

# Blockchain Byte

R3 Research  
Emily Rutland



The Blockchain Byte features a question from the distributed ledger space

r3.



# What is the distinction between a blockchain and a distributed ledger?

Blockchain has a shared and replicated ledger comprised of information stored in “blocks” and sits below a distributed ledger and acts as a way to verify transactions submitted by producing a new “block” to the chain.

Distributed ledger is a record of consensus with cryptographic audit trail maintained and validated by nodes. It can be decentralized or centralized.

While often used interchangeably, a blockchain and a distributed ledger are distinct – though subtly different – technologies.

A distributed ledger is a record of consensus with a cryptographic audit trail which is maintained and validated by several separate nodes\*. This cryptographically assured and synchronized data can be spread across multiple institutions. Distributed ledgers can be either decentralized, granting equal rights within the protocol to all participants or centralized, designating certain users particular rights. Actors typically employ distributed ledgers when they need a tool which permits the concurrent editing of a shared state while maintaining its unicity. The ledger's state is determined through a consensus algorithm, which can vary in its mechanics but ultimately serves to validate information from inputs to the network.



Often described as the technology that underpins the Bitcoin network, a blockchain data structure has a shared, replicated ledger comprised of digitally recorded and unchangeable data in packages called *blocks*. A blockchain sits below a distributed ledger and acts as a way to order and validate the transactions in the ledger. While a blockchain automatically produces a new “block” after certain predetermined criteria is met, a distributed ledger only verifies a transaction once it is submitted, as opposed to pushing out empty blocks. **A blockchain is a way to implement a distributed ledger, but not all distributed ledgers necessarily employ blockchains.**

\*In a distributed ledger, it is not necessarily the case that all nodes either receive all the information or, if they do, can understand it. In the Ethereum model, for example, all nodes receive and understand all the information. In Corda, only the nodes involved are aware that a transaction exists.





# What are Blocks and Why Aren't They Necessary?

Proof-of-Work groups transactions into blocks and broadcasts it to unrelated parties.

Therefore, blocks are not suitable for use in a trusted distributed ledger network between financial institutions.

A block structure allows for global broadcast of data and large amounts of unrelated transactions to be confirmed at once. While blocks make sense for public blockchain systems like Bitcoin, this logic does not apply to trusted distributed ledger networks used within financial institutions, like Corda.

A “block” is a bundle of transaction data\*. Most closely associated with Bitcoin, a blockchain structure connects blocks such that each block includes the hash value of the previous block, thereby forming a chain of valid transactions.

“Miners” create new blocks by validating transactions through a Proof of Work (PoW) exercise. This PoW exercise allows anyone (without permission) to create valid blocks if they can expend the required computing power (calculating hashes). This resource-intensive process also creates a financial barrier to deter malicious actors from adding fraudulent blocks.



This PoW system deliberately takes time both to prevent arbitrary coin generation (which would lead to hyperinflation of the currency) and to prevent too many solutions at a given time (once a miner finds a solution, it must propagate around the network, which takes time and may lead to collisions and forks). In Bitcoin, this interval is an average of 10 minutes. To accommodate for this delay, transactions must be batched together and confirmed in blocks.

In trusted, private distributed ledger networks, PoW and global broadcast of transactions are not necessary - it does not make sense to group unrelated transactions into blocks that are then shared with many, unrelated parties. Those private blockchain systems that do use blocks most likely use the Bitcoin structure without considering why Bitcoin works the way it does.

Corda is different. Recognizing that the block structure is a flawed design decision for financial services, R3 deliberately chose not to use blocks for Corda and its financial services clients





# What is the difference between a centralized and decentralized blockchain?

In a decentralized network, anyone can transact on the ledger. Bitcoin's network uses mining and proof-of-work to maintain the integrity of the ledger.

In a centralized network, only known and identified parties can transact on the ledger. Therefore, their transactions can be audited.

A blockchain can be either centralized or decentralized. It is important, however, that decentralized not be confused with distributed. While a blockchain is inherently distributed (meaning that many parties hold copies of the ledger), it is not inherently decentralized.

Whether a blockchain is centralized or decentralized simply refers to the rights of participants on the ledger, and is therefore a question of design.

In a decentralized network, anyone can participate and transact on the ledger. As a result, mechanisms must exist in order to combat the vulnerabilities that arise from this design and to ensure that transactions are correct. Bitcoin, for example, is a decentralized blockchain that uses mining and proof-of-work\* to maintain the integrity of the ledger and to prevent people from corrupting the system.

A centralized network, on the other hand, is made up of parties whose identities are known. Thus, the system is valid because only credible and reputable participants can post to the ledger. Because participants' identities are known, their transactions can therefore be audited.



Ultimately, a centralized distributed ledger should be used in any highly regulated industry - such as financial services - to minimize vulnerability. While both decentralized and centralized blockchains have respective risks, a centralized network is much preferred for our purposes in that the identity of participants is known and, as a result, an audit trail exists should an actor attempt to corrupt the system. It is preferable, for example, to fine a known entity that entered an invalid transaction rather than attempt to account for every possible manipulation that could potentially occur in an anonymous, decentralized network.

\*Proof-of-work: a function to deter denial of service attacks and other potential spam on a network by requiring some level of work that is usually costly and time-consuming. In the Bitcoin network, miners must complete a proof-of-work in order to regulate the rate at which new blocks are generated.





# What is Central Bank Digital Currency?

Central Bank Digital Currency is a digitally stored currency that theoretically can achieve real time settlement for individuals and banks.

CBDCs are intended to be issued by the central bank and able to exchange for fiat currency. It can also be an alternative to current market currency.

The concept of digital currency precedes distributed ledger technology. It began as a way to distinguish between Internet versions of digital money (i.e. DigiCash, eCash) and electronic cash that existed on physical platforms (i.e. Chipknip and Mondex) and grew more complex with the advent of web platforms that provided access to accounting (i.e. Paypal). These examples are all private market currency solutions that do not involve a central bank.

Though the concept of a Central Bank Digital Currency (CBDC) remains largely theoretical, the evolution of new technologies such as DLT is increasing the feasibility of putting a CBDC into practice. At a high level, a CBDC is a digital store of value (money) and method of exchange issued by a central bank. Theoretically, a CBDC introduces a new digital mechanism for real-time settlement between individuals.



CBDCs are intended to be 1:1 exchangeable with other forms of money (such as notes, coins and deposits at banks). They may be issued in an alternate form exchangeable for fiat currency that is held on deposit by a central bank and payable on demand to the owner. CBDCs can also be issued as a new form of money supply in addition to traditional central bank money issuance.

One of the main purposes of a CBDC is to broaden access to central bank liabilities (such as notes and coins) in digital form, thus increasing the ease and convenience required to hold and pay these liabilities. In addition to broadening this access, a CBDC system must also be designed to be practically functional (for example, it cannot only be accessible through proprietary networks such as SWIFT or Fedwire).





# What is a Digital Signature?

Digital signature authenticates the integrity of data by using asymmetric cryptography.

Digital signature is nearly impossible to counterfeit without access to the sender's private key.

A digital signature is a method to verify the authenticity and integrity of a digital message.

Digital signatures employ asymmetric (public key) cryptography, a cryptographic system that uses a pair of keys – a public key, which may be shared widely and a private key, which is known only to the owner – to encrypt and decrypt messages.

A digital signature ensures the authentication and integrity of data. Authentication of data refers to the verification of the source of the data, meaning that the message was indeed sent by the person or entity who claims to be the sender. This is ensured because the digital signature uses a private key that is known only to one specific user.

The integrity of data refers to the content of the message itself, meaning that the message was not altered in transit. A digital signature ensures this because, if a message is digitally signed, any change in the message after it has been signed invalidates the signature.

A digital signature also provides non-repudiation, meaning that the signer cannot later claim that he/she did not sign the message.



A digital signature is created by encrypting the hash of a document using the private key. A recipient can use the public key to decrypt the signature and ensure that the result indeed matches the hash of the document. If a document changes after signing, the digital signature is invalidated because the document's hash will not match the decrypted signature.

Unlike a handwritten signature, a digital signature is nearly impossible to counterfeit. A fraudulent party cannot fake a valid signature without access to the secret, private key, held only by the sender. Furthermore, unlike a handwritten signature, a digital signature not only verifies the identity of the signer but also validates the contents of the message itself.





# What is a Golden Copy?

A golden copy is a master version of a record of data and referred to as the the "truth" that users can be sure that it is true.

In order to make changes to the "truth", changes must be validated by participants that has the power to make and validate.

A golden copy – or golden record – typically serves as the official, master version of a record of data. It is an authoritative single source, sometimes referred to as the "single version of the truth", where "truth" is understood as that which users can ensure is the single correct piece of information which can then be actioned upon.

What constitutes a golden copy varies based on the particular use of that data. For example, a passport or social security number may constitute a golden copy in the case of customer KYC information, while a driver's license can be thought of as golden copy of a form of identity at the DMV.

For financial institutions, there is a reliance on a single source of data that is untainted, "true" and represents something that is actionable. In trading, for example, this means that the initial booking of a trade and agreement of the attributes of that trade between counter parties forms a representation that can be used for some action.



In other words, the initial construct of a golden copy consists of the data attributes (of a trade contract information that is both common like reference data and counter party specific, like a trader ID) that define the data and confirmation/validation that makes this golden copy reliable and, ultimately, actionable.

While existing processes - like reconciliation - currently carry out this validation process in a bank, distributed ledger technology (DLT) may ultimately be able to carry out these actions far more efficiently. DLT enhances the notion of a golden copy in that it enables a group of participants to confirm that data is true, lowering the cost of reconciliation and relieving the burden of having a single institution validate or act as a golden copy. DLT allows for multiple, distributed confirmations, strengthening the claim of factuality.

A golden copy is considered the original, incepting transaction – any amendments or changes to the data must be confirmed and validated by participants in the distributed ledger network that have the authority to make and validate those changes. In this sense, a golden copy is a living piece of data that is continuously validated.





# What is Immutability?

Immutability implies that data cannot be edited or deleted by participants or system administrator after it is written to the ledger.

Immutability is a frequently used buzzword when discussing blockchain and distributed ledger technology, and the perception of a blockchain as a permanent record of information is one of the greatest benefits of the technology.

The term “immutability” refers to a state that cannot be changed or modified after it has been created. (On the other hand, a ‘mutable’ object can be changed after it is created.) As it pertains to blockchain technology, immutability implies that data written to a blockchain cannot be edited, even by a system administrator. This is beneficial, especially in financial transactions, in that one can be sure that sent and received data and transactions cannot be altered.

For Bitcoin, immutability of transactional information is achieved through the process of creating a hash itself, which is one-way. Additionally, the process of hashing bundles transaction outputs into blocks which contain the hash of the previous block, thus ensuring consistency of data and that previous blocks have not been altered without compromising the entire blockchain and revealing that this change has occurred.



While immutability is often discussed as an absolute, innate feature of a blockchain – and one of its greatest strengths – many claims of immutability are over-exaggerated and not entirely accurate. Immutability is not an absolute quality, but a relative quality. "Immutability" does not necessarily mean that data cannot be changed; rather, it is a measure of the difficulty required to change data on a blockchain. Additionally, any alteration can often be detected fairly easily, ensuring the blockchain's integrity.

The difficulty to change a data entry on a private blockchain depends on the level of buy-in needed from participants to sign a new replacement block (for example, the number of digital signatures necessary to sign this replacement). The more buy-in necessary to make a change, the greater the "immutability" of the blockchain.

Even if changing original blockchain data is possible with necessary effort, it is preferable to instead produce a new line with this edited information that refers back to the original block. As it pertains to the financial services industry, regulators prefer this design principle as it ensures an audit trail for reference.





# What is an Oracle?

Oracles bring in trustworthy external data to a distributed ledger, which is necessary if a smart contract needs to interact with external data.

Information are digitally signed in the transaction and are non-repudiable.

An oracle is an external source of data to a distributed ledger which is considered to be authoritative, trusted and definitive. What makes an oracle unique (compared to a smart contract, for example) is this ability to derive information from sources external to the ledger that it supports.

An oracle is necessary once a smart contract is written to interact with external data. One example is a bet on the Super Bowl, where one party picks one team, one party picks another, and an oracle determines the winner from ESPN and pushes that data to a smart contract to carry out the transaction. Similarly, within financial institutions, external data is often necessary in transactions – for example, to verify that a loan has reached maturity, one must gain knowledge about the current time.



An oracle attests to the validity of data by digitally signing facts within transactions as well as the state resulting from the transaction. A signed message from an oracle indicates proof of an event and can then be used as an input into a transaction and distributed within the transaction data itself. Once signed, an oracle cannot later “change its mind” and invalidate transactions that were previously found to be valid (allowing an oracle to change its mind would result in a loss of consensus, undermining the integrity of the system).

There are similarities between oracles and smart contracts and, in fact, they work together to carry out transactions that require data from the outside world; however, there are key differences as well. While the integrity of a smart contract depends on it being contained within the ledger itself, an oracle is independent of the entities on the ledger it interacts with and can access data outside of the ledger. Additionally, an oracle does not contain legal agreements nor can it change the state of agreements on the ledger as a smart contract does.

Once the integrity of a transaction is exposed to a third party, it creates significant vulnerabilities. Within “permissioned” or known-participant systems, trust can be relied upon and fraud and malicious attacks are much easier to monitor and police. In a permissionless system, on the other hand, the use of

oracles is more problematic as trust may be more difficult to establish.





# What is Provenance?

Provenance refers to sources/processes that produce information/data.

Provenance in Distributed Ledger Technology can be used to track the history of the asset's ownership.

The term provenance is very broad and used in many different domains including food, manufacturing, business data, art and more. At a high level, provenance refers to the sources and processes that produce a resource, piece of information, or piece of data.

For example, food provenance refers to the origin of the ingredients we purchase and eat to ensure food quality and safety. Similarly, manufacturers must understand past processes in order to ensure traceability, essential to minimize and discover inefficiencies, waste and excess cost.

The provenance of information is essential to determine whether a piece of data is trusted and authentic\*. However, while provenance is often conflated with trust, these terms are not the same (trust is derived from provenance, but is a subjective judgement).

In cryptography, provenance provides the linkage and optics to determine where something came from. As such, distributed ledger technology can be used in order to determine the provenance of an asset, which can determine history of ownership.



A distributed ledger could potentially help act like an added financial control, since (in theory) you would need to compromise some subsection of the network to change the historical record (though even then the original cryptographic signature stays the same for each individual transaction). Instead of consulting multiple systems to see if something has been rehypothecated, DLT would allow one to consult a single logical layer to determine the provenance of an asset and if it has changed ownership in the past.

Some startups in the space are currently attempting to create “provenance assurances” using distributed ledger technology in order to track the provenance of certain goods. This could potentially be applied more broadly to high value or luxury goods, for example, whose provenance would otherwise be based on paper receipts that could be tampered with or altered.



A hand reaching out to touch a digital globe with currency symbols. The globe is composed of a wireframe mesh and is surrounded by various currency symbols (Euro, Dollar, Yen, Pound, etc.) and a network of nodes and lines, suggesting a global digital network or blockchain technology.

# What is a Smart Contract?

Smart contract is a agreement whereby execution is both automated and enforceable and runs on a distributed ledger.

Smart contracts can be used to handle common contractual conditions, removing the need for intermediaries.

The term “smart contract” is used often when discussing distributed ledger technology; however, its definition tends to be broad, unclear and often controversial.

The term “smart contract” itself can be misleading, as the notion of “smartness” and the extent to which something is truly a “contract” is questionable in many instances. Smart contracts have become so closely associated with distributed ledger technology and referred to so often, that in many cases the meaning has strayed from its original intent and has been generalized to the point of inaccuracy.

The phrase and concept of a smart contract was originally developed in 1994 by Nick Szabo, a legal scholar, computer scientist and cryptographer. Szabo's background reveals the original intent and requirements of a true smart contract: a confluence of both legal and computer code. In fact, Szabo's entire methodology is centered around the improvement of contract law and practice through electronic protocols.



It is critical to distinguish between the technically accurate concept of a “smart contract” and today’s buzzword. There are often situations where a simple piece of code is claimed to be a smart contract when, in reality, it is neither “smart” nor technically a “contract”. A smart contract is not simply any piece of self-executing computer code, but is rather an agreement whose execution is both automated and enforceable.

A smart contract is automated in the sense that the core business terms (the actual “transaction” among parties) is expressed through, and independently executed by, computer code and which no party can block or otherwise tamper with. It is enforceable in the sense that it constitutes legally binding rights and obligations of the involved parties.

Currently, most smart contracts consist of a combination of both computer code – a programmable transaction protocol that defines the business terms of the contract – and legal prose that reflects that the computer code constitutes part of the binding legal agreement between the parties, and is therefore also legally binding.

Smart contracts are critical aspects of distributed ledger technology for financial institutions. They allow parties involved in an agreement to conclude with certainty that they are in consensus at all times as to the existence, nature and evolution of the facts shared among them, which are

governed by the program. Additionally, they can be used to satisfy common contractual conditions, lower transaction costs and risk, and eliminate the need for trusted intermediaries.

A hand is shown reaching out from the left side of the frame towards a digital globe on the right. The globe is composed of a wireframe mesh and is surrounded by various currency symbols (Euro, Dollar, Pound, Yen, etc.) and a network of glowing nodes connected by lines, suggesting a global digital network or blockchain technology.

# What is a State Object?

A state object captures relevant information about an agreement shared between parties.

State object references both contract code and legal prose contained by an agreement.

A state object is a digital document that represents and completely captures all relevant information about an agreement shared between parties, including its existence, content and current state.

More broadly, states can be thought of as referring to a fact at a point in time (for example, a cash state or an identity state). In Corda, states usually represent an obligation between parties. For example, a state object could represent a \$100 obligation issued by a bank, an interest rate swap, or a zero coupon bond. The Corda Vault maintains the current position of state objects upon which two firms have agreed upon. In Corda, a state object is intended to be shared only with those who have a legitimate reason to see it.

A state object references (in the form of hashes) both the contract code and legal prose contained by an agreement). States are established and can only be changed by accepted transactions, which are governed by the rules of the contract code.



Digitally signed transactions consume zero or more states and create zero or more new states (see UTXO model)\*. Some transactions dictate that a new state be created while other transactions cause current states to evolve into new states. Potential new states may be proposed by parties to an agreement, but a “consensus state” is only reached once all parties to the agreement achieve consensus.

\*A state object is either current (considered a live obligation) or historic (considered no longer valid). Note that these terms are also sometimes referred to as unspent/unconsumed or spent /consumed.





# What is Tokenization?

Tokenization is the process of referencing an asset with a “token” so that it may be more easily and quickly exchanged, or so that an existing asset may be subdivided into smaller units of value.

Tokenization within the scope of distributed ledger technology is the process of moving the registry of ownership of an asset onto a distributed ledger.

Tokenization is the process of referencing an asset with a “token” so that it may be more easily and quickly exchanged, or so that an existing asset may be subdivided into smaller units of value\*.

## Moving from Possession to Registry

Tokenization within the scope of distributed ledger technology is essentially the process of moving the registry of ownership of an asset onto a distributed ledger or a blockchain. Once an asset is on a ledger, the process of ownership exchange is simplified as an edit to the ledger that occurs on a separate cycle, ideally increasing the efficiency and velocity of settlement and potentially reducing operational risk at the same time. The mechanical settlement of an asset off-ledger may take several days, or involve the physical transfer of a large or less-liquid asset, while the transfer of the tokenized asset may occur instantaneously (or within whatever cycle is determined by the controller of the ledger, which may be centralized or decentralized).



Tokenization may use a third party that holds the asset, though it does not have to depend on a trusted third party. A swap, for example, can be tokenized between two parties that trust one another. The third party may add value to the system by removing counterparty risk, though it may also introduce both default and fraud risk.

The “token” itself has no extrinsic value – it is not a real representation of an asset on a ledger but is rather a derivative of an asset – an agreement to transfer title or rights (and in some cases, obligations) to the underlying asset. The owner of a token does not store a token on a computer; rather, the owner stores the keys that let him create a new entry on the ledger and reassign ownership to someone else.

Tokens may be *intrinsic* (built-in) to a blockchain or asset-backed (issued by a party onto a blockchain). Intrinsic tokens are fundamental to the blockchain of which they are a part, and are usually an incentive/reward to validate transactions or create blocks (BTC on the Bitcoin blockchain, XRP on the Ripple network, ETH on Ethereum).

### Asset-Backed Tokens: Digital Representations of Title Over Physical Objects

Asset-backed tokens represent ownership of an underlying asset. A non-digital example of this is bullion depository certificates, which represent the ownership of gold bullion or coins. These certificates act as proof of ownership for the physical

gold and are then able to be passed, like cash, which is much easier and more efficient than moving physical gold. Asset-backed tokens are a digital equivalent of this bullion depository certificates example. These tokens are passed between parties and these transactions are recorded on a blockchain. An issuer is required to receive a token in exchange for the underlying asset.

The movement of asset-backed tokens must accurately represent the state of the underlying asset. In order to accurately track a tokenized asset that originates off-ledger (for example, a car), the physical asset must be tagged with a unique identification method that is associated with the corresponding location or hash on-ledger.

A car is already associated with a unique, tamper-proof ID – a VIN number – that in turn can be associated with the on-ledger hash. If the asset isn't already easily associated with a unique ID, this identification method may be a physical secure device that ties back to this unique hash on ledger\*\*.



\*Another benefit and use of tokenization is to subdivide an existing asset into smaller units of value. In this case, tokenization allows for the association of a single large value asset (such as a cargo ship or aircraft) with a larger number of tokens, each representing fractional ownership of the original asset. An existing example of this would be a Real Estate Investment Trust (REIT), which subdivides large scale real estate investment into shares and allows a wider range of investors access this sector.

\*\*Such methods include a secure tag, digital locking and artificial DNA. This is an emerging field which offers the promise of multiple, indelible or tamper-proof methods of associating a physical object with its representation on the ledger.

Note: Colored coins are problematic as a tokenization method because the lack of standards means you risk the loss of the meaning of the “color” between wallets, and thus risk losing the associated data about the tokenized assets.





# What is a UTXO?

Unspent outputs can be used as inputs to a transaction. Inputs will then be converted to outputs as new UTXO that can be spent in future transactions.

A UTXO is an unspent transaction output. In an accepted transaction in a valid blockchain payment system (such as Bitcoin), only unspent outputs can be used as inputs to a transaction. When a transaction takes place, inputs are deleted and outputs are created as new UTXOs that may then be consumed in future transactions.

In the Bitcoin network, which uses this model, a UTXO is the amount that is transferred to a Bitcoin address (along with information required to unlock the output amount\*) during a transaction. Received amounts (UTXOs) are used individually during a transaction and new outputs are created – one for the receiver and, if applicable, one for the amount that is left over (change output). The amount sent to the recipient becomes a new UTXO in the recipient's address while the change output becomes a new UTXO in the sender's address that may be used in a future transaction.



Corda uses a UTXO model – an entry is either historic (sometimes referred to as "spent" or "consumed") or current (sometimes referred to as "unspent" or "unconsumed"), but it cannot be changed. Corda assumes that data being processed represents financial agreements between identifiable parties and that these institutions require a significant number of such agreements to be managed by the platform. Therefore, Corda must be able to support parallel execution, while also ensuring correct transaction ordering when two transactions seek to act on the same piece of shared state. A UTXO model provides this, allowing transactions to be processed independently and in parallel, even for high traffic legal entities.

