MongoDB®

Blockchain, Ledgers, and Databases :

A Guide to Navigate the Confusion

January 2022

1633 Broadway, 38th Floor, New York, NY, 10019 United States | mongodb.com

Table of Contents

1. What is blockchain?

Blockchain is a fast-growing technology with both hype and reputability, making it exceptionally exciting and buzzworthy. However, what is it really and when does it actually come into play for use? According to the National Institute of Standards and Technology Blockchain Technology Overview, “Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e. a bank, company, or government). At their basic level, they enable a community of users to record transactions in a shared ledger within that community, such that under normal operation of the blockchain network, no transaction can be changed once published.”

Overall, blockchains are characterized by three distinct key features:

- Decentralization (data is stored by multiple [legal/non-trusted] entities)
- Append-only (data once stored is not easily deleted)
- Immutability (attempts to change stored data is easily evident)

There are roughly two types of blockchain networks in existence today - permissionless and permissioned. An example of permissionless blockchains would include Bitcoin or Ethereum. These blockchain networks are characterized by the fact that anyone can download the Bitcoin or Ethereum software and join the respective network without requiring permission from anyone.

In the permissioned blockchains, entities (e.g. companies) establish a governance structure for adding and removing new entities or sharing data using the underlying permissioned blockchain. Hyperledger Fabric, for example, is a permissioned blockchain, as all members are known and authenticated..

Regardless of the blockchain type, permissionless or permissioned, they share the above three key features: decentralization, append-only, and immutability. Essentially, they are each a form of a distributed ledger. However, not all distributed ledgers are blockchains. This is because a ledger can be distributed, such as for high availability, but still managed by a central authority.

2. What properties of blockchain are you really interested in?

When it comes to choosing the right tools to address business needs, certain considerations need to be made, such as business values, prioritizations, and what trade-offs make sense to the business applications. The three aforementioned blockchain properties (i.e. decentralization, append-only, and immutability) can fit into a business strategy in the following ways:

Decentralization

Decentralization is often the most confusing aspect of blockchain, and it is not uncommon to conflate it with distributed. The confusion arises because a decentralized system is a distributed system by definition- each entity participating in a decentralized system owns a full copy of the data and must agree on the changes through a consensus algorithm.

Append-only structure

Blockchains store data in an append-only (added, not overwritten) manner in order to facilitate data integrity/immutability. Once new information is stored, it should not be possible to delete it. Any modifications result in a new copy of the data being stored in the blockchain, this copy is also linked back to the earlier version, thus producing a complete chain of events that are also cryptographically linked. This historical record of information storage in an append-only structure is a key property of blockchain, but it is not unique to it.

Applications often have a requirement to store historical records of the information change. This is not a unique feature and will be discussed further along in this paper. For example, think account credit/debit, where the history of all transactions of a bank account is maintained. This application-level ledger scenario is not unique to banking or blockchain, and arises in many different applications as a necessary feature.

While applications need to maintain a history of information change, they are often faced with a regulatory requirement of eventually deleting that information after a certain time has elapsed or if an application user invokes the right to erasure; this is common practice for government compliance and user rights. If faced with such a requirement, storing data in a permissionless or permissioned blockchain must be carefully evaluated.

Data integrity/immutability

Blockchains achieve data integrity and immutability by providing mechanisms for tamper resistance and tamper evidence together with their append-only structure. Informally, tamper resistance means that it is difficult (computationally or otherwise) to change data once it is stored. Tamper evidence means that once data is stored, any changes to it can be easily detected. Blockchains achieve these two properties using decentralization, cryptographic proofs, and append-only structure.

However, this has its fallbacks too. Data distribution and cryptographic proof in a decentralized system add computational overhead. When new information needs to be stored in the blockchain, some or all nodes in a permissioned or permissionless blockchain network have to agree on the change. As previously mentioned, the change is validated through consensus algorithms (e.g. Proof of work for Bitcoin), mechanisms designed to prevent malicious behavior that constitute the key technology of decentralized systems. The change validation process is typically slower than say the transactional databases by an order of magnitude. Therefore, it should be asked whether the use case justifies the higher cost. For instance, if we are dealing with high-volume time series measurements that don't actually need extra guarantees, it may not be worth storing it in the blockchain network. On the other hand, if we are working with any low-volume data that might need to be audited in the future, such as asset tracking, financial transactions, or event logging, it might be worth the effort.

Databases have long maintained tamper resistance (access control) and tamper evident (audit logs) mechanisms in order to determine who did what with the information stored in the database. What differentiates blockchain from a database's tamper-evident mechanisms is that the data integrity information in blockchain is stored right next to the information itself, and it can be used to verify tamper evidence.

3. Ledgers and databases

Ledgers are one of the oldest forms of data storage, and if an entity does not require the decentralization property of blockchain, but instead is interested in an append-only database with data integrity and immutability mechanisms, they are probably looking for

a ledger, managed and governed by a single trusted authority. Before diving into ledgers, let us break down how databases typically process information.

A typical database updates or deletes information by overwriting existing information:

{ key1: val1 } --> {key1 : val2}

In the above example, the value “val1” of “key1” has been updated to “val2”. In order to keep track of previous changes to a piece of information already stored, a database often provides auditing capability which can record who did what as well as a previous version of that information. Some databases also provide the capability to emit or query the prior version of the information that was changed with an update. Once information is deleted from the database, it is eventually deleted from all the internal structures of the database. This deletion operation can also be recorded in the audit logs. If someone wanted to track how data was changed by solely relying on database semantics of auditing or emitting a prior version, they would have to take the existing data stored in the database and correlate against the audit logs or the prior emitted version stored elsewhere. This task is cumbersome.

Let’s now compare how a ledger processes information:

For a typical “create” or “insert” operation, a ledger always adds new information alongside the existing one.

When an existing information is updated or deleted, a ledger adds new information alongside the old one, thus “remembering” the past:

{ key1: val1 } + {key1 : val2}

As shown in the figure above, the “+” symbol means “appended to”, similar to blockchain. When the application queries the value of “key1”, it would obtain its latest value “val2”. If an application wanted to query all the previous values of “key1”, it would invoke a “historical query” function to retrieve some or all previous values of “key1”.

Why is this structure better than auditing or storing the emitted prior version for analyzing what changed? The change is stored along with the information, this way applications can use existing database query semantics to retrieve the latest or historical versions without

combining audit or prior version data that is typically stored elsewhere (e.g. log files) or another database.

This approach creates a transparent history of records being stored, and makes it easy to query and audit the history. Moreover, similar to blockchain, this approach provides the foundational capability to add data integrity with cryptographic verification, thus enabling the blockchain capabilities of tamper resistance and tamper evidence in a ledger.

In scenarios where a “hard delete” may be required due to regulatory reasons, such as GDPR, ledgers can provide a mechanism to hard delete the old data while still maintaining tamper resistance and tamper evidence mechanisms, but without the overhead of decentralization associated with blockchains. Thus, ledgers’ flexibility can make them more suitable for many real-world applications.

4. What classes of applications can benefit from ledger properties?

Properties of ledgers can easily be leveraged to reduce friction, streamline processes, and improve trust among parties. Some common real-world use cases include:

Financial services

Financial transactions can be broadly broken down into three steps: submission, clearing, and settlement. The transaction is initiated by an entity, its identity needs to be validated and availability of funds verified to proceed to settlement, where the asset is finally transferred. The asset transfer between two end users such as households or businesses is typically initiated through a bank or a broker-dealer, which interacts with other intermediaries, such as financial market infrastructures (FMIs), that often operates on a multilateral basis, thus connecting other banks and ultimately other end users. In this simplified model, each bank or broker-dealer owns its copy of the transaction ledger (a general ledger typically used in accounting, not the one described in the above Section 3) of trades with the FMI, while the FMI owns a ledger containing all the transactions going

through it. A crucial aspect of the process is verifying that the records of the various actors agree. This step is called reconciliation.

As has already been proven in several studies, decentralized ledgers support the high volumes of real-world trading scenarios. However, maintaining and operating such infrastructures can be challenging both from a technical and a regulatory standpoint, as the system must be kept up and running by all the nodes, data access, and security needs to be properly managed. Do financial institutions really want to take this burden on their shoulders?

In this context, financial institutions are interested in three key requirements which immediately translate into risk reduction and transparency:

- Immediate clearing
- Immediate settlement
- Immediate reconciliation

This scenario can be achieved by using a centralized ledger. Before digging into the implementation details, we examine two possible configurations:

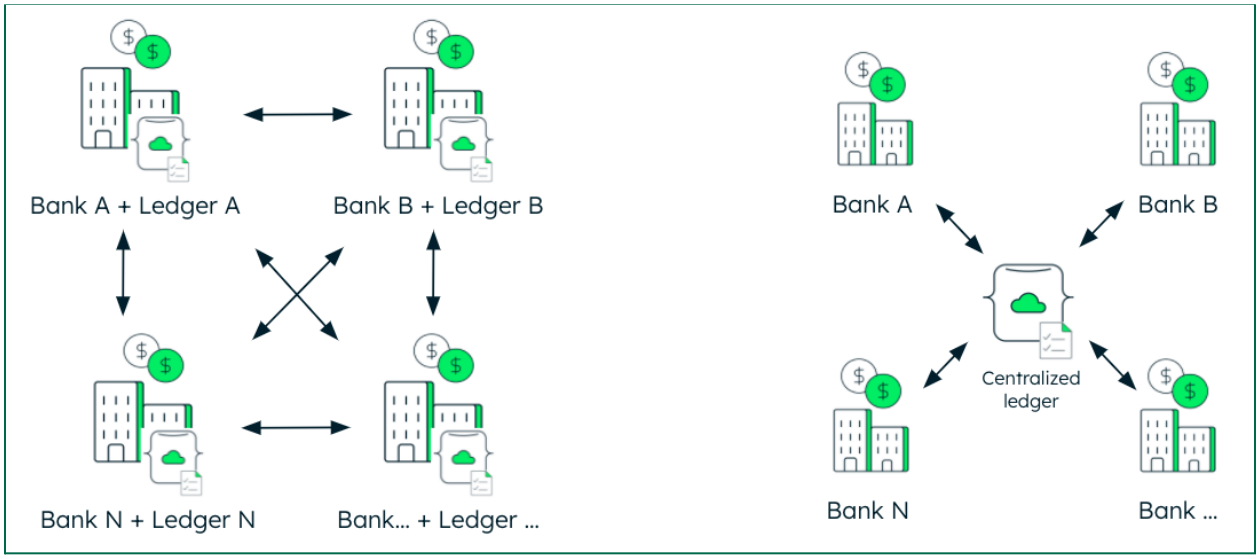

Figure 1. Two configurations: (left) each bank owns its own ledger vs (right) central entity owns it

On the left side of Figure 1, each bank owns and operates its own ledger and all the other banks connect to it. In the second setup, on the right, one ledger is managed by a central

trusted FMI (e.g. the Depository Trust & Clearing Corporation - DTCC) and each bank only connects to that one.

Does the first option really make sense to financial institutions? Are they willing to build and maintain their own ledger and all the costs associated with that or would they be happier having a separate entity doing it for them?

If the second option is the preferred one, then a centralized ledger managed by a trusted third party guarantees performance, security and cost effectiveness. A MACH based architecture provides an agile, nimble and future-proof solution that perfectly dovetails with the principles of Open Banking, the new standard default in banking, providing a richer customer experience as well as the building blocks for the financial ecosystem to thrive.

Supply chain and product tracking

Supply chains are intricate networks involving a great variety of stakeholders interacting with each other. Regulatory standards compliance, origin guarantee, authenticity of goods and transparency of ethical and ecological standards of practices are only a few of the challenges that an ever globalized and fast-changing world is posing to supply chains. Their traditional management processes, characterized by expensive and error-prone steps such as on-site inspections and record reconciliation, are proving to be inadequate at solving such criticalities. Also, outdated IT systems and siloed data hinder interoperability and automation.

Each and every step of the chain entails an exchange of data. Issues of accountability, trust and transparency among parties can be strongly mitigated by the introduction of a ledger serving as a single source of truth. The stakeholders benefit from increased visibility, and data integrity guarantees each step of the process.

The adoption of technologies such as Internet of Things (IoT) sensors and Radio Frequency Identification (RFID) tags that connect the physical and the digital world, are further guarantees of data reliability and product quality. In fact, such sensors can directly update the ledger, transmitting real-time information of goods and eliminating manual data entry issues. For example, the pharmaceutical industry is testing such a solution to ensure compliance and to reduce costs of its supply chain by tracking each step (IoT sensors directly updating a ledger) of the transport of medical products that need to be stored within a specific temperature range, thus forcing accountability of the parties involved (e.g. carriers). Embedded RFID tags uniquely identify objects to prevent

their substitution with a fake. This is especially relevant for luxury goods, whose value is connected to their provenance.

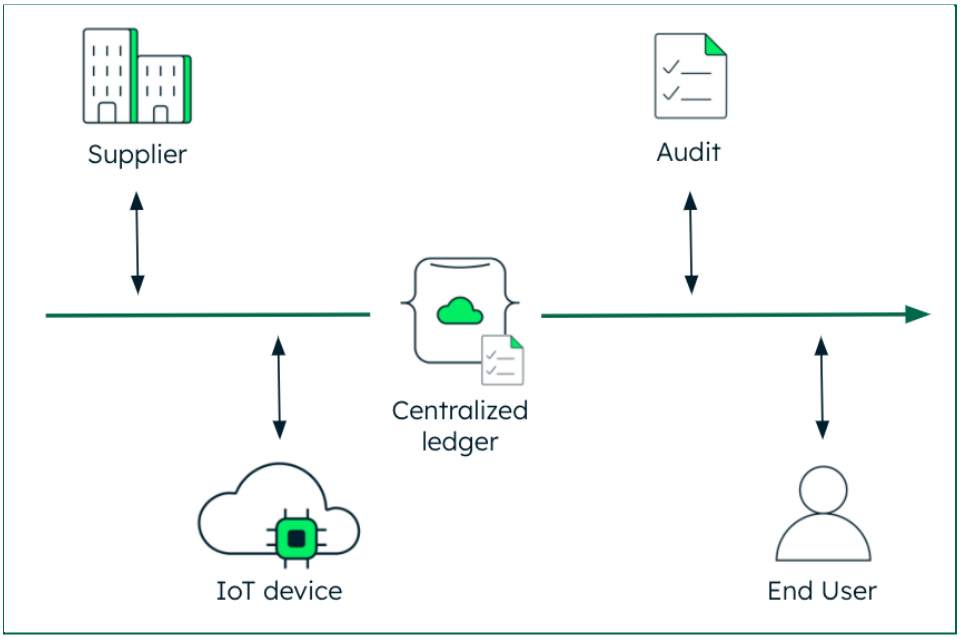

Figure 2. Example of supply chain and different parties involved

Furthermore, as consumers and investors are increasingly concerned about corporate ethics and sustainability, ESG criteria compliance is becoming ever more relevant to companies striving to ensure trust and brand reputation. Innovation in product traceability and supply chain transparency are key factors to provide the data quality and reliability needed for ESG reporting.

Similarly to financial services, for the supply chain use case we should again ask the question: do participants such as farmers, suppliers, or custom authorities want to maintain and operate the ledger infrastructure? Would it be easier for them to offload the heavy lifting on to a trusted party that already exists in the supply chain process such as the end retailer or major distributor? If that is the case, the best option may be a centralized permissioned ledger where its owner authorizes a list of users external to the organization to access and update records as it deems appropriate (e.g. a supplier can read and write data related to its section of the supply chain), thus avoiding data confidentiality and protection issues that arise from decentralized storage of data. At the same time, it allows the organization that owns it to have a complete view of the entire ledger. Moreover, immutability and tamper evidence improve trust between parties, and can potentially reduce the number of reconciliations.

5. MongoDB, Blockchain, and Ledgers

After discussing the potential benefits, scenarios, and use cases for both blockchain and ledgers, and breaking down the distinct differences and similarities of each, where does MongoDB come into play? The MongoDB data platform can be used in both blockchain networks and ledgers, highlighting the flexibility that distinguishes it.

Blockchain

Starting with blockchain, MongoDB data platform can be used as an “on-chain data store”, an “off-chain data store for analysis”, or simply as an “off-chain chain data store” for a crypto exchange to allow its users to transact digital assets.

On-chain data store: A node in a blockchain network needs to store some or all of the blockchain data. Conceptually, MongoDB can be used as the data store of a blockchain node.

Off-chain data analysis and crypto exchanges: Entities may need to analyze the blockchain data for operational, legal, and other use cases. In such a setting, entities can store the blockchain data from their participating node into MongoDB Atlas, and then analyze it using various features of MongoDB Atlas such as MongoDB Charts, Data Lake, and Search.

A crypto exchange may allow its participants to exchange a digital asset represented on the blockchain without actually transacting it through the blockchain network. For improved traceability, these assets can be stored in a ledger with cryptographic proofs. For example, crypto exchanges, such as Coinbase, already use MongoDB to underpin its systems of off-chain operations (e.g. wallet management, transaction categorization, NFT custody, etc.).

Ledger proof of concept using MongoDB

A proof of concept version of a ledger built using MongoDB is available on MongoDB Labs github as a python library (https://github.com/mongodb-labs/ledger). The library augments the collection object available in MongoDB's python driver with the ledger functions of insert, update, delete, and verifying change history. Let us break down these functions one by one.

Inserting data: The library provides a method insert_one_ledger() to insert a document into a collection. Semantically, this method is the same as the insert_one() method in MongoDB's python driver, but with an important distinguishing detail- the method also inserts the document into a collection managed by the library that maintains all historical updates to all the documents stored in this collection. The name of this collection is the same as the original collection, but with a "_history" suffix. For example, if the name of the collection in which the document is being inserted is "foo", the history collection name is "foo_history".

Updating data: The library provides a method update_one_ledger() to update an existing document in a collection. Semantically, this method is the same as the update_one() method in MongoDB's python driver, but it performs the following additional steps. The method retrieves the last version of the document, if any, from the history collection, computes the SHA256 hash of the last version of the document along with the update being stored, updates the main collection with the new version and the new hash, and finally appends the new version of the document and the new hash to the history collection.

Verifying change history: The library provides a method verify_one_ledger(), which can be used to verify that the historical updates of the document match with their stored hashes.

Deleting data: The library provides a method delete_one_ledger() which deletes a document from a collection. Semantically, this method is the same as the delete_one() method in MongoDB's python driver, but it performs the following additional steps. It stores the last version of the document along with the last operation performed on the document, namely, delete. Upon querying the document by its identifier from the main collection using the find_one() method in MongoDB's python driver, no document is returned.

6. Conclusion

The three main properties of blockchain, decentralization, append-only structure, and immutability, make it an attractive technology for certain applications, but it comes with its own issues of data governance, low performance, and legal and regulatory complexity, just to name a few. However, append-only structure and data integrity/immutability

properties that often attract many to blockchain to begin with are readily achievable via ledgers and databases, and without the overhead of decentralization.

In real-world use cases, the flexibility and resilience of MongoDB's data platform allows it to be used as an "on-chain" store, or as an "off-chain data store" to either analyze blockchain data, or as an append-only ledger with cryptographic proofs. Regardless of the use case or application, to learn more about how MongoDB is a secure solution for your organization's unique needs, connect with the MongoDB team here.

About the authors

Luca Napoli, an Industry Solutions Consultant at MongoDB, is a part of the Industry Solutions Team. With a background in energy, innovation, sustainability, data science and artificial intelligence, he has a passion for data thought leadership and technology.

Salman Baset works in MongoDB's product team where he is responsible for the strategy and execution of security and compliance for MongoDB Server and Cloud, and MongoDB's FedRAMP Initiative. Previously, he worked at IBM, where in his last role as CTO Security, he was responsible for the security, compliance, and identity management for blockchain solutions such as IBM Food Trust and Tradelens. He has over 25 patents issued.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.
Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Realm (mongodb.com/realm)

Legal Notice

This document includes certain "forward-looking statements" within the meaning of Section 27A of the Securities Act of 1933, as amended, or the Securities Act, and Section 21E of the Securities Exchange Act of 1934, as amended, including statements concerning our financial guidance for the first fiscal quarter and full year fiscal 2021; the anticipated impact of the coronavirus disease (COVID-19) outbreak on our future results of operations, our future growth and the potential of MongoDB Atlas; and our ability to transform the global database industry and to capitalize on our market opportunity. These forward-looking statements include, but are not limited to, plans, objectives, expectations and intentions and other statements contained in this press release that are not historical facts and statements identified by words such as "anticipate," "believe," "continue," "could," "estimate," "expect," "intend," "may," "plan," "project," "will," "would" or the negative or plural of these words or similar expressions or variations. These forward-looking statements reflect our current views about our plans, intentions, expectations, strategies and prospects, which are based on the information currently available to us and on assumptions we have made. Although we believe that our plans, intentions, expectations, strategies and prospects as reflected in or suggested by those forward-looking statements are reasonable, we can give no assurance that the plans, intentions, expectations or strategies will be attained or achieved. Furthermore, actual results may differ materially from those described in the forward-looking statements and are subject to a variety of assumptions, uncertainties, risks and factors that are beyond our control including, without limitation: our limited operating history; our history of losses; failure of our database platform to satisfy customer demands; the effects of increased competition; our investments in new products and our ability to introduce new features, services or enhancements; our ability to effectively expand our sales and marketing organization; our ability to continue to build and maintain credibility with the developer community; our ability to add new customers or increase sales to our existing customers; our ability to maintain, protect, enforce and enhance our intellectual property; the growth and expansion of the market for database products and our ability to penetrate that market; our ability to integrate acquired businesses and technologies successfully or achieve the expected benefits of such acquisitions; our ability to maintain the security of our software and adequately address privacy concerns; our ability to manage our growth effectively and successfully recruit and retain additional highly-qualified personnel; the price volatility of our common stock; the financial impacts of the coronavirus disease (COVID-19) outbreak on our customers, our potential customers, the global financial markets and our business and future results of operations; the impact that the precautions we have taken in our business relative to the coronavirus disease (COVID-19) outbreak may have on our business and those risks detailed from time-to-time under the caption "Risk Factors" and elsewhere in our Securities and Exchange Commission ("SEC") filings and reports, including our Quarterly Report on Form 10-Q filed on December 10, 2019, as well as future filings and reports by us. Except as required by law, we undertake no duty or obligation to update any forward-looking statements contained in this release as a result of new information, future events, changes in expectations or otherwise.