BSV BLOCKCHAIN

WHAT IS BLOCKCHAIN?

Chapter 1: The evolution of data software and systems

Chapter 1:

The evolution of data software and systems

To understand the type of data system the blockchain is, let's take a look at the kinds of data software and systems we deal with in everyday scenarios.

A spreadsheet on your computer

When you create a new spreadsheet, you're essentially constructing an electronic version of information so you can store it on your computer's internal hard drive as data. The spreadsheet software, as well as your internal hard drive (the hardware), are designed for a single person to use at a time.

Pros:

- As long as your computer remains secure, you'll be the only one who has access to your data.
- The spreadsheet software allows you to use formulas to automatically calculate values based on data in another data cell or spreadsheet stored on your hard drive. If your spreadsheet software is compatible with your database tool, you'll be able to automate data interactions between the two.

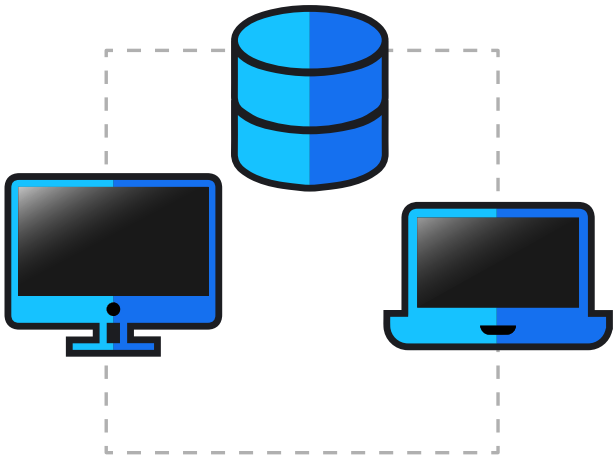
Cons:

- If your computer is stolen or your hard drive is corrupted, all of your data will be gone.
- Your computer could get hacked, giving unauthorised parties access to your data.
- If you wanted someone else to work collaboratively on the same spreadsheet, you'd have to send them the file. They would then have to return the latest version of the file. The file will pass to and fro, resulting in many different versions of the same spreadsheet.

A spreadsheet in your company's data management system

When you create a spreadsheet on the computer system at work, the electronic data will be stored on powerful computers (the server) located downstairs in the IT department. Most likely, your company will have backup servers keeping a duplicate of the entire database - either on or off-premise.

Since the company's servers store all the data generated by each person and department, the entire database will be accessible to people with the correct permissions - if they know where to look.



Pros:

- Anyone who has access to the server can be granted permission to read and/or write the same spreadsheet to improve collaboration.
- Depending on the sophistication of your company's software (ERP system), you'll be able to automate data interactions between a number of your software tools. E.g. Should the projected profit margins calculated by one of your programmes fall below a certain level, your email software would automatically send a notification to a relevant individual.

Cons:

- Everyone with access to the data can write, rewrite or erase data, with no record kept of the changes. Should an error slip in or a willful manipulation of data occurs, it will be very difficult to detect and trace it back to its source to correct and hold accountable the person at fault.
- When audit time comes, you'll need a team of auditors to sift through data stored in various locations on your server to combine it into a single, coherent report. Because they have to rely on the latest version of the file with no record of the changes (legitimate or illegitimate) that have occurred along the way, it will be impossible to guarantee the accuracy of the data they report on. Failing an audit could result in fines and/or other penalties for the company.
- Other departments or regional offices might use different software systems that's incompatible with yours. This creates data silos that limit collaboration and prevent anyone from gaining a Big Picture overview of operations..

- The company is responsible for the entire cost of ownership of the system, including computers, servers, software, security and the cost of employing IT personnel.
- Company servers can go down and cause operational downtime..
- Company servers are a prime target for cyber-attacks. If your server is hacked, sensitive data like client's personal information and corporate secrets might be compromised. A ransomware attack could encrypt your data and hold you ransom to get access to the decryption key.
- If a cyber-attack occurs, it will be essential to detect it early to isolate the affected machines. But, because your database allows people to read, write and erase data, hackers can erase traces of their access. This will let them lurk in your servers until they can do maximal damage to your systems. The SolarWinds security breach is a perfect example.

A remote working model

In a remote working setup, you would create your spreadsheets via Cloud-based software that stores data on a Cloud server, like Google Drive.

In a cloud computing model, the servers that run your software and store your data are made up of powerful computers located in special remote warehouses (a data centre). Hundreds or even thousands of computers combine their power and storage so many clients (i.e. different companies) and users can access the database simultaneously. To access your data from this data centre, you could use an Internet connection or a more secure direct line.

Pros:

- If your server is hooked up to the Internet, you can collaborate with anyone, wherever they find themselves in the world.
- Depending on your software's sophistication (Cloud-based ERP system), you'll be able to automate data interactions between your software tools.

Cons:

- Everyone with access to the data can write, rewrite or erase data, with no record kept of the changes. Should an error slip in or a willful manipulation of data occurs, it will be very difficult to detect and trace it back to its source.
- Company auditors have to assume that the data in your system is accurate, even though changes could have occurred between the time the data was logged and the present.
- If your company needs to collaborate with other organisations, the other parties will have to use the same software. Alternatively, you'd need a software integrations specialist to build a bridge to allow different programmes to communicate with each other.
- Although the cloud provider bears the costs of the outlay, maintenance and security of their infrastructure, you will have to pay a subscription fee to use it. There are different costs associated with different types of data. For example, a customer would pay ingress and egress fees for accessing their archival data. If it's a free product, the provider is probably selling your data to fund their business model.
- When your provider's servers go down, your data will become inaccessible.
- Cloud servers can get hacked or mistakenly share your data with unauthorised individuals.

Blockchain: a data integrity layer for your data network

(Bitcoin was the first implementation of the blockchain concept that successfully solved the double-spend problem to create digital cash. And so we consider the Bitcoin blockchain as the archetype for a blockchain system. Read chapter four for more info.)

Scenario 1: storing data on-chain

If your company operates on a public blockchain system, **you have the option to store a copy of your database on a computer network (the server) distributed across the globe.**

Going about this way is necessary and useful - mostly for smart contracts.

The blockchain database is **publicly accessible**. If you do decide to store your data on-chain encrypting the data ought to be considered as anyone who accesses the database (via a blockchain explorer over the Internet) will be able to sift through the data.

Though they won't necessarily be able to make sense of how the data relates to other data or who created or made changes to any particular data record, certain actions might betray business entities. Privacy can be achieved with proper consideration and a wallet configuration where public keys aren't reused which firewalls identity from pseudonymous transactional activity. This means immutable records can be maintained on the blockchain for enhancing data integrity and auditability in your business.

Scenario 2: storing a hash of your data on-chain

More likely, you would continue storing your data on existing servers. What you would store on the blockchain (on-chain) is 'a digital fingerprint' of your data, in blockchain terms, a hash.

Each time someone makes an update or change to the data records you've elected to store on-chain, **a new fingerprint that references the previous record will be created, with the latest hash published on the blockchain.**

Much like a version control system, the **chain of hashes will** show the entire history of changes to your data, allowing you to validate the authenticity of data and identify the moment an unauthorised data breach had taken place to allow you to restore your system to the latest accurate version.

How the BSV blockchain approaches integrity?

The BSV blockchain utilises a SHA-256 algorithm that converts a string of text into a deterministic output called a hash. The output cannot reveal the input, but anyone with the input can easily verify its authenticity if the corresponding hash is public. When you have a secure hash, even the slightest change in the input will be met with a completely different output - for example, changing a letter from upper to lower case or simply adding a period.

INPUT	HASH
This is my secret	48a6d0cd5fe60d4d615e1c499bc6a90e3342742e5326e484edb4412214d496d6
This is my secret.	e910905c6e2be03cd5e1e2b4a3f5e57ac9331ff146c92bb7a84d89fb0f05334c
This is my Secret.	1151545fc7083937fac94e26abbdea7536811f9ea549f95be06207b339607d6a

This feature creates a ‘digital fingerprint’ and is used by transaction processors (aka nodes) for securing the integrity of the blockchain. The blockchain is simply a sequence of blocks (or a linked list) with each block containing the previous block ID (which is itself a hash of its header data) in the next block’s header data. The block header also contains a value known as a Merkle root which is a hash of all transaction data present in that block and works to securely summarise the relative order of one transaction to another.

This powerful format provides a single chain of valid blocks leading all the way back to the very first ‘Genesis’ block. As these blocks are built upon each other through proof-of-work, any attempt to overwrite or change this stored data would require an attacker to construct a new proof-of-work chain while outpacing honest nodes, making it computationally impractical. Thus, any information contained in blocks with an established quantity of proof-of-work on top of them is extremely resistant to change.

Any node or user can quickly validate and authenticate the integrity of the data inside a transaction by using just the Merkle proof of a transaction to ensure the hash is sound.

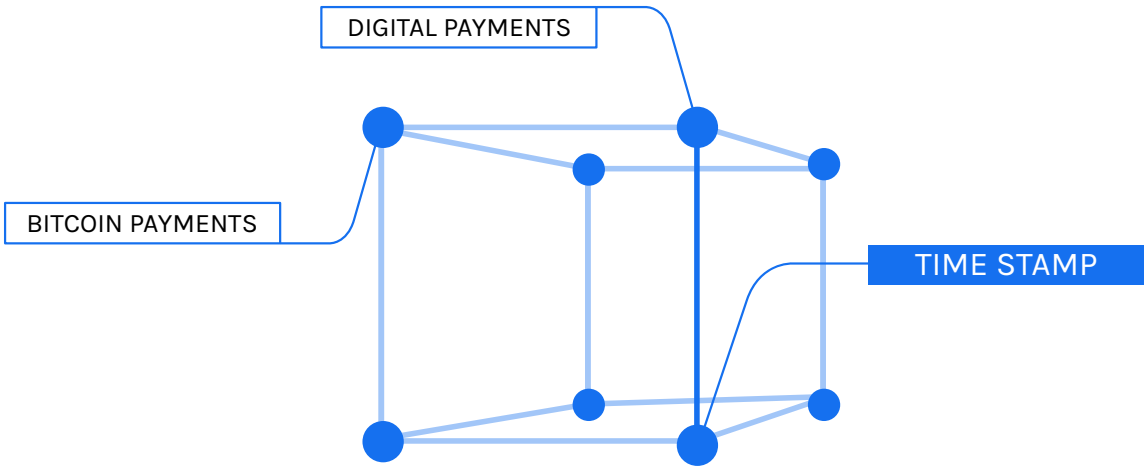
A merkle proof is extremely efficient as it only involves relaying the binary logarithm of the number of transactions in a block worth of 32 byte values. For example, a transaction in a block with 4.3 billion others can be verified simply with the provision of just 32 × 32 byte values.

Chapter 2: Why call it a ‘blockchain’?

Chapter 2:

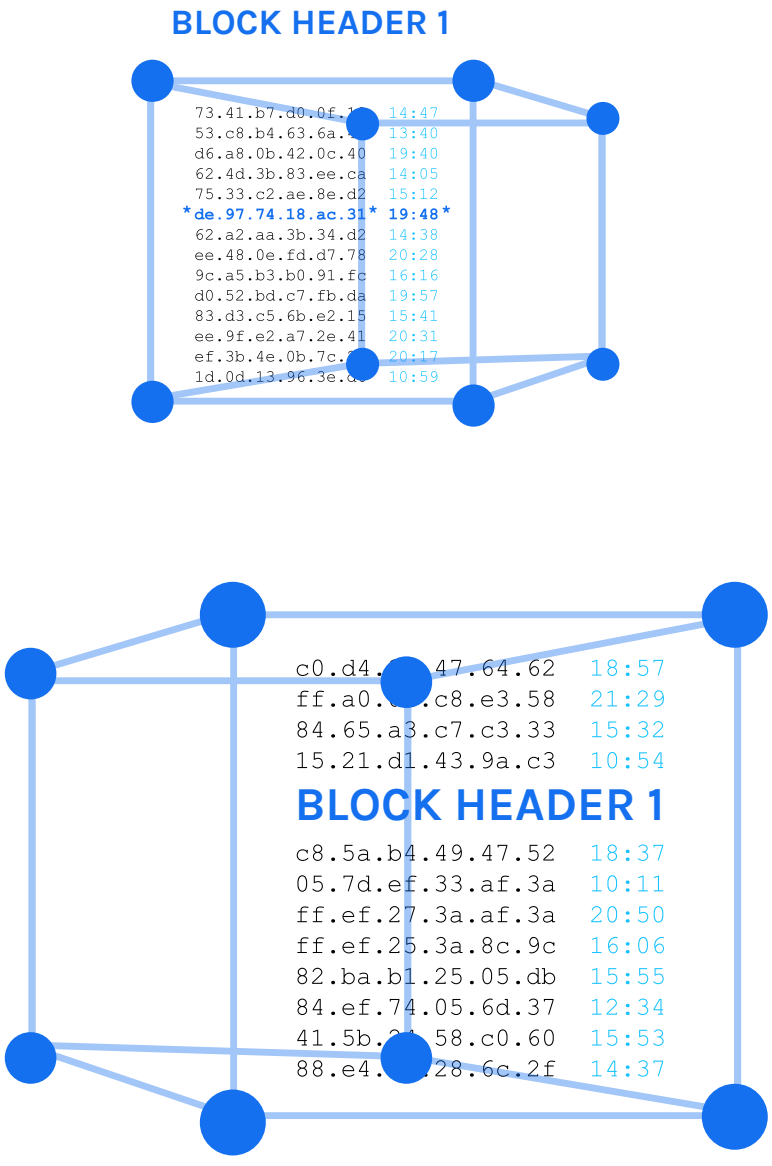
Why call it a ‘blockchain’?

The Bitcoin blockchain’s software indicates precisely when it published each record by adding a timestamp upon publication.



Doing so creates something a bit like a bank statement - or ledger - with each event listed along with its time.

Long lists of events and times are grouped into blocks. Each block has a block header - an identifier that is unique to that block and mathematically connected to the block header of the previous block. This system of ordering blocks through a mathematical connection between their headers to link (chain) the blocks in a fixed sequence is known as the blockchain.



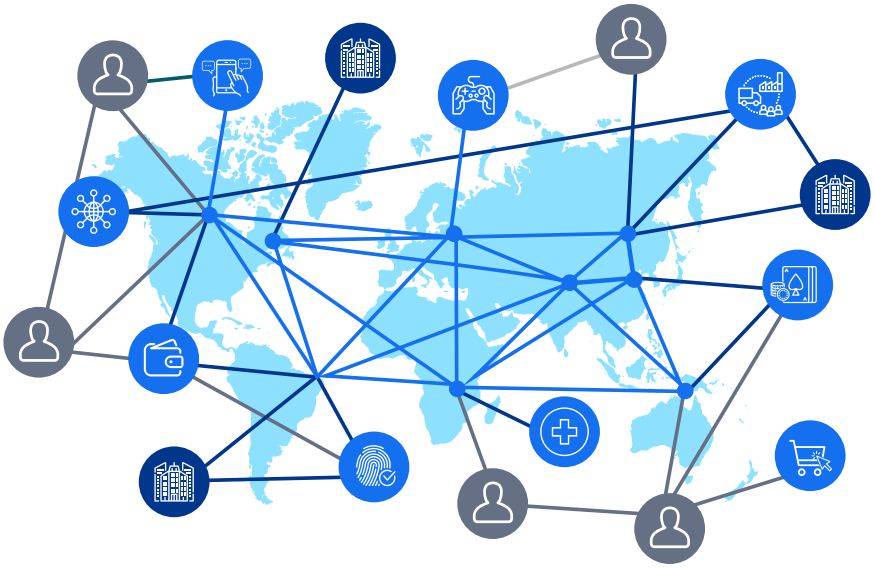
Chapter 3: Blockchain architecture

Chapter 3:

Blockchain architecture

Like all data management systems, a blockchain system is composed of hardware, software and a protocol:

- The blockchain hardware is made up of specialised computers that are used to enforce the network consensus rules.
- The blockchain protocol establishes the rule-set that governs the network and its participants. According to the Bitcoin white paper, this is based on proof-of-work, where one CPU equals one vote to incentivise honest activity.
- The blockchain software provides the bridge between the transaction processors and protocol rule-set that delivers the work to the rest of the network.



Chapter 4: How do Bitcoin and blockchain fit together?

Chapter 4:

How do Bitcoin and blockchain fit together?

The blockchain model is derived from the invention of Bitcoin. We can consider the Bitcoin blockchain **the archetype or model for blockchain systems.** The characteristics of this model are **described in the Bitcoin white paper.**

This blockchain model can only achieve the **greatest efficiency by operating as the sole data management system to serve billions of applications globally.** In such a scenario, interoperability between all parts of business and society is maximal, while the infrastructure costs are kept to a minimum as it replaces all other data systems.

Though there are blockchain projects that have established different versions of the model, **they will not deliver the same efficiencies.** as 'blockchain' is commonly used as a buzzword, it's worth taking any project with that label with a grain of salt.

A TRUE "BLOCKCHAIN" DELIVERS A KEY SET OF CAPABILITIES

Public	Public, global distribution of data
Transparency & auditability	Timestamp ledger, with business events (payments and data transactions) recorded in chronological order
Immutability	Tamper-proof record keeping, so data is immutable
Fast & low Cost	Fast, efficient and low-cost payment system
Privacy with traceability	Transactions have privacy but are traceable (not anonymous)

BUT NOT ALL "BLOCKCHAINS" ARE ACTUALLY BLOCKCHAINS

"BLOCKCHAIN" HAS OFTEN BEEN USED TO INCORRECTLY DESCRIBE DISTRIBUTED LEDGER TECHNOLOGIES (DLTS), FEDERATED DATABASES, AND EVEN DATABASES LOCATED IN MULTIPLE SERVER LOCATIONS

✗ Some supposed "blockchains" are just private, permissioned ledgers, which do not guarantee immutability.	✓ A true blockchain should be kept in distributed fashion and publicly verifiable, which helps (along with proof-of-work) ensure their immutability.
✗ Certain technologies, like DLT, do not have cryptographically linked blocks of time-ordered transactions.	✓ A true blockchain should record a time-ordered record of business events (payment or data transactions), and secure them via cryptographically linked blocks of such transactions.