Cloud Customer Architecture for Blockchain

Executive Overview

Blockchain technology has the potential to radically alter the way enterprises conduct business as well as the way institutions process transactions. Businesses and governments often operate in isolation but with blockchain technology participants can engage in business transactions with customers, suppliers, regulators, potentially spanning across geographical boundaries.

Blockchain technology, at its core, features an immutable distributed ledger, a decentralized network that is cryptographically secured. Blockchain architecture gives participants the ability to share a ledger, through peer to peer replication, which is updated every time a block of transaction(s) is agreed to be committed.

The technology can reduce operational costs and friction, create transaction records that are immutable, and enable transparent ledgers where updates are nearly instantaneous. It may also dramatically change the way workflow and business procedures are designed inside an enterprise and open up new opportunities for innovation and growth.

Blockchain technology can be viewed from a business, legal and technical perspective:

- From a business perspective, blockchain is an exchange network that facilitates transfer of value, assets, or other entities between willing and mutually agreeing participants, ensuring privacy and control of data to stakeholders
- From a legal perspective, blockchain ledger transactions are validated, indisputable transactions, which do not require intermediaries or trusted third-party legal entities.
- From a technical perspective, blockchain is a replicated, distributed ledger of transactions with ledger entries referencing other data stores (for additional information related to ledger transactions). Cryptography is used to ensure that network participants see only the parts of the ledger that are relevant to them, and that transactions are secure, authenticated and verifiable, in the context of permissioned business blockchains.

This document will introduce basic blockchain concepts that define a standard reference architecture that can be used in creating blockchain applications.

Blockchain Fundamentals

A blockchain is a shared ledger distributed across a business network. Business transactions are permanently recorded in append-only ***blocks*** to the ledger. All the consensually confirmed and validated transaction blocks are linked from the genesis block to the most current block with each block linked to its previous block using the cryptographic hash of the previous block - hence the name ***blockchain***.

A blockchain is a historical record of all the transactions that have taken place in the network since the beginning of the blockchain. The blockchain serves as a single source of truth for the network.

High-level View of a Blockchain Network

Figure 1 shows the basic components that comprise a blockchain and its environment. There are many variations on this basic conceptual design that add other features, but the diagram is a useful way to introduce the way that blockchains work.

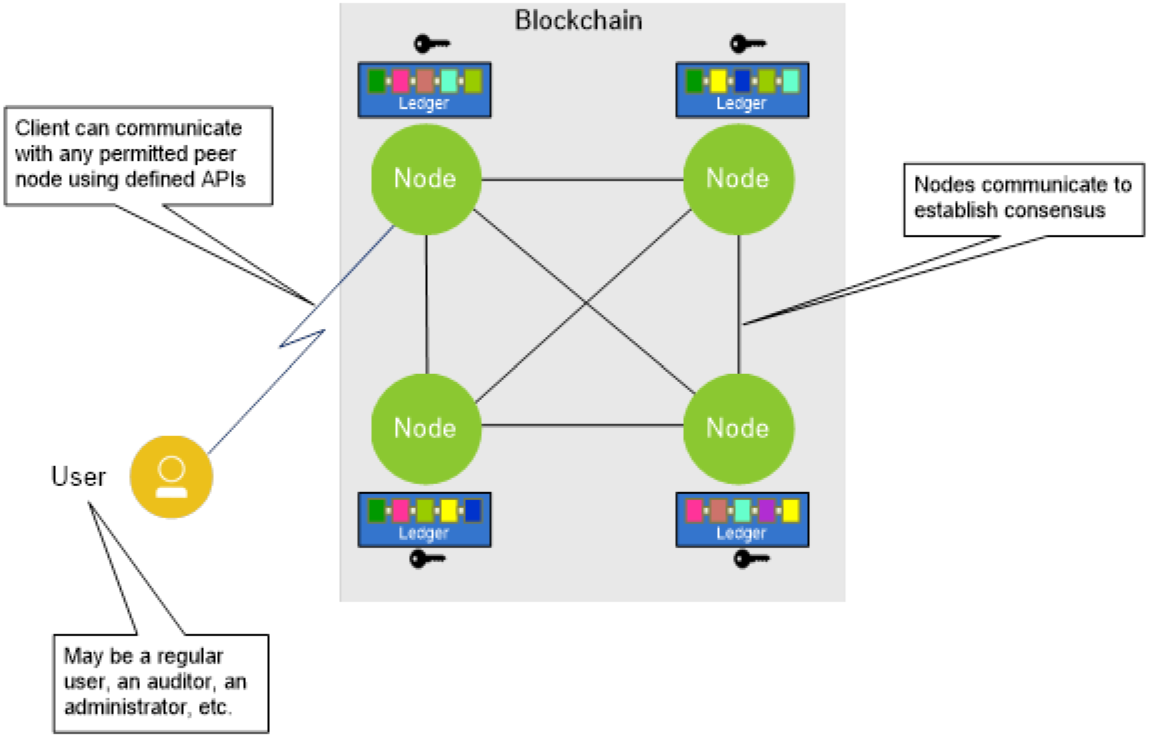

Blockchain
Ledger
Ledger
Node
Node
Node
Node
Ledger
Ledger
Client can communicate with any permitted peer node using defined APIs
Nodes communicate to establish consensus
User
May be a regular user, an auditor, an administrator, etc.

Figure 1: Components of a Generalized Blockchain

In general, a blockchain system consists of a number of ***nodes***, each of which has a local copy of a ***ledger***. In most systems, the nodes belong to different organizations. The nodes communicate with each other in order to gain agreement on the contents of the ledger and do not require a central authority to coordinate and validate transactions.

The process of gaining this agreement is called ***consensus***, and there are a number of different algorithms that have been developed for this purpose. Users send ***transaction*** requests to the blockchain in order to perform the operations the chain is designed to provide. Once a transaction is completed, a record of the transaction is added to one or more of the ledgers and can never be altered or removed. This property of the blockchain is called ***immutability***.

Cryptography is used to secure the blockchain itself and the communications between the elements of the blockchain system. It ensures that the ledger cannot be altered, except by the addition of new transactions. Cryptography provides integrity on messages from users or between nodes and ensures operations are only performed by authorized entities.

The authority to perform transactions on a blockchain can use one of two models, ***permissioned*** or ***permissionless***. In a permissioned blockchain, users must be enrolled in the blockchain before they are allowed to perform transactions. The enrollment process gives the user credentials that are used to identify the user when he or she performs transactions. In a permissionless blockchain, any person can perform transactions, but they are usually restricted from performing operations on any data but their own.

Most business-oriented blockchains include the ability to use ***smart contracts***, sometimes called ***chaincode***. A smart contract is an executable software module that is developed by the blockchain owners, installed into the blockchain itself and enforced when pre-defined rules are met. When a user sends a transaction to the blockchain, it can invoke a smart contract module which performs functions defined by the creator of that module. Smart contracts usually have the ability to read and write to a local data store which is separate from the blockchain itself and can be updated when transactions occur. The business logic contained in a smart contract creates or operates on business data that is contained in this persistent data store.

In a simple blockchain, every node is identical and every copy of the ledger is identical. However, more complex blockchains allow differences in the nodes and the ledgers. Some blockchains support the concept of ***subchains,*** which are sometimes called ***channels***.

Subchains are logically separate chains that occupy the same physical blockchain. Each subchain may be owned by a different entity and may be accessible to a different set of users. Nodes may be set up so that some nodes participate in certain subchains and not in other subchains. The result of this configuration is that the ledger on some nodes will contain transactions for that subchain while the ledgers on other nodes will not. Another variation on the basic blockchain is one in which nodes are assigned specific purposes instead of being identical in their function. This configuration may be used to optimize performance since the system can be faster if every node does not have to perform every operation required for a transaction on the chain.

Key Characteristics of a Blockchain Network

There are several characteristics that apply to Blockchain systems that affect their architecture and implementation:

- **Cryptography:** Blockchain's transactions achieve validity, trust, and finality based on cryptographic proofs and underlying mathematical computations between various trading partners.
- **Immutability:** This term refers to the fact that blockchain transactions cannot be deleted or altered.
- **Provenance:** In a blockchain ledger, provenance is a way to trace the origin of every transaction such that there is no dispute about the origin and sequence of the transactions in the ledger.
- **Decentralized computing infrastructure:** These computing infrastructures feature computing nodes that are capable of making independent processing and computational decisions irrespective of what other peer computing nodes may decide.
- **Distributed transaction-processing platform:** This platform handles a range of transactions, including exchanging value, assets, or other entities.

- **Decentralized database:** Each participating partner has access to a distributed database in its entirety at all times. No single party controls the database, which every party can verify or regenerate if required without having a central intermediary.
- **Shared and distributed accounting ledger:** These ledgers can be public, private, or semi-public/private. Ledgers can be shared amongst participants with privacy. In permissioned blockchains, participants can see the transactions fully with permission and still maintain anonymity. These transactions are final and irreversible since each transaction is linked to every preceding transaction in the ledger. The ledger entries are time ordered and computationally and cryptographically architected to ensure permanence, and the ledger itself is widely replicated.
- **Software development platform:** A software development platform makes use of APIs, peer-to-peer networks, and public, private, or hybrid networks. Transactions are programmable since the underlying ledger is digital in nature, which leads to intelligent and programmable contracts and contract enforcements.
- **Cloud computing:** Blockchain systems frequently involve the use of cloud computing platforms. Cloud computing platforms offer the potential to use large amounts of resources in relation to data storage and also the ability to bring flexible and scalable processing resources to the analysis of data.
- **Peer-to-peer network:** In these networks, participating nodes communicate with each other directly and without a central or intermediate node or entity.
- **Wallet:** A secured data store of access credentials of a user and related information, which includes user IDs, passwords, certificates and encryption keys.

Blockchain network implementations strive for scalability and concurrency, ensure no single point of failure, and include pluggable components like databases and other consensus mechanisms. Successful implementations support multi-level confidentiality and privacy which is achieved through multichannel or subchain communication, multiple sub-ledgers, and multiple stakeholders for transaction visibility based on a need-to-know-basis.

Blockchain Reference Architecture Capabilities

Figure 2 explores the typical capabilities needed for a node or an enterprise participating in the blockchain architecture. The reference architecture is expressed across three networks – public, cloud, and enterprise.

While the location of capabilities in these networks is represented as a best practice, any capability can be implemented in any network according to the needs of the blockchain solutions. While cloud computing is not required to support blockchain platforms, services, or networks, using cloud is recommended because of its elasticity, performance, and networking characteristics.

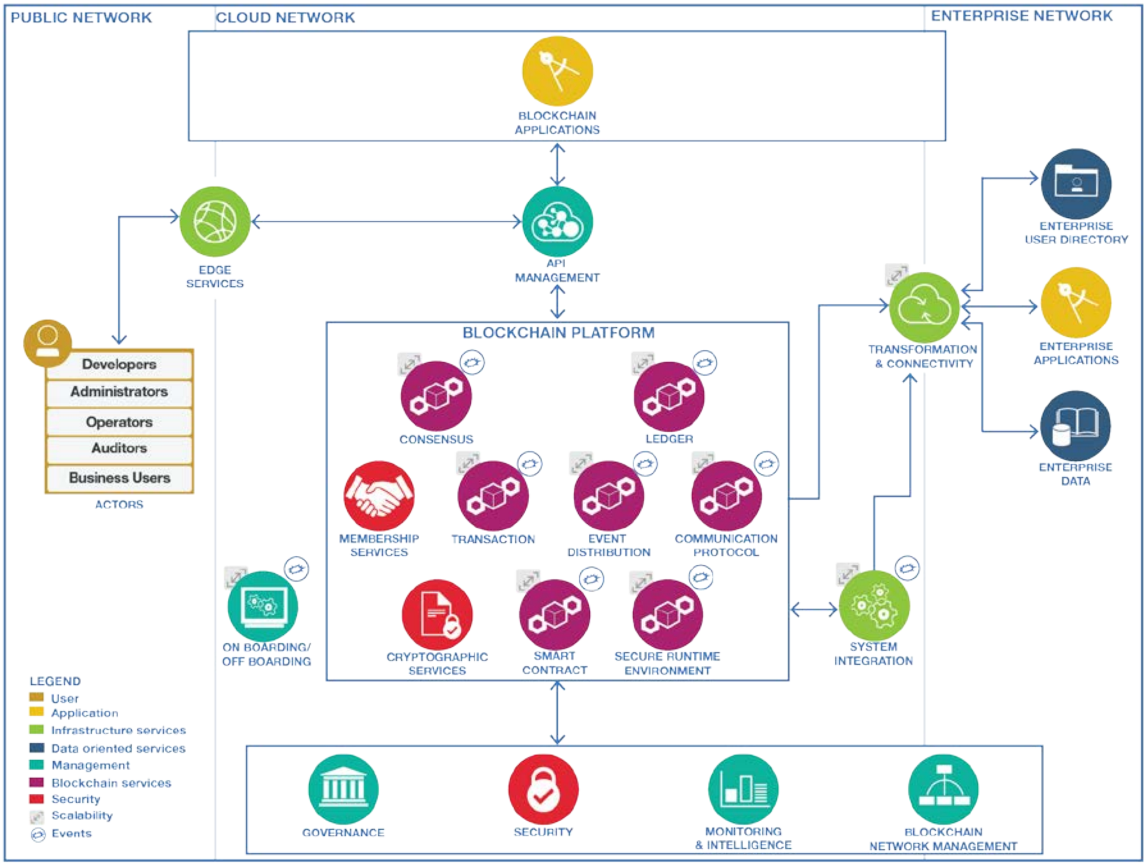


Figure 2: Blockchain Reference Architecture Capabilities

Public Network

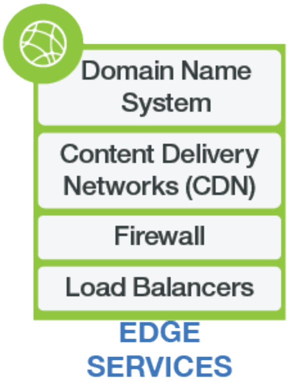
The Public Network contains the wide-area networks (typically the Internet), peer cloud systems, and edge services.

Edge Services

Edge services allow data to flow safely from the Internet into the provider cloud and into the enterprise. Edge services also support end-user applications. Edge services include:

Domain Name System Server (DNSS) - The DNSS resolves the URL for a particular web resource to the TCP-IP address of the system or service that can deliver that resource.

Content Delivery Networks (CDN) - The CDNs support end-user applications by providing geographically distributed systems of servers deployed to minimize the response time for serving resources to geographically distributed users. This ensures that content is highly available and displayed to users with minimum latency. Which servers are engaged will depend on server proximity to the user and where the content is stored or cached.

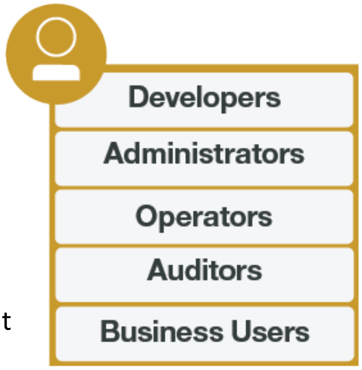


Firewall - The firewall controls communication access to or from a system, permitting only traffic meeting a set of policies to proceed and blocking any traffic that does not meet the policies. Firewalls can be implemented as separate dedicated hardware or as a component in other networking hardware, such as a router, or as integral software to an operating system.

Load Balancers - Load balancers distribute network or application traffic across many resources (such as computers, processors, storage, or network links) to maximize throughput, minimize response time, increase capacity, and increase application reliability. Load balancers can balance loads locally and globally. They should be highly available without a single point of failure. Load balancers are sometimes integrated as part of the provider cloud analytical system components like stream processing, data integration, and repositories.

Users

Users are the parties of a blockchain who create and distribute blockchain applications and perform operations using the blockchain. These actors are consistent with the cloud computing actors and roles from ISO/IEC ISO/IEC 17788. [1]



USERS

Users may include the following:

Developers - Blockchain developers create applications for end users (client side) and develop smart contracts (server side) that interact with the blockchain and are used by blockchain users to initiate transactions. They also write code to enable the blockchain to interact with legacy applications.

Administrators - Blockchain administrators perform administrative activities related to the blockchain network and application such as deployment and configuration of the blockchain network or application.

Operators - Blockchain operators are responsible for defining, creating, managing, and monitoring the blockchain network and application.

Auditors - Blockchain auditors are part of the business network and are responsible for reviewing the blockchain transactions or access control lists and validating the integrity of those transactions from a business, legal, audit and compliance perspective.

Business Users - Business users operate in a business network and interact with the blockchain using an application. It is often the case that business users are not aware of the blockchain.

Now that we've established who is using the technology, let's take a look at how the actors access the blockchain platform and what components make up a blockchain platform.

Cloud Network

Blockchain Applications

Blockchain applications are used to present (business) capabilities to end users of the blockchain system. This is particularly the case for business users, where capabilities need to be presented in terms that relate to the particular application area with concepts and processes familiar to those business users. Applications may also exist to serve other users with different roles including administrators, operators, and auditors.

Blockchain applications can take many forms including web applications (with code centralized on a server closely associated with the blockchain node), or applications running on the end user device(s), potentially connected to server-side application services.

The blockchain applications and services interface with the blockchain platform using the APIs offered by the platform. The applications may have access to other server-side resources such as databases and services, as needed, to implement their capabilities.

Blockchain applications are built to benefit the business networks within specific industries including financial services, healthcare, insurance, energy and utilities, public sector, and retail. Blockchain will also enable cross-industry networks to help revolutionize supply chains, secure and integrate Internet of Things (IoT) applications, and reduce cost and risk.

See Appendix A: Specific Examples of Blockchain Applications.

Application Programming Interface (API) Management

API Management capabilities publish catalogs and update APIs in a wide variety of deployment environments. This enables developers and end users to rapidly assemble solutions through discovery and reuse of existing data, analytics, and services.

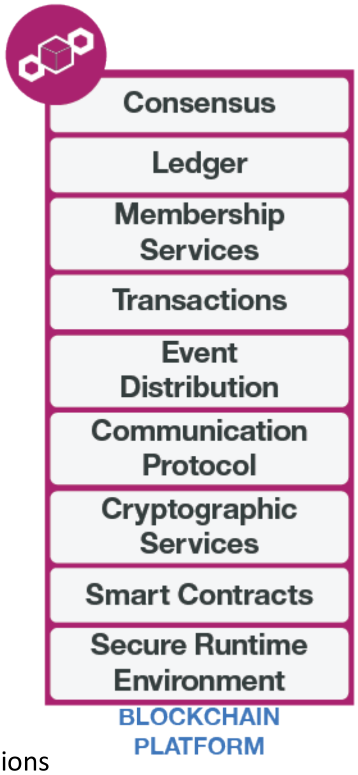
Blockchain applications need to interface with the blockchain network as part of their operation. They achieve this by using the programming interfaces provided by the blockchain. Blockchain offers various APIs that are programming interfaces the applications use to interface with the blockchain platform components to achieve business transaction outcomes.

Refer to the CSCC's *Cloud Customer Architecture for API Management* whitepaper for more detailed information on this subject. [2]

Blockchain Platform

The platform supports essential capabilities for blockchain solutions in a blockchain network node or enterprise. While each blockchain platform is set up and implemented a bit differently, these core capabilities should be considered in blockchain platforms and solutions.

Consensus - Enables a consensus process used by the nodes within the blockchain network to agree on the validity and order of transactions appended to the ledger. The consensus process maintains a consistently replicated ledger within the network.



Ledger - A ledger is a sequence of cryptographically linked blocks that contain transactions.

Membership Services - These services manage identity, privacy, confidentiality, and auditability on the network.

Membership only applies to permissioned blockchains. Permissioned blockchains only allow specific actors to submit transactions or validate the network. In a permissioned blockchain, the actors may be given different roles granting them permission to perform a specific set of operations.

In a non-permissioned blockchain, participation does not require authorization and all actors can equally submit transactions or attempt to accumulate them into acceptable blocks. There are no distinctions of roles.

Transactions - Transactions are records that are appended to the ledger. They can record the exchange of ownership for anything of value such as stocks, bonds, commercial paper, diamonds, etc. Other use cases include changes to medical records or critical device status in an IoT example.

Event Distribution - Events are notifications of significant changes or operations that occur in the blockchain network. For example, events result from execution of a smart contract or the creation of a new block. Events are of interest to participants taking part in the blockchain network.

Event distribution assigns listeners to receive the events from the blockchain. Events will have event producers and event consumers. Producers publish events of interest to the blockchain network, and consumers of events subscribe to events of interest and process the events as they receive them.

In an atomic broadcast, the sender of messages in a blockchain network sends messages to all connected peer members in the same order of sending sequence. This concept is also termed *total-order broadcast* or *consensus* in the context of distributed blockchain network systems.

Communication Protocol - This protocol is the mechanism by which participating computer systems communicate with each other in the blockchain networks. Typically, the participating computer network members use peer-to-peer protocols, such as gRPC, to communicate with each other in blockchain networks.

Cryptographic Services - The Cryptographic Services component provides the blockchain with access to the necessary cryptographic algorithms, either directly or by providing an interface to hardware or software that implements the algorithms. Hash functions and digital signatures are examples of algorithms that are commonly used in blockchains.

Hash functions are often used to protect the ledger from modifications. Any change to information in the ledger will result in a computed hash that is different from the hash that was previously computed and stored for the ledger. A new hash is computed each time a transaction is added to the ledger.

Digital signatures ensure that the receiver receives the transactions without intermediate parties modifying or forging the contents of transactions, while also ensuring that the transactions originated from senders (signed with private keys) and not imposters.

Smart Contract - Smart contracts, sometimes termed *chaincode*, are computer programs that execute in a secure environment within the blockchain platform of any node in the network. Smart contracts encapsulate business logic involving contract terms and conditions between agreeing participants.

The smart contract code determines what transactions are recorded into the blockchain and what information they contain. Smart contracts can be written in a programming language that depends on the blockchain platform.

The smart contract code is stored in the ledger. Transactions can invoke smart contract functions, which can be stateless or stateful, to perform business logic. If required, the code can access external information and systems, via the system integration component (see below).

Smart contracts help to make decisions and automate relationships - all possible outcomes of the contract must be explicitly specified in advance.

Secure Runtime Environment - During runtime, a blockchain transaction may invoke smart contract functions requiring a secure environment. A secure runtime environment is a hosting environment for server-side blockchain business logic.

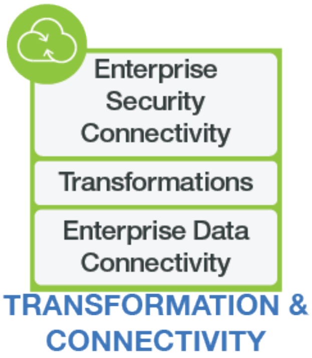
An example is the use of a secure container that contains a set of signed runtime components such as a secure operating system, libraries for blockchain-supported programming languages, their respective runtimes, and the like.

Systems Integration

Typical integration methods include application programming interface (API) adapters and enterprise service bus (ESB) connections between the blockchain platform and the enterprise systems.

Transformation and Connectivity

This capability enables secure connections to enterprise systems and the ability to filter, aggregate, or modify data or its format as it moves between cloud and blockchain components and enterprise systems (typically systems of record).


Enterprise Security Connectivity
Transformations
Enterprise Data Connectivity
TRANSFORMATION & CONNECTIVITY

Within the blockchain reference architecture the transformation and connectivity component sits between the cloud network and enterprise network. However, in a hybrid cloud model these lines might become blurred. The Transformation and Connectivity component includes the following capabilities:

- **Enterprise Secure Connectivity -** integrates with enterprise data security systems to authenticate and authorize access to enterprise systems.
- **Transformation -** transforms data going to and from enterprise systems.
- **Enterprise Data Connectivity -** enables cloud provider components to connect securely to enterprise data. Examples include VPN and gateway tunnels.

Enterprise Network

The enterprise network is comprised of the enterprise user directory, enterprise applications, and enterprise data.

Enterprise User Directory

The enterprise user directory stores user information to support authentication, authorization, or profile data related to the enterprise applications. The transformation and connectivity services use this to control access to the enterprise network, enterprise services, or enterprise-specific cloud provider services.

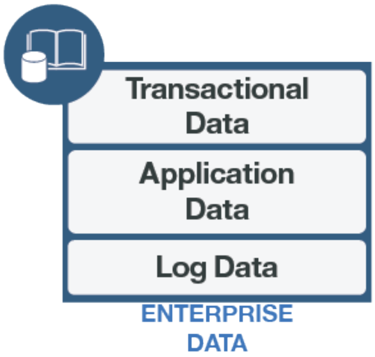
Enterprise Applications

Enterprise applications are created or used by an enterprise that is interacting with the blockchain network. The enterprise applications may interact with the smart contracts on the blockchain. The smart contract may obtain data from the enterprise application, send data to the enterprise application, or request services from the enterprise application.

Enterprise Data

Enterprise data includes metadata as well as systems of record for enterprise applications. Enterprise data may flow directly to data integration or the data repositories providing a feedback loop in the analytical system for blockchain systems. Enterprise data that relates to blockchain includes:

- **Transactional Data** - Data about or from business interactions that adhere to a sequence or related processes (financial or logistical). This data can come from reference data, master data repositories, and distributed data storage.
- **Application Data -** Data used by or produced by enterprise applications functionally or operationally. Typically the data has been improved or augmented to add value and drive insight.
- **Log Data -** Data aggregated from log files for enterprise applications, systems, infrastructure, security, governance, etc.



Blockchain Foundational Services

Governance

The procedures and policies that govern the operation of the blockchain network are known as governance. The network participants agree upon these policies.

Security

Security refers to the security policy and standards that are in place to secure the blockchain platform. Security and privacy in blockchain deployments must address both information technology (IT) security as well as operations technology (OT) security elements. Blockchain protocols architecture relies on public-key cryptography.

Monitoring and Intelligence

These capabilities include monitoring, analytics, and automation tools that are used to respond to changes in the platform and environment. This could include responding to changes in the required system capacity and error analytics.

Blockchain Network Management

The Blockchain network management component provides visibility of the blockchain network operations including business process metrics and performance and capacity data. It also provides a management interface used to change configurations and other parameters.

Context and Options

This section describes a few additional concepts and components that are common in blockchain environments and can be helpful in understanding blockchain solutions and networks.

Permissions options

Permissionless networks are open to any participant and transactions are verified against the pre-existing rules of the network. Any participant can view transactions on the ledger, even if participants are anonymous. Bitcoin is the most familiar example of a permissionless network. [3]

Permissioned networks are limited to participants within a given business network. On permissioned blockchains, participants are allowed to view only the transactions relevant to them and are only allowed to perform operations for which they have permission.

Blockchain storage options

When using blockchain to record transactions, it is typical that only a relatively small amount of the data associated with a particular transaction is stored directly in the blockchain ledger itself. Other data associated with the transaction, which might be much larger, is stored separately from the entry in the blockchain ledger, but is referenced by the entry. This approach is desirable to avoid overwhelming the blockchain ledger with large volumes of data.

As an example, consider an order for goods that is made by a customer to a supplier. The complete order might be a large document that includes a list of order items, quantities, prices, and ancillary information with details of the customer, delivery location(s), and so on. To record the transaction, the blockchain entry might reference an order number, a customer identity, and the total cost, plus a security token (such as a hash) relating to the complete order document. The complete order document itself is stored separately in an order database. Other examples include the storage and administration of personal health records.

Ledger Storage

Ledger storage is the physical storage where the transaction data in the blockchain ledger is stored.

Data Storage

Data storage supports data other than the blockchain ledger itself and can take many forms depending on the nature of the data. It can take the form of a database – either an SQL database (particularly for record-oriented data) or a NoSQL database (particularly for document-oriented data). It can also take the form of an object storage service or a block storage service.

The choice of which form of data storage is used depends primarily on the nature of the data objects themselves and the operations that need to be performed on them. In some cases, multiple different forms of data storage might be used.

Separately, the matter of replication and backup of the data objects needs to be considered. For some data storage services, replication and backup are built into the service itself. In other cases, it is necessary to create replicas or backups through deliberate actions of the application.

Interaction Options

There are several ways that users can interact with the blockchain. Examples include command line interfaces, client software development kits, or software development kits.

Command Line Interface (CLI)

Blockchain developers and administrators sometimes have a need to interface with the blockchain network and perform various activities such as import, export, manage accounts, monitoring, and the like using a simple text command structure. The administrators may use a Command Line Interface to achieve these.

Client SDK

The Client SDK is a client-side programming library that features a set of APIs in the form of "methods" or "calls" which can be used by client programs to access the capabilities and functionalities of the blockchain network. The client programs can be written in Java, Node, Python, or other supported languages. The SDK may also include development tools.

Software Development Kit (SDK)

SDKs for blockchain support communication and integration between blockchain applications and a blockchain platform. Blockchain application development involves developing, testing, debugging, and deploying the application.

The SDK is a set of tools that enables blockchain application developers to create and deploy the applications that interact with the blockchain network during the software development life cycle.

Runtime Flow

The technical implementations of blockchains vary depending on the type of blockchain that is chosen. They can use different methods to establish the network topology, manage participation, execute smart contracts, and manage growth. The Linux Foundation is creating an open standard for blockchain called Hyperledger. [4]

The runtime flow below describes a supply chain scenario that uses the Hyperledger Fabric blockchain implementation. The scenario illustrates the movement/import of goods from an exporter from the port of origin to the port of destination via a selected transporter. The transaction process is initiated by the importer and completed when the imported goods are cleared by custom officials.

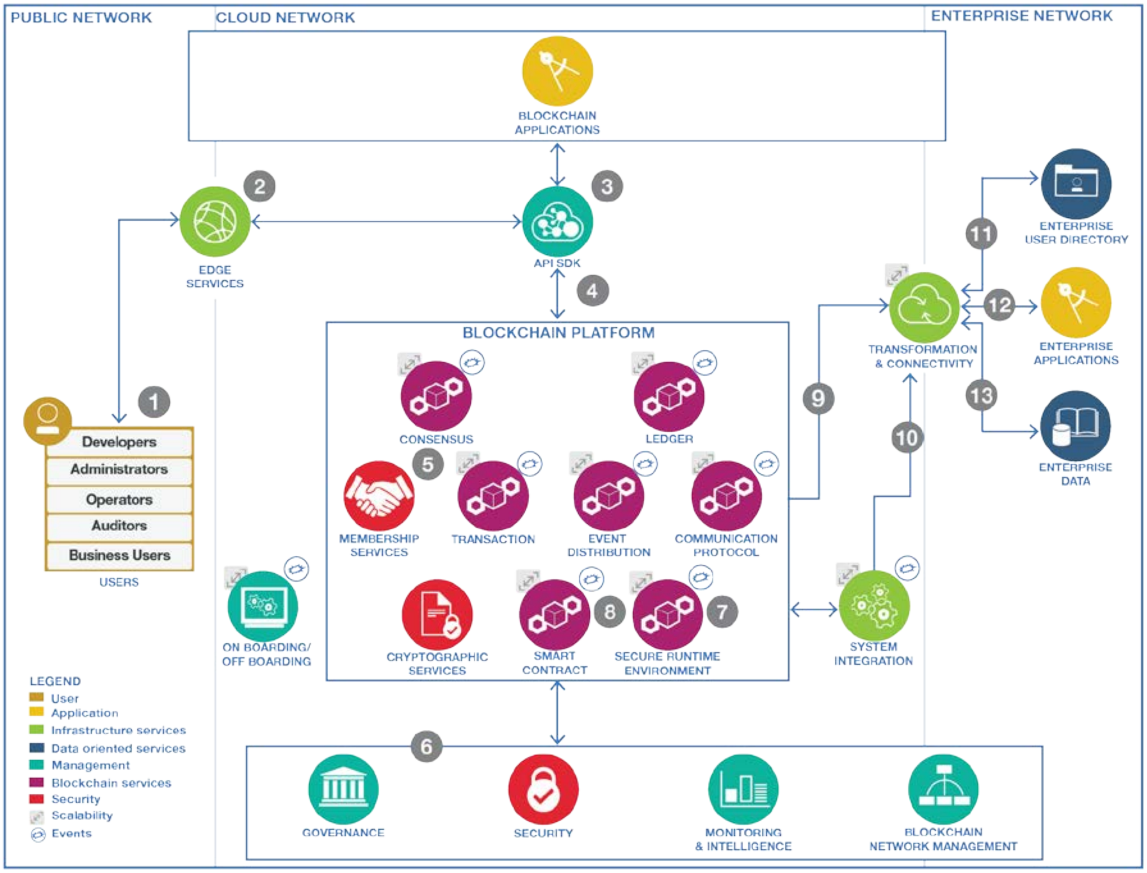


Figure 3: Runtime Flow

1. A blockchain user accesses the blockchain platform to perform a transaction:
   a. A Developer develops the blockchain application and deploys the application using an SDK and commands on a command line interface.
   b. An Administrator sets up the blockchain network.
   c. An Operator manages the day to day operation of the blockchain network.
   d. An Auditor performs business related audits.
   e. A Business user may use a browser client to enable interaction. The various business users in this scenario include:
      - Importer
        i. *Importer* requests a Letter of Credit from its bank.
        ii. Following the receipt of the goods, the Importer lets the bank know of the receipt.
      - Importer's Bank
        i. Creates a Letter of Credit on behalf of the *Importer* through the bank's legacy system and publishes it on the blockchain through the System Integration component.
        ii. Upon receipt of the goods by the *Importer*, releases the payment to the *Exporter's Bank.*
      - Exporter's Bank
        i. Receives the Letter of Credit, publishes the receipt on the blockchain and notifies the *Exporter.*

    ii. Upon receipt of the funds from the *Importer's Bank,* it transfers the money to the *Exporter.*

- Exporter
    i. *Exporter* readies the goods for shipment and requests *Export Control Authority* for clearance.
- Export Control Authority
    i. Upon notification that *Exporter* is ready for shipping, the Export Control Authority performs necessary inspection on the goods and publishes the export certificate on the blockchain.
    ii. The Export Control Authority notifies the *Shipping Company.*
- Shipping Company
    i. *Shipping Company* creates a Bill of Lading and publishes it on the blockchain.
- Departing Port Authority
    i. *Departing Port Authority* confirms the shipment by publishing the shipment on the blockchain.
- Arrival Port Authority
    i. *Arrival Port Authority* confirms the arrival of the shipment by publishing it on the blockchain.
- Customs Authority
    i. *Customs Authority* inspects the goods and if passed publishes the customs clearance on the blockchain.

Business users are likely to access the blockchain network through blockchain applications running on the cloud network. Other users may access the blockchain network from within the environment for administrative and operational purposes.

2. Upon receiving the request to access the blockchain network, edge services route the request to the security gateway.
3. Business users may access the application using a browser client that passes the request to the blockchain platform.
4. The request passes through the API SDK to the blockchain platform for authentication, authorization and execution.
5. Hyperledger Fabric Membership Services, which provides security, privacy and protection for blockchain users, authenticates the user and authorizes the action requested by the user based on their role.
6. Security services enable the Hyperledger Fabric Membership Services to establish the user's identity and provide authentication, authorization and integration capabilities.
7. A secure environment is spawned to execute server-side blockchain business logic.
8. Based on various triggers, smart contract business logic enforces contract terms between the stakeholders.
9. The validation of the smart contract may necessitate access to enterprise data. The request goes through the Transformation and Connectivity service which transforms the data and ensures secure and reliable delivery.
10. In addition, external events may trigger a blockchain transaction such as the creation of the Letter of Credit by the *Importer's Bank* through its legacy systems.

11. Before any processing is performed by the Enterprise Application, the blockchain service instance may authenticate the user via the Enterprise User Directory.
12. The Enterprise Application may leverage data used by the client app as well as logs and context data for analytics. If the client application updates the data then the Enterprise Application may process those changes.
    a. For example, an Enterprise Application is used by the *Importer's Bank* to approve a fund transfer through their legacy system that subsequently sends a message through an *interbank financial network.*
13. Enterprise Data is the data store for the Enterprise Application.

Cloud Deployment Considerations

There are several considerations that need to be evaluated when deploying a blockchain solution.

Scalability and Elasticity

In a blockchain architecture, the number of transactions can be very large. Transformation and connectivity needs to provide scalable messaging and scalable transformation of data in the cloud for these data flows. Elasticity is the ability for a cloud solution to provision and de-provision computing resources on demand as workloads change.

Public clouds have a distinct advantage since they generally have larger pools of resources available. Customers also benefit by only paying for what you use. Private clouds and dedicated hardware can make up some of the difference with higher bandwidth data paths.

Data Bandwidth

Public and private clouds need to be optimized for handling large data sets. Large cloud data sets requiring fast access benefit from processing components with fast and efficient data access. In many cases, this means moving the processing to the data, or vice versa. Cloud systems can effectively hide the physical location of data and processing. Tuning activities can be carried out continuously with minimal impact on deployed applications.

Data Sovereignty

The physical location in which data is stored may be regulated, with the regulations varying from country to country. This is particularly the case for personally identifiable information (PII) and for sensitive data such as health data and financial records. The European Union has particularly stringent regulations that apply to the PII of European citizens.

As a result, any blockchain system must take into account data sovereignty rules and store and process data only in those locations permitted by the regulations – this requires that the cloud service provider allow the cloud service customer to control storage and processing locations, as well as backup locations.

Refer to the CSCC's *Data Residency Challenges* whitepaper for more detailed information on this subject. [5]

Resilience

In blockchain systems, resilience and fault tolerance are very important. Blockchain systems should not depend on one single component at any point and should tolerate the failure of a single component.

Components in the provider cloud should be made resilient through the use of multiple instances of programs and cloud services allied with data replication and redundancy on multiple storage systems.

The networks should also be resilient, for example with multiple paths and multiple providers in the public network. There is no silver bullet to make the entire network available all the time but it should be highly available and resilient. It is important to ensure that the connectivity capabilities can support resilience.

Security

As more data about people, financial transactions, and operational decisions are collected, refined, and stored, the challenges related to information governance and security increase. Data privacy and identity management are incredibly important. The cloud generally allows for faster deployment of new compliance and monitoring tools that encourage agile policy and compliance frameworks. Tools that monitor activity and data access can actually make cloud systems more secure than standalone systems. Hybrid systems offer unique application governance features: Software can be centrally maintained in a distributed environment with data stored in-house to meet jurisdictional policies.

Enterprise blockchain provides a design avenue where transaction data, value, and state are inherently close to the business logic. A secure community process validates the security of the execution of business transactions, enabling a foundation of trust and the robust processing of transactions.

Appendix A: Specific Examples of Blockchain Applications

There are many use cases for blockchain applications. Some well-understood examples include:

- **Letter of credit (LOC)**: Blockchain provides a common ledger for letters of credit that allows banks and all counter-parties to have the same validated record of transactions and fulfillment of conditions. Smart contracts, also referred to as chaincode; define the conditions for payment and LOC fulfillment.
- **Trusted supply chain**: Blockchain provides an agreed-upon, shared record of the asset information – as it transverses the supply chain - recording who owns what, as well as when and where an asset is in the supply chain.
- **Healthcare payments**: Clinical attachments in healthcare claims involve a lot of actors, information, and records, where successfully associating information is a complex task. The matching of clinical versus payment data involves complex information requirements and can be very costly and time consuming. Blockchain can simplify this complicated relationship and automate the information collection and sharing.
- **IoT to Economy of Things**: Blockchain functions as a distributed transaction ledger for IoT transactions, helping to enable autonomous device coordination and peer-to-peer messaging.
- **Commercial paper**: Blockchain connects corporate treasuries with global advisors for investment advice, with subsequent execution through to clearing and settlement.
- **Contract management**: Blockchain is used as a shared repository of the legal documents and their approval histories. The business process or workflow of document handling is enabled by chaincode. When there are multiple related documents (such as a master service agreement, service contract, invoice, etc.), chaincode automatically checks the consistency of these documents to reduce errors.
- **Manufacturing provenance**: Blockchain holds complete provenance details of each component part, accessible by each manufacturer in the production process, such as aircraft owners, maintainers, and government regulators.
- **Vehicle registration and maintenance**: Government regulators can create a vehicle template on blockchain, which is updated by the manufacturer on transfer to the dealer, initial owner and future owners. This could be augmented with other information such as records of maintenance or vehicle theft. Use blockchain as a shared ledger of vehicle history, detailing usage, maintenance, warranty work, and replacement parts.
- **Livestock registry**: Blockchain provides a common ledger for the government, farmers, slaughterhouses, and cattle markets. All parties have access to validated records of livestock, including transactions and movements. Smart contracts define the conditions of transactions/movements according to animal type.
- **Equipment records**: Use blockchain as a shared ledger of equipment history, detailing usage, maintenance, and warranty work and replacement parts.
- **Food safety provenance**: Secure documentation is added to raw material and consolidated onto packaging with aggregation for shipment. Blockchain provides real-time visibility of the food supply chain to distributors, consumers, retail buyer, auditors, and regulators.

References

[1] ISO/IEC 17788. https://www.iso.org/standard/60544.html

[2] Cloud Standards Customer Council 2017, Cloud Customer Architecture for API Management. http://www.cloud-council.org/deliverables/cloud-customer-architecture-for-api-management.htm

[3] Bitcoin. https://bitcoin.org/en/

[4] Hyperledger. https://www.hyperledger.org/

[5] Cloud Standards Customer Council 2017, Data Residency Challenges. http://www.cloud-council.org/deliverables/data-residency-challenges.htm

Acknowledgements

The major contributors to this whitepaper are: Todd Arnold (IBM), Nitin Gaur (IBM), Subodh Manicka (IBM) and Bhargav Perepa (IBM).