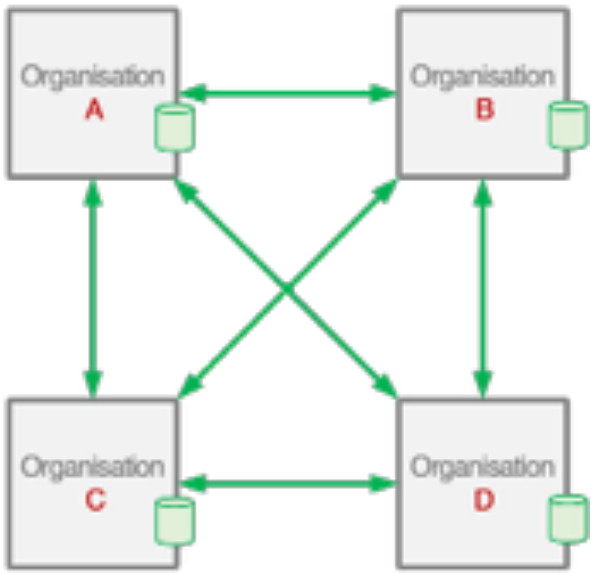
A Centralised Ledger involves the main store of data being held in a single place by a central authority or intermediary who is responsible for managing this data.

A centralised ledger solves the problem of reconciliation because everyone is using the same ledger.

The downside of this approach is that the central authority must manage the interactions, which means that the organisations are placing trust in an external third party. The rules of the interactions must also be clearly defined; for example, permissioning and data validation.

Distributed Ledger

Distributed Ledger allows a ledger to be shared without needing a central authority. This solves the problems prevented by separate ledgers and centralised ledgers.



It is a database that is **decentralised** – ie. distributed among multiple participants and/or locations. It is a solution to the ownership, permissions, sharing, and reconciliation problems that are inherent in a centralised ledger. All participants work together to store, distribute, and validate data.

Bitcoin operates on the bitcoin ledger which is ***decentralised,*** meaning that there is no central authority sitting in the middle managing transactions. It was indeed designed to avoid having a centralised ledger, and a centralised authority.

Chapter 2.2 Cryptography

Two important concepts used by Distributed Ledgers are **cryptography** and **consensus mechanisms**. Cryptography is used to ensure the integrity of all data. Consensus mechanisms are used to ensure all participants in the distributed ledger agree on a single version of the truth. (I can see why you're capitalising Distributed Ledger, etc, in the titles and first sentence. But if you're going to do that then you should really capitalise it all the time, or not at all – CM).

Cryptography is the practice of securing data.

Cryptocurrencies,blockchains, and DLT use three important cryptographic concepts:

Cryptographic **hashing**
Public key encryption
Cryptographic **authentication**, or **digital signatures**

Modern cryptography is a weighty subject, based in mathematical theory. You don’t need to understand the technical details of how it works, but rather what it does and why it is used.

The cryptography used in blockchain is thoroughly proven. The same concepts underpin e-commerce and almost all kinds of digital 'secret' keeping.

Modern cryptography is based on improvements in computing power, and breakthroughs in number theory in the

mid-1970s.

Chapter 2.3 Cryptographic Hashing

Cryptographic hashing is a function, meaning a formula that, when given an input, gives an output called a 'hash value'. It takes any input data of arbitrary length, and returns a short, fixed length value that uniquely represents the input data.

Hashing is used to verify data. Changing the input data even slightly will cause the hash value to change. A hash value will always be short, so it is an efficient way of verifying data.

It stores enough information to uniquely identify the data, but not the content itself. It is akin to a digital fingerprint.

Hash functions are one way, or **asymmetric**. This means that the same hash value can always be encoded from the same input data, but you can never take a hash value and derive what the input data was from this.

A lovely way of explaining this is used by Adrian Patten, the founder of Cobalt; a DLT-inspired Financial services business. He states thatit's like putting a piece of steak into a mincer - whatever you do, you cant reverse the process, and recreate the steak! By the same token, you can't recreate the input data from the cryptographic hash.

Hashing is performed with hash algorithms. If you use the same hash algorithm, whether implemented on a PC, in a browser, smartphone, or tablet...with the same input data, you will always get the same hash value. Examples of hash algorithms are SHA-256 used by Bitcoin, or KECCAK-256 used by Ethereum. Generally, they are actually represented as alpha-numeric characters of 27 to 34 digits in length.

Chapter 2.4 Public Key Cryptography

Public key cryptography is used for encryption and authentication. It is a way of keeping data secret. A **key** in cryptographic terms is the information you need to translate encoded cyphertext to readable plaintext.

Public Key Cryptography was invented to solve the problem of key distribution. Symmetric encryption means the same key is used to encrypt and decrypt. If you are going to pass a secret to someone, you must both have the key, and both keep it secret. This is a logistical problem because you must manage separate keys with everyone that you communicate with.

Public key cryptography is asymmetric. It uses two different (but linked) keys. There is a public key for encryption, so anyone can encrypt a message, and a private key for decryption, so only the private key holder can decrypt the message. This solves the key distribution problem. The private key holder can distribute the same public key to everyone, safe in the knowledge that they can send encrypted messages, but they are never able to read other messages to the private key holder.

A message is encrypted with the public key, resulting in an encrypted message

The message can be decrypted using the private key to get back to the unencrypted message

With Bitcoin, your private keys live in your wallet (see chapter 7). This is the information only you hold, which ensures that yourbitcoins are safe.

The public key is akin to an open padlock. After encryption, the padlock is closed, and a message locked inside. Only the private key can unlock it.

Public key cryptography is based on mathematical problems that are easy to solve in one way, but incredibly difficult to the point of being virtually impossible to solve the other way around. They are described as *trapdoor functions*. You can go one way through the trapdoor, but it springs shut, and without a key to unlock it you can't get back through.

There are several mathematical 'problems' that are used. One is based on *prime factorisation,* where two large prime numbers are multiplied together to get an even bigger semi-prime number. The two individual prime numbers (prime factors) are the private key, and the product of them is the public key. It is virtually impossible to know how the product was made without knowing one of the prime factors.

Bitcoin uses a different technique called Elliptical Curve Cryptography, based on something called the *elliptic curve discrete logarithm problem*. That's all you need to know for now! Too much more information would be too much for this book!

Chapter 2.5 Authentication and Digital Signatures

Digital signatures are used to validate data to prove its authenticity and integrity. They definitively tell you who authored the data in question, and confirm that it has not been changed since the signature was created. They are effectively a stamp of authenticity, a wax seal proving the provenance of the information. You can't change the content of the message, without damaging the seal.

A digital signature can be used to authenticate an email, so you can be certain that the email from your bank really did originate from your bank, or to authenticate a PDF invoice so you know that the account number is correct, and that a fraudster hasn't sent you an amended version with their own account details.

In Bitcoin, digital signatures are used to prove that you are the instigator of a transaction, and also to authenticate the details, the amount, and the recipient of that transaction.

Digital signatures are based on hashing and public key cryptography. Public keys can work in two ways. They can be used for one-way encryption, as we have seen, but they can also be used the other way around for one-way decryption. This effectively means that anyone can decrypt, but only the private key holder can encrypt.

Digital signatures use private keys for encryption, and public keys for decryption. The private key holder is the only person who can create a suitable digital signature. They prove the authenticity and integrity of the data. Public key holders can confirm and validate by using the public key, and decrypt the signature safe in the knowledge that the only person who could produce a correct digital signature is the private key holder.

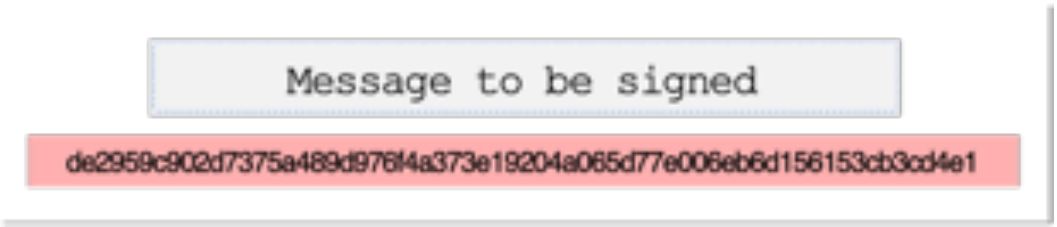
The process is as follows:

Signing

Step 1 - Data that is going to be signed is hashed

Step 2 – The hash is encrypted with the private key

The message is now signed with a digital signature



Validating

Step 1 – The signature is decrypted with the sender's public key giving the hash

Step 2- The decrypted hash is compared to our own hash of the data

Step 3 – If the hashes match we know the message is valid and the sender is authentic as they are the only key holder able to produce a correct hash. If the message were changed the hash comparison would flag this up.