Building Enterprise-Grade Blockchain Databases with MongoDB

November 2017

Table of Contents

Introduction

Bitcoin was released in 2009 as a digital currency system via a simple posting to a niche mailing list. Over time, interest grew not only in the digital currency, but also the underlying blockchain technology. This has led to a global blockchain movement that has the potential to transform not only each industry that it touches, but also the internet itself.

The World Economic Forum forecasts that blockchain technologies will become the "beating heart of finance" within five years. In November 2016 Barclays claimed they had completed the world's first blockchain-based global trade transaction – compressing a process that would normally have taken up to 20 days to just a matter of hours.

Beyond financial services, blockchain technology has the potential to fundamentally transform interactions across almost every industry – from media and entertainment, to logistics and supply chain, medicines and patient care, energy trading and loyalty programs, through to government and military applications.

What is key to all of these use cases is the elimination of friction. The blockchain movement is following a path established by earlier technology revolutions:

- In the 1990s, mass adoption of the internet removed friction from the creation and distribution of information and content.
- In the 2000s, mass adoption of open source software removed friction from developers accessing the building blocks to create new applications.
- In the early 2010s, mass adoption of cloud computing and distributed systems removed friction from acquiring and scaling data center infrastructure, while freeing enterprises from expensive, legacy technology stacks.
- Now as we approach the 2020s, we are moving towards mass adoption of blockchain technologies.

The blockchain movement removes friction along three key axes: control, trust, and value, which in turn unlock new applications & opportunities:

1. **Control** – in the new model, multiple entities share control, that is, the system is **decentralized**. A fully

public blockchain network acts much like a public utility. Or, in consortiums, different parties (even traditional competitors) can more easily share data infrastructure to the benefit of all participants. Finally, even a single enterprise can dilute risk by having multiple system administrators share control of that organization's IT infrastructure.

2. **Trust** – in the blockchain model, writes are considered **immutable** to enable tamper-resistant audit trails, for applications from supply chain tracking to art provenance, from financial auditing to food safety. It also gives a single shared source of truth, which simplifies applications from financial reconciliation to tracking music rights.
3. **Value** – now, one can **issue and transfer assets**, without reliance on a central entity. For example, you own (and can transfer) the bitcoins at an address if you have the private key at that address. This concept works for almost any digital or physical asset. Accordingly, there are significant opportunities to streamline and better-secure registries and exchanges, such as land registries and stock exchanges.

We can summarize as follows. Removing friction along the axes of control, trust, and value manifest directly as three blockchain characteristics respectively: **decentralized control, immutability, and assets**.

With these characteristics in mind, let's explore some sample applications. Following that, we will examine conceptual deployment architectures and the role of blockchain scalability.

Emerging Blockchain Applications

This section explores sample applications; and which blockchain characteristics – decentralized, immutability, assets – they exploit.

Decentralized. Blockchains instantiate trust between a transaction's counterparties, eliminating the need for central authorities to arbitrate inter-party transactions. Figure 1 shows how clearing financial transactions changes as one goes from a centralized ledger to a decentralized one, i.e. a blockchain. Note that some in the industry call it a "distributed" ledger though "decentralized" is more accurate as distributed systems may still be centralized under the control of a single party.

Blockchains even allow competitors to work together for a common benefit. An example is R3, which has dozens of financial institutions exploring how to share their data infrastructures by applying some principles from blockchain design, and the Open Music Initiative for the music labels.

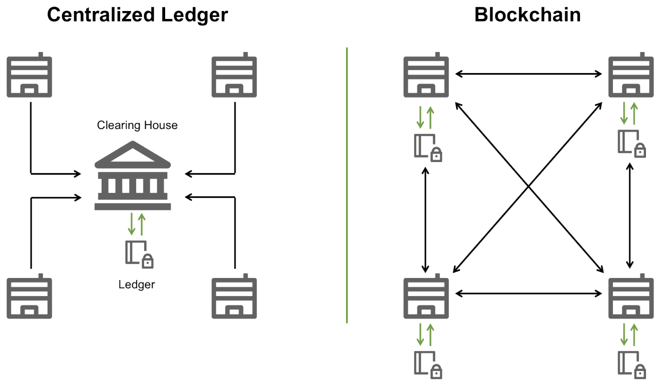


Figure 1: A blockchain can replace a centralized ledger for reduced complexity, cost, and delay

Immutability. This manifests itself in audit trails, or provenance (the history of ownership). Walmart is testing blockchains to trace food through its supply chain – from farm to fork – helping ensure food authenticity and safety, while reducing wastage. Europe's largest shipping port is joining a consortium to use blockchains for sharing of logistics and contract information, while research in Finland is focused on integrating shipping containers with Internet of Things (IoT). An initiative within the Hyperledger project is building a trusted record of the provenance of medical drugs, potentially attacking the $75B per annum market in the supply of fake medications, and the tens of thousands of lives lost to dangerous counterfeits. In art, verified sources are a prerequisite to value; 10,000 artists use ascribe for blockchain-secured digital and physical art provenance.

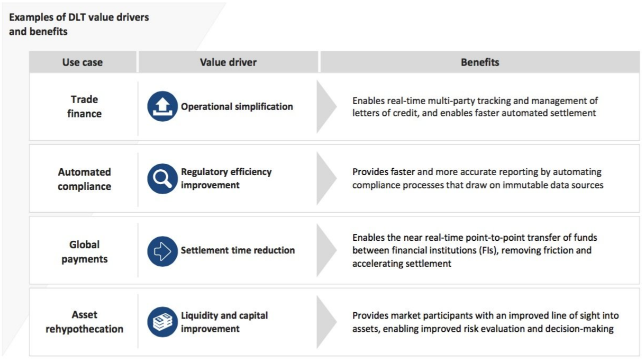
Illustrating broader industry applicability, MarketsandMarkets research predicts that the fastest growth in blockchain spending will come from the media and entertainment industry. It will use blockchains to assert copyright, manage rights transfer, and help distribution for content creators and owners.

In the public sector, the Estonian government uses blockchains to help secure citizen data across a range of services including e-voting, medical services, and filing tax returns. Honduras and the Republic of Georgia are both experimenting with blockchains for property title registration. Educational institutions are exploring blockchains to track student progress and record academic certificates for verification by potential employers, while DARPA is seeking proposals for a "secure messaging system" that would use blockchain technology to facilitate the broadcast of encrypted secrets in a transparent fashion between Department of Defense units.

Assets. Financial clearing, as depicted in Figure 1 is also an example that benefits from blockchain-style asset transfer. In this case, asset transfer may be automatically triggered by other events, typically via an "escrow" or "multi-signature" functionality. For example, invoices can be paid when a delivery is recorded, or property title deeds can be transferred to the new owner once mortgage funds are released.

The expected efficiencies gained from settlement time alone are compelling. Accenture predicts that settlements on corporate bonds, equities, and debt instruments can be reduced from 3 days to seconds, while syndicated loans taking nearly 3 weeks to settle can clear in less than 24 hours.

In its Future of Financial Infrastructure analysis, the World Economic Forum predicts 80% of banks will have initiated blockchain projects by the end of 2017, identifying four main financial industry use cases for the technology, as illustrated in the figure below.



Examples of DLT value drivers and benefits

Use case	Value driver	Benefits
Trade finance	Operational simplification	Enables real-time multi-party tracking and management of letters of credit, and enables faster automated settlement
Automated compliance	Regulatory efficiency improvement	Provides faster and more accurate reporting by automating compliance processes that draw on immutable data sources
Global payments	Settlement time reduction	Enables the near real-time point-to-point transfer of funds between financial institutions (FIs), removing friction and accelerating settlement
Asset rehypothecation	Liquidity and capital improvement	Provides market participants with an improved line of sight into assets, enabling improved risk evaluation and decision-making

Figure 2: Financial services use cases for blockchains. Source: World Economic Forum

All of the examples above represent just some of the use cases for blockchains, and are joined by potential new applications every week.

Blockchains in the IT Stack

The blockchain characteristics of decentralization, immutability, and assets also help us reason about real-world & enterprise IT (Information Technology) systems and applications. How do those characteristics relate to the IT stack, and databases in particular? How does scale play a role? What are the deployment scenarios? We explore these questions now.

Usually, "blockchain" is treated as a noun, assuming one monolithic type of system, which is useful for providing a high-level conceptual picture to the market. But for decision-making, we need to be more fine-grained. Let's consider blockchain characteristics in each building block of the IT stack, and in particular processing, file systems, and databases. Figure 3 illustrates three different levels of the centralization-to-decentralization spectrum. Let's explore further.

Figure 3 left shows a centralized stack, where a single entity controls each part of the application infrastructure. Centralized systems power all modern IT systems from finance to supply chain to web properties and the IoT (Internet of Things). They have achieved massive scale by being distributed – spreading the processing and storage of workloads across more than one physical machine. MongoDB has emerged as a leading database in these classes of distributed systems.

In Figure 3 middle, partly decentralized applications add in only as much blockchain technology as needed to achieve some of the benefits described above (control, trust, value). The remaining application infrastructure remains centralized to maintain scale, ease deployment, and leverage existing IT investments. For example, we can add a blockchain platform beside existing centralized processing, file systems, and databases. That blockchain platform could be running MongoDB as its core database, thereby inheriting MongoDB's distributed data management characteristics.

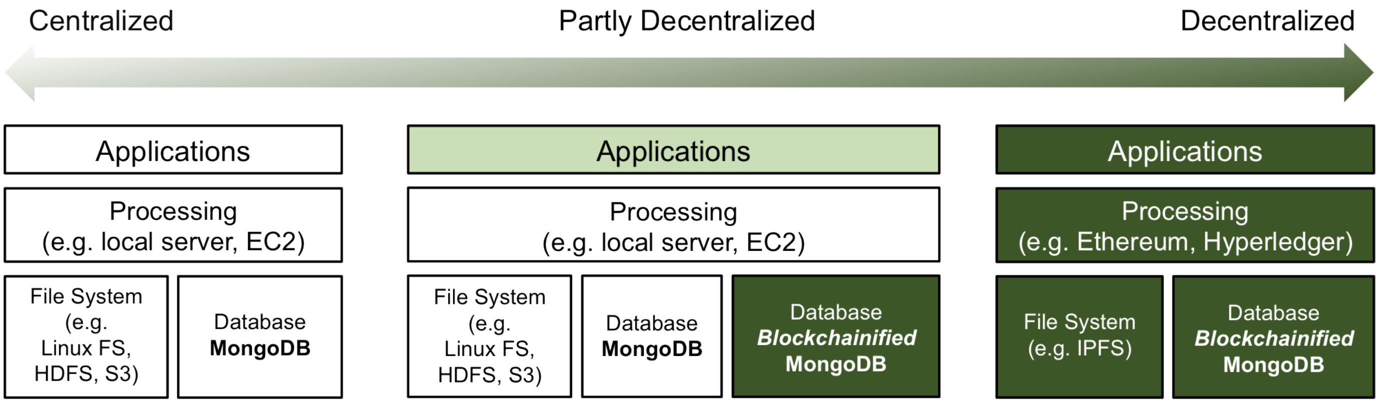

Centralized
Partly Decentralized
Decentralized
Applications
Processing (e.g. local server, EC2)
File System (e.g. Linux FS, HDFS, S3)
Database MongoDB
Applications
Processing (e.g. local server, EC2)
File System (e.g. Linux FS, HDFS, S3)
Database MongoDB
Database Blockchainified MongoDB
Applications
Processing (e.g. Ethereum, Hyperledger)
File System (e.g. IPFS)
Database Blockchainified MongoDB

Figure 3: IT stacks, from today's centralized systems (left), to partly decentralized systems (middle), to fully decentralized systems (right)

On Figure 3 right is a future vision, in which every IT part of the application infrastructure is decentralized. There is dedicated application infrastructure for decentralized processing (aka smart contracts), decentralized file systems, and decentralized databases. While there is much discussion around fully decentralized systems, it comes with trade-offs that enterprises need to consider – specifically much lower scalability, complexity of development, operations, and privacy, and the abandonment of existing IT investments. The reality is that complete decentralization is not practical for many enterprise-grade blockchain projects.

The sweet spot for enterprise blockchain usage over the next several years is in partly-decentralized applications. We also see that the key benefits are audit trails and tracking who-owns-what; these are functions of the database infrastructure. Therefore, this paper will focus on blockchain databases for deployment to partly-decentralized applications.

Blockchain Deployment Architecture

Enterprises have come to expect databases to offer rich querying, scale, and to be operationalized for production-hardened deployment.

- **Querying** is a prerequisite to truly being a database, compared to a simple file system or ledger. With rich querying, we can serve a range of applications that need to consume data stored in the blockchain for operational and analytical tasks.
- **Scale** includes high capacity, high throughput, low latency, and the ability to improve performance as more hardware is added. Data partitioning, or ”sharding” is a prerequisite to horizontal scale-out, so that each new hardware node is only storing a subset of the data, and handling a fraction of requests.
- **Operationalized**. MongoDB has risen to these querying and scale challenges, and from ten years of hardening against demanding enterprise applications in over 50% of the Fortune 100, can easily be considered operationalized.

On top of those characteristics, we need the three blockchain characteristics: decentralization, immutability, and assets. How do we achieve all six characteristics? This becomes an engineering challenge.

One might consider starting with a traditional blockchain and giving it queryability, scale, and “operationalized” characteristics. Each characteristic is hard to achieve on its own; and the combination of multiple characteristics is even harder. For example, to achieve scale, one needs sharding, where each node stores a subset of the data; but then queries must manage data distribution.

But there is another way to marrying blockchain with enterprise-grade database capabilities: we can start with MongoDB, and add a blockchain control layer above it. In other words, we can blockchain-enable MongoDB. Figure 4 illustrates this combination. On the left is MongoDB in a traditional (centralized) setting, where there is just one administrator representing one organization that can control all the data. There may be thousands of clients or users.

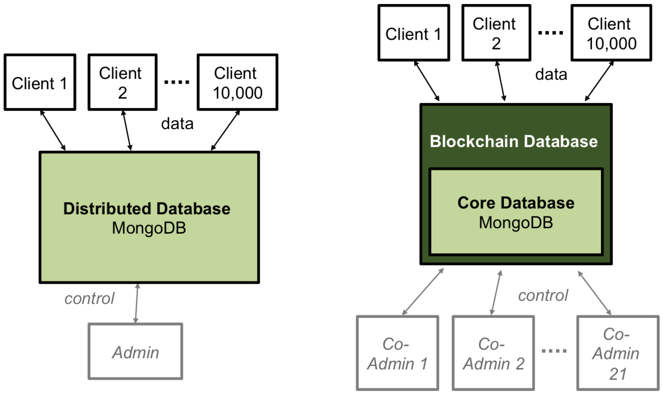


Figure 4: Left: MongoDB deployment architecture. Right: Blockchain database deployment architecture

On Figure 4 right is a **blockchain database**: a blockchain-enabled MongoDB that wraps the core database (MongoDB) and implements the three blockchain characteristics of decentralization, immutability, and assets. This figure shows the conceptual deployment architecture. A blockchain database can't have just one administrator, since that would be centralized. Rather, there are several co-administrators, which collectively share control.

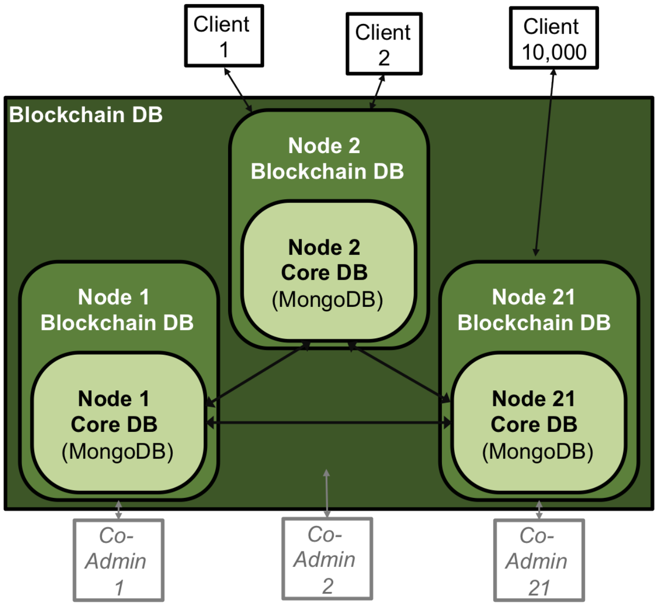


Figure 5: Internal architecture of a blockchain database

Figure 5 drills deeper into the blockchain architecture, to show how the three blockchain characteristics are implemented as part of the solution.

- **Decentralization.** Each database co-admininstrator controls one of the nodes in the database. Each node includes a MongoDB node, and the blockchain-enabling software that wraps the node. Together, these nodes form a federation, which by definition is decentralized (no single entity controls it). Each authorized node in the federation votes on whether each incoming transaction is valid.
- **Immutability** is engineered-in via continuous backups to write-only media (drawing on MongoDB functionality), hashes on groups of transactions ("blocks") pointing to previous blocks, and more.
- **Assets.** The ability to create and transfer assets is part of the API, where the client creates a transaction – manifested as a MongoDB document – and then signs it with client's private key.

Being a distributed database, MongoDB already has fault-tolerant consensus. The blockchain-enabling software layer handles higher-level consensus to prevent double-spends and other Byzantine (malicious) behavior. In this model, the identity of each federation node is known; therefore one does not need to solve for Sybil attacks ("attack of the clones").

Benefits of a Blockchain Database

In blockchain platform powered by MongoDB, we achieve all the target characteristics of an enterprise blockchain database; we get the best of both worlds. Table 1 illustrates.

	Characteristic	**MongoDB**	**Traditional Blockchain**	**Blockchain-Enabled MongoDB**
Database Characteristics	**Queryability**	✔		✔
	Scale	✔		✔
	Operationalized	✔		✔
Blockchain Characteristics	**Decentralized**		✔	✔
	Immutable		✔	✔
	Assets		✔	✔

Table 1: We can "blockchain-enable" MongoDB to incorporate blockchain characteristics, resulting in an enterprise blockchain database

Blockchain Database Deployment Scenarios

There are two axes affecting deployment decisions:

- Is the blockchain database being deployed **within an enterprise** or **within a consortium**?
- Is the blockchain data "operational" (i.e. directly used by clients) or not (i.e. there is an intermediary centralized database)? "Operational data" refers to data from existing, non-blockchain business processes, some of which is stored in the blockchain database, but is not directly accessed by clients.

These factors lead to four possible deployment scenarios, as Figure 6 illustrates. The following sections describe each scenario in detail.

Is blockchain data operational?	Yes – blockchain data is used directly by clients	Scenario in Fig. 7	Scenario in Fig. 9
	No – there is an intermediary centralized database	Scenario in Fig. 8	Scenario in Fig. 10
		Within enterprise	**Within consortium**

Deployment?

Figure 6: Blockchain database (blockchain-enabled MongoDB) deployment scenarios

There are analytics / business intelligence (BI) variants for each of these scenarios as well, which are beyond the scope of this paper.

Scenario: Within the Enterprise, Blockchain Data is Operational

Figure 7 illustrates this scenario. The blockchain data is used directly by clients, i.e. it is operational. Clients may be within the enterprise, or external to it.

Deployment is within the enterprise. This possibility may come as a surprise to blockchain enthusiasts, because enterprises have centralized control, e.g. the CEO. However, that places too much focus on just one of the three blockchain benefits (decentralization); it is the other two benefits (immutability and assets) that apply here. One gets partial decentralization by simply having more than one administrator for the blockchain database, for example one admin for each regional office, or for each business line. This is sufficient to unlock the advantages of immutability and assets for practical use cases in the enterprise.

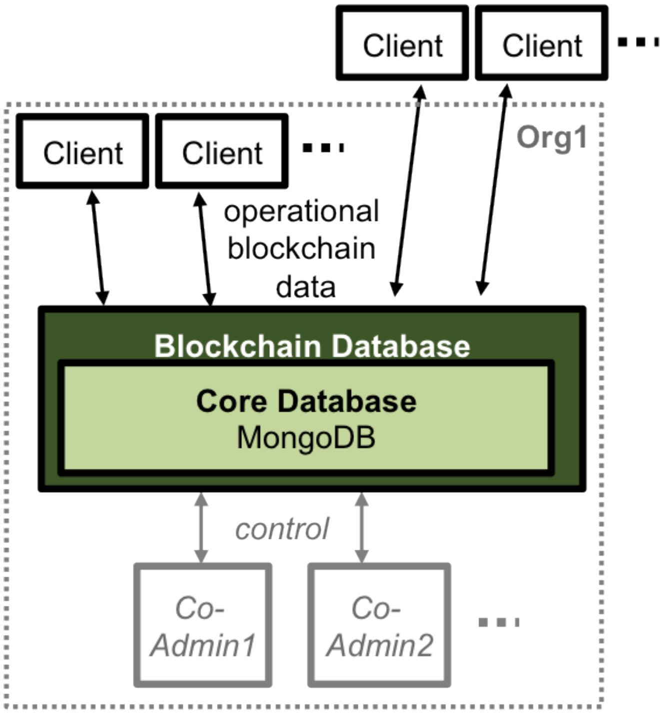


Figure 7: Deployment scenario: blockchain data is operational; deployed within the enterprise

Within-enterprise deployment has several benefits. First, it's familiar – this is a very similar model to regular application deployments. Second, it's easier for the enterprise to reason about privacy and regulatory compliance if all the data is clearly maintained within its walls. Finally, developers can make immediate progress on their own blockchain implementation, while still participating in what are often slower evolving consortium-based approaches.

An example use case is Foreign Exchange (FX) reconciliation, where the blockchain database serves as an immutable audit trail. Every transaction committed in the FX flow (front and back office) is persisted into the database. If there is ever an issue, the FX reconciliation team can quickly examine all related transactions, compressing a process that might take days into minutes or hours. The front office, back office, and FX reconciliation teams are the internal clients to the database; and 3rd party auditors would be external clients.

Scenario: Within the Enterprise, Blockchain Data is Not Operational

Figure 8 illustrates this scenario.

While deployment is within the enterprise (like the previous scenario), the blockchain data itself is not used directly by clients. Rather, there are MongoDB instances that are centrally controlled. These instances send data to and from the blockchain database. Clients access these instances, rather than the blockchain database. Put another way, clients only use blockchain data via intermediaries.

There are three reasons for these intermediate instances: speed, convenience, and privacy. Let's explore each.

Speed: When one writes a transaction to a blockchain database, then the majority of nodes in the federation must agree. This necessarily incurs more inter-node messages than writing to a centralized MongoDB instance, and therefore has a higher delay. Network latency is on the order of 50-100 milliseconds if all nodes are within the same geographic region, but on the order of 1 second if spread across the globe.

Convenience: Developers writing code can iterate faster if they don't have to deal with network latencies on the database, let alone governance around business processes. Enterprises have existing applications that include MongoDB, which often took person-decades to develop. There's usually no need to rebuild those systems from scratch to get the benefits of blockchain technology. It can be far more convenient to start with a fully centralized application using MongoDB instances; and then offload only the data that benefits from a blockchain database.

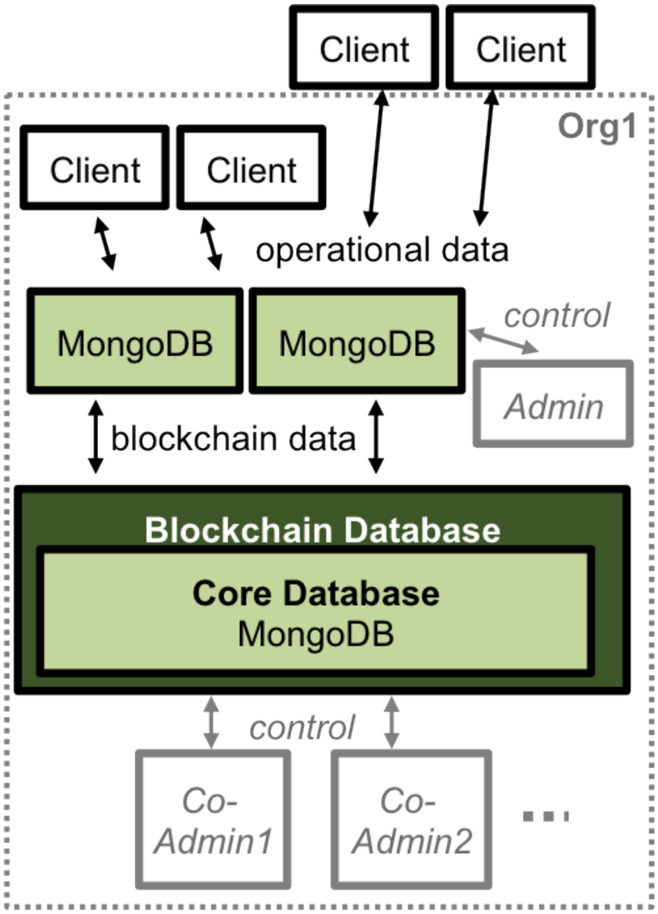


Figure 8: Deployment scenario: blockchain data is not operational; deployed within the enterprise

Privacy: Keeping blockchain data separated from external clients via intermediate MongoDB instances means there are no privacy concerns related to the blockchain database itself. We'll discuss privacy in more detail later.

Example use cases for this scenario are similar to the other "within enterprise" scenario, but where more speed, convenience, or privacy is desired. Examples include managing customer credit status across finance and sales, or customer orders between ecommerce, shipping, and billing systems.

Scenario: Within A Consortium, Blockchain Data is Operational

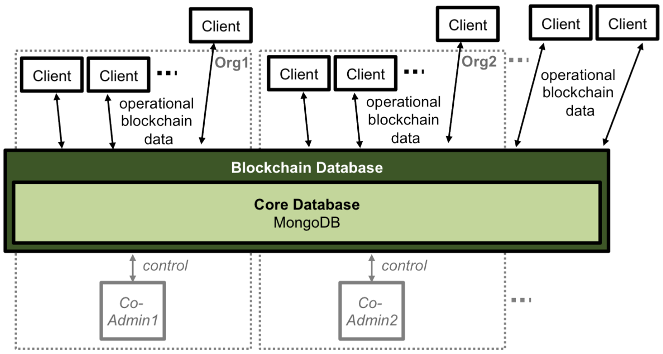


Figure 9: Deployment scenario: blockchain data is operational; deployed within a consortium

Figure 9 shows this scenario. The main objective of creating a consortium is decentralization: no single entity controls the database infrastructure. This gives benefits related to sharing infrastructure, increased immutability due to a single source of truth extending beyond a single organization, and increased tangibility of assets that are claimed or transferred within the system. The technology enables even traditional competitors to work together, because a shared infrastructure can simplify IT landscapes and unlock business opportunities higher in the value chain. A consortium typically forms around an ecosystem, such as financial services, music, or supply chain.

As Figure 9 shows, each organization in the consortium (org1, org2, etc.) has an administrator (co-admin 1, co-admin 2, etc.) that controls one node in the blockchain database (blockchain-enabled MongoDB) deployment. Client users within an organization can directly use that blockchain data. External clients may also access the blockchain data directly.

Blockchain consortiums have many applications. Examples include R3 for finance applications, Open Music Initiative for music applications, or Genesis of Things, led by Innogy SE, for supply chain applications.

Privacy needs rise once you have a consortium. For example:

- Clients who wish to keep their data private, only unlocking read permissions to other entities on an as-needed basis. In fact, for certain financial and personal data related applications, this is a legal requirement for data protection regulations such as the European Union's Data Protection Directive and forthcoming General Data Protection Regulation
- Clients who wish to issue an asset, and have more fine-grained control on how that asset can propagate to other owners. For example, in licensing music.
- The constraints above should apply to co-admins as well. That is, there should be a way for co-admin 1 to commit a client-style write to the database, where co-admin 2 cannot see the data payload.

We've already discussed two approaches to privacy: within-enterprise control of the blockchain database, and to have intermediate centralized MongoDB instances. Each of these helps some use cases, and are straightforward to reason about.

However, there is more broadly applicable approach, which should feel familiar to any MongoDB user: permissioning. As of MongoDB 3.4, only certain clients are given access to defined fields of specific transactions. These capabilities can be leveraged by the blockchain database.

Scenario: Within A Consortium, Blockchain Data is Not Operational

Figure 10 shows this scenario. One would use this scenario for the benefits of a consortium (e.g. shared resources) and for the benefits of non-operational blockchain data (e.g. speed, convenience, privacy) as discussed earlier.

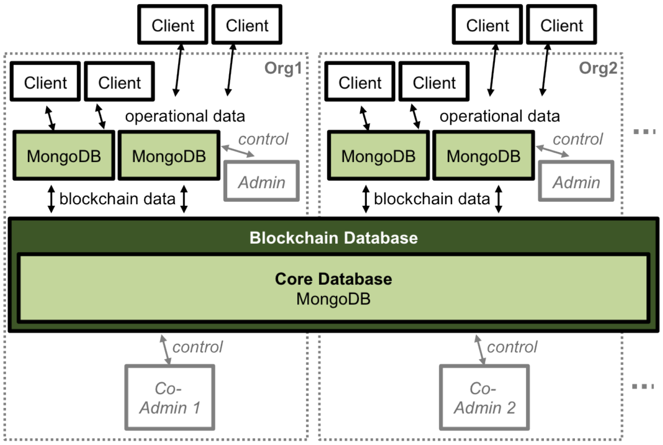


Figure 10: Deployment scenario: blockchain data is not operational; deployed within a consortium

Requirements for the Core Database

We have seen many high-value applications and business processes that can benefit from blockchain databases. The blockchain database software layer implements the core characteristics of blockchain technology: decentralized control, immutability, and assets.

However, for enterprises to innovate and maximize the opportunity presented, they need to carefully evaluate the core database technology, which manages the persistence of data in the blockchain. The contents of the core databases may be blended with other data sources across the enterprise for operational intelligence and single view applications.

The core blockchain database needs to offer the following key attributes:

- **Data integrity.** Overall, transactions need to be strictly ordered. This is an extra challenge in the face of sharding, which is necessary for scale. The blockchain layer and the core database layer must collectively enforce data integrity. As a baseline, this means that each local node of the core database must have a consistent view of its own data; while the top-level blockchain software can manage overall block ordering, prevention of double spends, and other blockchain-specific requirements.
- **Data model flexibility.** Blocks, embedded transactions, and associated metadata can contain many different attributes. For performance and manageability concerns, these attributes should be stored in a single data structure. The structure should be flexible so that new attributes can easily be added as the blockchain application evolves, without compromising data integrity.
- **Fast block verification and operational intelligence.** The database needs to be able to support queries that can validate the lineage of blocks. It also needs to easily service complex queries powering real-time business analytics and reporting dashboards, without first having to move blockchain database contents to data warehouses or data lakes.
- **Highly scalable and continuously available.** While the Bitcoin blockchain is currently around 100GB in size, supporting fewer than 7 transactions per second and multi-minute latency, enterprise blockchain applications serving finance, supply chain, IoT, etc. are much more demanding. In addition to scalability, resilience to failures and planned maintenance is clearly a critical attribute of databases supporting the blockchain.
- **Robust security controls.** The transfer, storage, and querying of value-bearing assets demands extensive security protection in the core database. Access controls with user permissioning and encryption of data are essential.
- **Open.** To assure adoption between different entities, the core database should be based on open source, open standards technologies. By building on open development foundations, cross-enterprise adoption is fuelled by collaboration, transparency, and interoperability.

Meeting Blockchain Database Demands with MongoDB

MongoDB is the fastest growing database in the market today, powering digital transformation initiatives in over 50% of the Fortune 100. Leading companies from the financial services, healthcare, retail, manufacturing, technology, communications, and media sectors use MongoDB, as well as federal and state governmental institutions. MongoDB is the only database that harnesses the innovations of NoSQL – data model flexibility, always-on global deployments, and scalability – while maintaining the foundation of rich query capabilities and robust security that have made relational databases the platform for traditional ledger-based technologies over the past three decades.

MongoDB's distributed architecture makes it an ideal database platform powering the next generation of blockchain databases and applications in the enterprise.

Data Integrity in Blockchain Databases

With blockchain databases recording exchanges of value, the core database needs to enforce integrity of data stored in the blockchain database. MongoDB provides a number of controls to implement the required data integrity guarantees.

Consistency & Transactional Guarantees

Blockchain integrity is predicated on a strict ordering of blocks, with each block referencing the previous block. This ordering provides a complete and transparent audit trail for every record stored in the chain, and prevents committing fraudulent transactions such as double spends. Through strong consistency, MongoDB makes it straightforward to enforce blockchain integrity, ensuring the correct sequencing of blocks as they are committed to the core database, and that one consistent state of the core database is observed by all nodes.

Building on its consistency guarantees, MongoDB also provides ACID guarantees for each block. The block's hash, timestamp, metadata, and transaction identifiers (or transactions themselves, depending on schema design considerations) are stored in a single JSON (JavaScript Object Notation) document, with all fields inserted into the core database in a single operation. If any part of the operation fails, the write is rolled back and can be automatically retried, or notified to the client. As a result of this guarantee, no partial blocks can be inserted into the core database.

Through MongoDB change streams, blockchain users can build reactive, real-time, blockchain apps that can view, filter, and act on data changes as they occur in the database. Change streams enable seamless data movement across decentralized database and application estates, making it simple to stream data changes from the blockchain as new blocks are committed, and trigger actions using a fully reactive programming style. For example, regulatory reporting and analytics applications can be immediately notified when a new transaction is written to the blockchain.

Data Guarantees

MongoDB stores block data in a binary representation called BSON (Binary JSON). The BSON encoding extends the popular JSON representation beyond just strings and numbers to include additional data types such as int, long, date, floating point, and decimal128. By avoiding serializing complex data types into strings, application developers can store data in its native format, reducing code complexity and reducing the potential for errors. For example, MongoDB's `decimal` data type stores decimal values as high precision 16-byte floating point numbers, eliminating rounding errors in operations such as currency conversions or tax calculations.

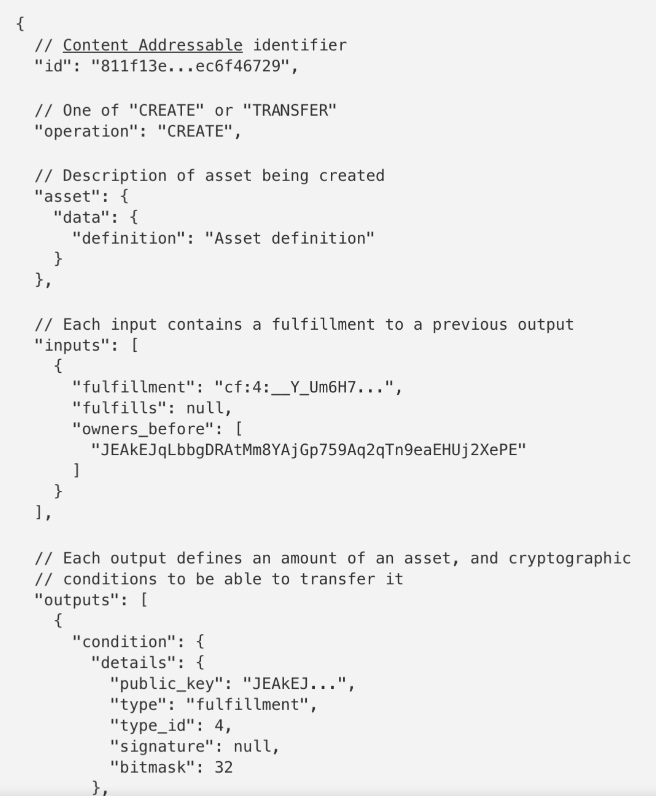
With schema validation, MongoDB allows DBAs to enforce data governance controls against each block and transaction as it is inserted into the core database. Checks can be implemented to enforce proper document structure against specific fields, data types, data ranges, and the presence of mandatory fields. This type of schema control ensures all blocks are properly formed before being presented to the blockchain database.

Flexible Data Model with a Dynamic Schema

With its lightweight format and support for rich data structures, JSON has become the standard data interchange for modern web, mobile, and Internet of Things (IoT) applications, and has been adopted by many blockchain implementations. Blocks themselves are represented as rich data structures, containing both top level attributes that uniquely identify and chain the record, and, depending on data modeling considerations, many embedded transactions. Each transaction, whether stored inside of the block itself, or represented as separate documents, can have varying attributes describing the transaction's inputs, outputs, processing scripts, and payload.

These rich data structures are perfectly matched to MongoDB's BSON documents, which can embed all data contained in the block into a single document, with transaction identifiers, or the transactions themselves, modeled as arrays of sub-documents. All values can be stored as native data types (e.g., strings, doubles, decimals, dates), rather than CLOBs or BLOBs. Any attribute, at whatever level of the document, including sub-documents and arrays, can be indexed and efficiently queried.

Developers can persist blocks directly into the database, without first having to flatten them into the rigid row and column data structures imposed by traditional relational database tables. Maximizing throughput and latency, all of the block's data can be written and retrieved in a single database operation.



```
{
  // Content Addressable identifier
  "id": "811f13e...ec6f46729",

  // One of "CREATE" or "TRANSFER"
  "operation": "CREATE",

  // Description of asset being created
  "asset": {
    "data": {
      "definition": "Asset definition"
    }
  },

  // Each input contains a fulfillment to a previous output
  "inputs": [
    {
      "fulfillment": "cf:4:__Y_Um6H7...",
      "fulfills": null,
      "owners_before": [
        "JEAkEJqLbbgDRAtMm8YAjGp759Aq2qTn9eaEHUj2XePE"
      ]
    }
  ],

  // Each output defines an amount of an asset, and cryptographic
  // conditions to be able to transfer it
  "outputs": [
    {
      "condition": {
        "details": {
          "public_key": "JEAkEJ...",
          "type": "fulfillment",
          "type_id": 4,
          "signature": null,
          "bitmask": 32
        },
```

Figure 11: Rich BSON documents allow complete transactions to be modeled in a single data structure

Further enhancing flexibility, MongoDB documents can vary in structure from one another. For example, all transactions with a consumer may need to store a social security number, while those transactions with a business would have to record a company number, registered tax number, and other unique fields. There is no need to declare the structure of documents to the system – documents themselves are self-describing. If new blocks need to store additional attributes, they can be added to the database without affecting existing blocks. Schema changes do not require updating a central system catalog or taking the core database offline. This flexibility allows blockchain application developers to quickly build and evolve functionality, without the constraints and friction imposed by the rigid schemas of relational databases.

JSON facilitates easy development of higher-level protocols for specific ecosystems as well. This is already happening in the blockchain space: COALA IP was recently developed in a community effort for a blockchain-ready intellectual property protocol. It can be used out-of-the-box on blockchain software with MongoDB providing the core database layer. Several organizations are deploying it to support their blockchain applications.

Rich Queries for Operational Intelligence

Beyond efficiently modeling and persisting blocks, the core database needs to also provide the ability to run rich queries on transactions and blocks. These queries need to be serviced in real time, without the complexity or delay of ETL processes that move data from the core database into a data warehouse or data lake. The expressive MongoDB query language and rich secondary indexes, exposed through the blockchain software, enable developers to build blockchain applications that can query and analyze the core database in any way the business demands. Data can be accessed by single keys, ranges, text search, graph traversals, and geospatial coordinates, returning responses in milliseconds.

The implementation of the blockchain layer itself benefits from these rich query capabilities as well. Consider the process of verifying a transaction or a block. This process involves accessing the data by multiple attributes such as the transaction identifier, input transactions, outputs, block hash, timestamp, source and destination accounts, and more. Each of these attributes needs to be indexed in order to provide the low latency and high throughput verification demanded by enterprise blockchain databases. MongoDB's $graphLookup operator is especially powerful for transaction verification and analytics. It can recursively traverse connected input and output transaction chains across multiple block depths to calculate transitive closure, thereby confirming transaction lineage and chain integrity. By efficiently traversing connected transactions, it is possible to analyze the exchange of assets between buyers and sellers, enabling deeper analytics – for example, to detect potential instances of fraudulent behavior.

With the MongoDB aggregation framework, developers and data engineers can build complex data processing pipelines that enable the business to generate operational intelligence directly from the blockchain database. For example, business users can aggregate trades or sales recorded in the core database over a day, week, or month, calculating sums, averages, and deviations. With MongoDB faceted navigation, results can be conveniently grouped by region and transaction type for comparisons and rapid drill down. Native, idiomatic drivers in over a dozen languages, including Python, R, and Scala, allow data scientists to build sophisticated machine learning models against the data stored in MongoDB. The certified MongoDB Connector for Apache Spark exposes all of Spark's libraries, enabling data from MongoDB to be materialized for further analysis with SQL, streaming, machine learning, and graph APIs. The MongoDB Connector for BI allows business analysts to query and visualize blockchain data in MongoDB with regular SQL syntax from their preferred BI and analytics platforms.

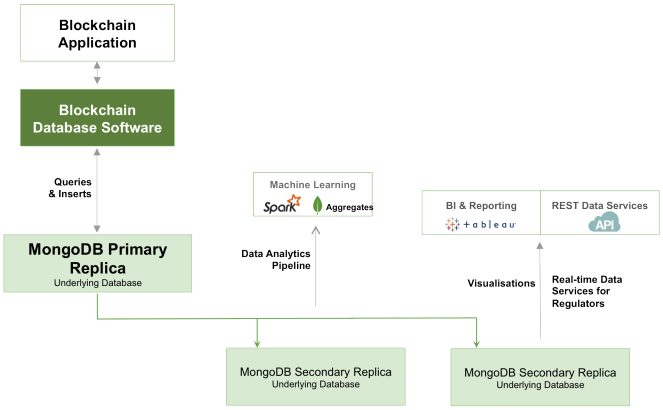


Figure 12: Single blockchain platform converging real-time blockchain and analytic workloads

A MongoDB cluster can be provisioned with dedicated replica nodes serving analytics queries. This allows analysts to simultaneously run exploratory queries and reporting against live data, without affecting the operational blockchain application, and again avoiding lengthy ETL cycles pushing data to external analytics platforms.

Elastic Scalability & Always-On Availability

As enterprises bring more applications onto blockchain platforms, it is essential that the core database can scale to meet demand, while ensuring continuous uptime.

MongoDB provides horizontal scale-out for blockchain applications on low cost, commodity hardware using a technique called sharding, which is transparent to the application. Sharding distributes and replicates data across multiple physical partitions called shards, allowing blockchain deployments to scale beyond a single server, and providing continuous uptime in the face of system outages. Sharding also allows partitions of the blockchain's database to be geographically distributed and controlled by each co-administrator / organization in the federation.

MongoDB uniquely supports multiple sharding policies that give the co-administrators precise control over how blockchain data is distributed across a cluster. As a result, data can be sharded according to application query patterns or regulatory considerations, providing higher scalability over diverse workloads and deployment architectures:

- **Range Sharding.** Documents are partitioned across shards according to the shard key value. Documents with shard key values close to one another are likely to be co-located on the same shard. This approach is well suited for applications that need to optimize range based queries against the core database, such as retrieving all transactions with a specific customer, or over a defined time range.
- **Hash Sharding.** Documents are distributed according to an MD5 hash of the shard key value. This approach guarantees a uniform distribution of writes across shards, but is less optimal for range-based queries.
- **Zone Sharding.** Provides the ability for DBAs and operations teams to define policies governing data placement in a sharded cluster.

When using zones, each shard is part of the same, single cluster and can be queried globally, but data is geographically distributed based on data sovereignty and local access requirements.

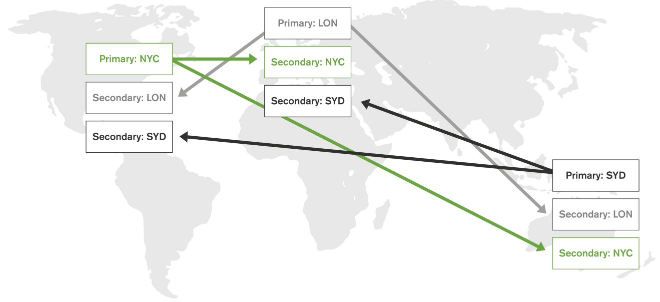


Figure 13: Creating geo-blockchain databases with MongoDB zone sharding

Zone sharding also allows the creation of tiered storage architectures. In this design, the most recently added blocks can be assigned to hosts equipped with high performance SSDs, while aged blocks are stored on hosts configured with less expensive conventional hard disks. By including the block's timestamp in the shard key, the MongoDB cluster balancer can automatically migrate core database data based on age from the high performance tier to the high capacity tier, allowing better cost optimization.

Enterprise-Grade Security for Compliance & Data Protection

With a blockchain database storing an organization's most important digital assets, securing it should be top of mind for co-administrators. While the blockchain application cryptographically signs contents of the blockchain, the core database needs to enforce strict security controls to protect assets from both internal and external threat actors.

MongoDB Enterprise Advanced features extensive capabilities to defend, detect, and control access to data, enabling organizations to meet the demands of regulatory compliance.

- **Access Control**. Enforce access permissions to sensitive data using industry standard mechanisms for authentication and authorization. These of course must get reconciled at the higher level of blockchain database software as well.
- **Auditing.** Enable forensic analysis to track any action against the core database.
- **Encryption.** End-to-end protection of data in motion over the network and at rest in persistent storage.
- **Administrative Controls.** Identify potential exploits faster and reduce their impact.

Authentication

Authentication can be managed from within the core database itself with Challenge/Response credentials or PKI x.509 certificates. MongoDB Enterprise Advanced provides additional integration with external security infrastructure including Kerberos, LDAP, and Active Directory.

Authorization

MongoDB's has a world-class centralized permissioning system that can be used by the decentralized blockchain application. In a centralized setting, MongoDB allows administrators to define permissions for a user or application and control access to data in the core database. With MongoDB you can configure granular, user-defined roles, making it possible to realize a fine-grained separation of duties between different entities accessing and managing the database. Authorization can be managed in MongoDB or via a central LDAP server. Read-only views allow administrators to implement field-level security through the filtering and masking of individual attributes within a transaction.

MongoDB's permissioning capabilities can be used by the higher-level decentralized blockchain application software as well. This manifests in at least a couple ways. First, the co-administrators may configure some admin-style settings but only if there is sufficient collective agreement. Second, read permissions may be treated as assets: person A can send to person B an asset, which is actually a permission, i.e. a permission to read field X in transaction Y. This combines the best of both worlds – the datastore and permissioning capabilities of MongoDB, with the asset capabilities of blockchain technology. This is highly useful in many privacy-related scenarios.

Auditing

Security administrators can use MongoDB's native audit log to track all access and operations taken against the core database, with events written to the console, syslog, or a file for forensic analysis.

Encryption

MongoDB data can be encrypted on the network, on disk, and in backups.

Support for TLS/SSL allows clients and other nodes in a cluster to connect to MongoDB over an encrypted channel. The MongoDB Encrypted storage engine protects data at rest. By natively encrypting database files on disk, administrators eliminate both the management and performance overhead of external disk and filesystem encryption mechanisms.

Managing the Database

Proactive database management and backup is a critical element of any security strategy, enabling administrators to identify and protect against potential exploits before they become expensive breaches. At the same time, organizations want the flexibility to run applications anywhere. MongoDB provides complete platform independence: on-premises, hybrid deployments, or as a fully managed service in the cloud, with the freedom to move between each platform as business requirements change.

MongoDB Atlas is a cloud database service that makes it easy to deploy, secure, operate, and scale MongoDB in the cloud by automating time-consuming administration tasks such as database setup, security implementation, scaling, patching, and more.

When running on your own infrastructure, the most comprehensive solution is provided by the Ops Manager platform, included with MongoDB Enterprise Advanced. Ops Manager provides:

- **Simple configuration and management** with single click database operations, zero-downtime upgrades, and patching.
- **Proactive monitoring** provides visibility into the performance of MongoDB clusters with tracking and alerts on 100+ database and host health metrics.
- **Point-in-time recovery** enabled by continuous backup and consistent snapshots of distributed clusters, allowing seamless recovery of the core database in the event of corruption caused by malicious users or application bugs.

You can learn more about MongoDB's native security protections from the MongoDB Security Reference Architecture guide.

Industry Voices

Bruce Pon, CEO & Co-Founder, BigchainDB

BigchainDB has chosen to build our blockchain database with MongoDB for three reasons:

1. MongoDB serves enterprises extremely well. It has the capability and technology to help large global organizations deploy state-of-the-art database technology.
2. MongoDB is mature and hardened. Through tens of thousands of deployments within leading global companies, we know that MongoDB reliably works in the toughest environments.
3. MongoDB performs. MongoDB has dedicated significant engineering effort to push the boundaries of performance for their database. We can use this to power a blockchain database.

Conclusion

Bitcoin sparked interest in blockchains; that has now grown into a full-blown movement. An increasing number of enterprises across all industry sectors are now exploring how they can use blockchain technology to remove friction from business processes and build systems of trust for value exchange. Blockchain applications, powered by enterprise-grade, scalable and secure core databases such as MongoDB are core to unlocking the potential.

Contact the MongoDB team, and we can schedule a workshop with you to explore how to integrate a blockchain database in your application.

We Can Help

We are the MongoDB experts. Over 4,300 organizations rely on our commercial products, including startups and more than half of the Fortune 100. We offer software and services to make your life easier:

MongoDB Enterprise Advanced is the best way to run MongoDB in your data center. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.

MongoDB Atlas is a database as a service for MongoDB, letting you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. With the click of a button, you can scale up and down when you need to, with no downtime, full security, and high performance.

MongoDB Stitch is a backend as a service (BaaS), giving developers full access to MongoDB, declarative read/write controls, and integration with their choice of services.

MongoDB Cloud Manager is a cloud-based tool that helps you manage MongoDB on your own infrastructure. With automated provisioning, fine-grained monitoring, and continuous backups, you get a full management suite that reduces operational overhead, while maintaining full control over your databases.

MongoDB Professional helps you manage your deployment and keep it running smoothly. It includes support from MongoDB engineers, as well as access to MongoDB Cloud Manager.

Development Support helps you get up and running quickly. It gives you a complete package of software and services for the early stages of your project.

MongoDB Consulting packages get you to production faster, help you tune performance in production, help you scale, and free you up to focus on your next release.

MongoDB Training helps you become a MongoDB expert, from design to operating mission-critical systems at scale. Whether you're a developer, DBA, or architect, we can make you better at MongoDB.

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 Enterprise Download (mongodb.com/download)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Stitch backend as a service (mongodb.com/cloud/stitch)