A Comparative Study of Permissioned Blockchain Platforms

Ravi Kanth Kotha[1,2], N. V. Narendra Kumar[1], and T. Ramakrishnudu[2]

[1] IDRBT, Hyderabad
[2] NIT, Warangal
ravikanth1027@gmail.com
naren.nelabhotla@gmail.com
trk@nitw.ac.in

Abstract. "Blockchain" is an append only data structure that provides high guarantees on integrity of the data stored in it. This feature of blockchain is particularly useful in distributed systems where multiple entities are responsible for the generation and use of data. It is a common practice to use the term "blockchain" to also refer to the complete ecosystem that enables distributed applications. Permissioned blockchain platforms restrict the permissible actions of the participants according to a policy. Several permissioned blockchain platforms with very different philosophies exist making it difficult for users to choose a platform that suits their application requirements. In this paper, we (i) describe a few important requirements of entities participating in a distributed system, (ii) discuss the gaps in current centralized systems, and (iii) compare four prominent permissioned blockchain platforms in terms of how well they satisfy the identified requirements.

Keywords: Blockchain technology, Security and Privacy, Platform comparison.

1 Introduction

Blockchain by definition is an append only linked list data structure of blocks each containing a set of transactions. Each block is connected to its previous block by storing the hash of its previous block making a chain kind of structure [6]. This type of data structure could be useful in a distributed environment where multiple entities generate data independently and still maintain the integrity of data. Bitcoin [5], has leverged the blockchain data structure to provide decentralized offline digital cash. In distributed systems one of the important aspect is to make the entities arrive at consensus on the list of transactions being added to the blockchain. This has reinvigorated research on distributed consensus models.

Bitcoin is a cryptocurrency, a form of digital cash is the first application of blockchain technology. In bitcoin, digital currency can be transfered without any central administrator from user to user. Though bitcoin is the first application

to leverage the blockchain data structure, its applicability was confined only to digital currency. Ethereum introduced the notion of smart contracts which in conjuction with blockchain techology forms a powerful distributed computing paradigm. Using Ethereum smart contracts a varity of applications can be developed in various domains such as finance, real estate, suply chain systems etc.

Bitcoin and Ethereum are based on public permissionless architecture where the peers do not require any permissions to take part in a transaction. This makes all the users / peers of the network to view all the data floating in the system. As the privacy of data is at risk in these systems bitcoin / ethereum were not useful for organizational purposes. Therefore a new class of platforms have evolved called permissioned blockchain platforms. Hyperledger fabric, Corda, Multichain and Qourum are some of the permissioned blockchain platforms where the roles and responsiblities of peers are clearly defined including their access to data. The notion of blockchain has been interpreted in many different ways which reflects in the design and working of these platforms. Not having a common basis hampers the ability of the organizations to comapre and choose an appropriate platform that suites their requirements.

The current literature on blockchain compares the platforms based on the features of the blockchain like usability, scalability, development and documentation [4]. Our objective is to establish a set of important parameters that could be used as a basis for evaluating blockchain platforms and choosing a platform suitable for a given set of requirements. The important contributions of this paper are (i) identification of technical requirements for business networks, (ii) highlight the gaps in current central systems with respect to the identified technical requirements, (iii) study of prominent permissioned blockchain platforms with respect to the requirements, and (iv) identification of application characteritics for which each of these is best suited.

The rest of the paper is organised as follows, identifying the technical requirements of business networks is discussed in Section 2 and the characteristics of current central systems with respect to the identified requirements are explained in Section 3. A detailed discussion of blockchain technology and its benefits are discussed in Section 4 and four prominent permissioned blockchain platforms are discussed in detail in Section 5. In Section 6, the above discussed platforms are compared against the requirements and followed by conclusions in Section 7.

2 Technical Requirements of Business Networks

Information and communication technology has been an enabler for business, and embracing the rapid advances in the field has become imperative to remain competitive. Particularly, in the case of transactions involving multiple businesses, there are some important requirements that the underlying technology is expected to fulfill. In this section, we highlight the requirements that we consider are crucial for B2B systems.

2.1 Guarantees on message delivery

A trusted and reliable messaging framework is at the heart of any digitally enabled business. The communications technology needs to guarantee that all the messages will be delivered to the intended recipients.

2.2 Guarantees on results of business processes

Though the main purpose of business networks is to enable exchange of information, it is equally important that the information exchanged is acted upon by both the parties in an agreed manner. When the records of the participants differ from one another, reconciliation becomes necessary which is a resource intensive and hence expensive process.

2.3 Business continuity

Despite the advancements, systems are prone to errors either due to technical reasons or for reasons beyond our control. Businesses need the ability to run smoothly in spite of the system failures. Fault tolerance and recovery are important requirements for business continuity.

2.4 End-to-end security

Complex business processes require interactions among multiple participants, where information is processed and shared from one party to another in multiple hops. In such a scenario, it is important that all the participants are accessing and acting on the same information, and have the same understanding about the state (success / failure / current status) of the transaction. This requirement is referred to as end-to-end security.

2.5 Privacy

For successful conduct of business it is important that information is available to all the participants that need to act on it. At the same time it is equally important that the privacy is preserved i. e. sensitive business information be not available to participants that do not need it.

2.6 Data integrity

Data integrity is the assurance that information is trustworthy and not corrupted. Maintaining the integrity of data is absolutely crucial for businesses since it is as an invaluable tool for decision making that directly impacts the sustenance.

2.7 Latency

Latency is the time taken to complete a transaction. To efficiently tackle the growing volumes of transactions, businesses need systems that provide a very low latency.

2.8 Asynchronous processing

Ideally, businesses should allow users the convenience to transact simultaneously through multiple channels, and have the capability to correctly handle all the transactions in a seamless manner. Such asynchronous processing also enables higher throughputs.

3 Characteristics of Current Central Systems

Currently, business networks have a hierarchical topology and are formed by establishsing messaging platforms that enable information transfer. There is a single trusted party through which all transactions are routed. To understand this better, let us understand how a simple fund transfer works.

Steps for funds transfer:

1. Sender initiates the transaction to send money to receiver by sending the request to his/her bank.
2. Sender bank on receiving the request transfers the request to the central bank after verifying the necessary conditions.
3. The central bank upon receiving the request from senders bank, executes the request and sends the response to the sender bank and also to the receivers bank.
4. Receivers bank on receiving the message from central bank updates the account of the receiver, to reflect the current updated balance.

Most traditional business networks follow a similar approach. Though these systems have evolved over time, there are certain concerns that need to be addressed.

- Single point of Failure
- Fault Tolerance
- Reconciliation Overhead

Single point of Failure
A single point of failure (SPOF) is a potential risk posed by a flaw in the design, implementation or configuration of a circuit or system in which one fault or malfunction causes an entire system to stop operating. In a data center environment, a single point of failure can compromise the availability of the system or the entire data center depending on the location and interdependencies involved in the failure. Adding redundancy to such systems could solve the problem to

an extent, however introduces the challenge of synchronizing the data between replicates.

Fault Tolerance
Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of some (or one or more faults within) of its components. If the operating quality for a system decreases, the decrease will be proportional to the severity of the failure. Fault tolerance is particularly sought after in high-availability or life-critical systems.

Reconciliation Overhead
Reconciliation is an accounting process that uses two sets of records to ensure figures are correct and in agreement. It confirms whether the money leaving an account matches the amount that's been spent, and making sure the two are balanced at the end of the recording period. The purpose of reconciliation is to provide consistency and accuracy in financial accounts.

In summary, though the currently used central systems address the business requirements to a large extent there are improvements possible. In particular, our study finds that central systems can be enhanced using the available technologies to alleviate their shortcomings in (i) guarantees on business processes i.e. reconciliation efforts, and (ii) business continutity by adapting a replication strategy which increases redundancy with appropriate mechanisms for ensuring consistency of data. However, both these efforts introduce processing overheads which impact the throughput. Further, we find that end-to-end security (as defined in Section 2.4) is difficult to achieve in current systems through simple enhancements.

4 Blockchain Technology and Benefits

A ledger by definition it is a book of record keeping all the financial transactions of the organization. Since ancient times, ledgers have been at the heart of economic transactions to record contracts, payments, buy-sell deals or movement of assets or property. In modern days computerized ledger came into existence and the general ledger works as a central repository for accounting data transferred from all sub ledgers cash management, fixed assets, purchasing and projects. The general ledger is the backbone of any accounting system which holds financial and non-financial data for an organization. The collection of all accounts is known as the general ledger. In a manual or non-computerized system this may be a large book.

A distributed ledger can be defined as a database that is consensually shared and synchronized across network spread across multiple sites, institutions or geographies. Blockchain is a kind of digital ledger. In the following two sections we will discuss the characteristics and advantages of blockchain technology.

4.1 Overview of Blockchain Technology

Blockchain is a shared and distributed ledger that keeps a record of the transactions. Blockchain data structure is a chain of blocks where each block contains a set of valid transactions and is connected to its preceding block using the concepts of cryptography.

Blockchain technology is useful in distributed systems where multiple entities transact with each other, and wish to maintain a verifiable and trust-worthy record of all the transactions without the need for a trusted central party.

In general a transaction workflow in a blockchain technology starts with the submission of a transaction to the network by an entity of the system. The submission of a transaction could be initiated by a human or a smart contract - a computer encoding of the business logic. The transaction is validated against the current state of the system (denoted by the chain of blocks) and included in a new block. The new block is added to the chain of blocks after consensus is reached.

Blockchain technology is realized by a clever combination of the following components: (i) cryptography, (ii) peer-to-peer networking, (iii) distributed consensus, and (iv) fault-tolerant computing. There are broadly two varieties of blockchains: permissionless and permissioned.

In permissionless blockchains any entity could join the network, submit transactions, validate transactions, create blocks, participate in the consensus and access the full information from the blockchain. Examples of permissionless blockchains are Bitcoin and Ethereum. Since all the data can be viewed by all the entities, privacy of data is at risk in permissionless systems. Since the nodes participating in consensus is not constrained by the protocol, consensus is only an emergent property, thus making settlement finalty probabilistic. A further disadvantage of this design is that the throughput (transactions processed per unit time) achieved is very low, which in turn leads to poor scalability and high latency. Hence, permissionless blockchains are not suitable for business usecases.

In contrast, in permissioned blockchains entities have clearly designated roles in the systems. Some of the important roles are end users, validators and approvers. End users can only submit transactions and access their transactions from the blockchain. Validatiors can only validate transactions and access their transactions from the blockchain. Approvers can only create blocks (update blockchain), and participate in the consensus process. Examples of permissioned blockchains include Hyperledger Fabric, Multichain, Quorum and Corda. Because of the designated roles to participants, and the permissioning of data access, the desired level of privacy is preserved in permissioned blockchains. An update to the state of the system is made known to all the participants only after consensus is achieved among the approvers. Thus settlement finality is achieved and at any time there is only one chain of accepted blocks. Permissioned blockchains achieve higher throughput than permissionless blockchains, but take higher processing time than centralized systems. However, this tradeoff is inevitable for achieving a higher degree of fault-tolerance.

4.2 Advantages of Blockchain Technology

In this section, we shall discuss the benefits of blockchain technology.

1. Immutability
2. Transparency
3. Business Continuity
4. Privacy
5. Disintermediation
6. Consensus
7. Trust
8. Smart Contracts

Immutability: Immutability can be defined as the property of an object of not being able to change its structure due to changes in its environment or external factors. It is impossible to achieve true immutablity in digital systems. In the jargon of blockhcina technology, immutability is used to mean that it is computationally infeasible to corrupt the data stored on the blockchain. This is achieved by the use of hash functions, and by including the hash of a block it its succesor block. By the nature of hash functions, blockchain makes its easy to verify and very difficult to modify the data stored in it, thus making it almost immutable.

Transparency: Transparency can be defined as a property where all the parties involved in a transaction know exactly what actions are being taken on what data. In blockchain, all valid transactions are added to the ledger which is made available to all the parties based on the configuration thus providing the necesary transaperancy.

Business Continuity: Businesses have a lot at stake based on the service provided to their clients. This makes the availability of services as one of the top requirements for any business. Despite failure of some components, the system is expected to support continuity of business as usual. Permissionless blockchains provide business continutity because several copies of the data is distributed at multiple geographies.

Privacy: Preserving the privacy of data shared between the parties of a transaction is a critical requirement. In permissionless blockchains all the data is shared to all the members of a network. However the identity of the entities is psuedonymous. Thus indirectly they provide limited privacy, but are prone to linking attacks. In contrast, in permissioned blockchains, because the data share only on a need-to-know basis, privacy is preserved better.

Disintermediation: Disintermediation means replacement of intermediaries by technology components. Permissionless blockchains being truly decentralized enable a higher degree of disintermediation. However, this comes at the cost of loss

of scalability and performance.

Consensus: The distributed consensus problem requires agreement among a number of entities for a single data value. Some of the entities may fail or be unreliable in other ways, so consensus protocols must be fault tolerant or resilient. In all the blockchain systems, achieving consensus on the state of the system is a crucial aspect and is realized through a variety of approaches. Permissionless blockchains use game theoretic strategies based on economic incentives, while permissioned blockchains use more traditional consensus strategies such as Paxos, PBFT etc.

Trust: The problem blockchain is trying to solve is how to establish a trustworthy record among mistrusting entities. The protocols embedded in blockchains and the use of cryptographic components are designed in such a way that trust is enforced and easily verified.

Smart Contracts: A smart contract is a computer code intended to digitally facilitate, verify, and enforce the negotiation or performance of a business logic. Smart contracts automate the performance of credible transactions without third parties. Though smart contracts can also be integrated with traditional systems, the guarantees of integrity of data in the blockchain together with the fact that all the parties see the same data makes blockchain platforms particularly amenable to leveraging smart contracts.

5 Working of Current Permissioned Blockchain Platforms

In this section, we discuss four prominent permissioned blockchain platforms including the roles supported and transaction flow:

1. Hyperledger Fabric
2. Corda
3. Quorum
4. Multichain

5.1 Hyperledger Fabric

Hyperledger Fabric is a permission distributed ledger technology (DLT) platform, designed for business organizations [1]. A permissioned blockchain system means the members enroll through a Membership Service Provider (MSP) which gives the ability to a group of organizations to form a channel to perform transactions within the channel and maintain a separate ledger. The organizations that take part in building the Hyperledger Fabric network are called the "members". Each member of an organization in the network will setup their peers for participating in the network. Client application uses the service provided by the Hyperledger Fabric network to initiate the transactions. All the peers maintain one ledger per channel that they are registered to providing the distributed ledger within the channel.

5.1.1 Roles in Hyperledger Fabric network Peers in Hyperledger Fabric blockchain network have different roles.

1. Client/Peer
2. Endorser
3. Orderer
4. Commiting peer

Client/Peer: As described the peer is a member in the organizations channel which initiates the transaction request. Based on the outcome of the endorsement result of the proposed transaction, the peer either forwards the proposal to ordering service in turn to commit service.

Endorser: Endorsing peer on receiving the transaction invocation request from the client application validates the transaction. Checks certificate details and roles of the requester. It executes the smart contract and simulates the outcome of the transaction. At the end of the above two tasks the Endorser may approve to disapprove the transaction. But it does not update the ledger.

Orderer: Orderer is considered as the central communication channel for the Hyperledger Fabric network. Orderer peer is responsible for consistent ledger across the network. It is responsible for creation blocks and delivers that to the network. Orderer is built on top of message oriented architecture.

Commiting Peer: Commiting peer is responsible for the committing the blocks given by the orderer. Committing peer updates the state resulting from executing valid transactions of the system.

5.1.2 Transaction Flow

1. Participant in the member Organization invokes a transaction request through the client application.
2. Client application broadcasts the transaction invocation request to the Endorser peer.
3. Endorser peer checks the Certificate details and others to validate the transaction. Then it executes the smart contract and returns the Endorsement responses to the Client. Endorser peer sends transaction approval or rejection as part of the endorsement response.
4. Client now sends the approved transaction to the Orderer peer for this to be properly ordered and be included in a block.
5. Orderer node includes the transaction into a block and forward the block to the committing peer.
6. Committing peer then commits the block in to the ledger and publishes the ledger to the network.

By this the transactions which are only valid are being added to the ledger and made known to the peers in the channel. Hyperledger fabric architecture

is based on the record first and execute later architecture which raises some concerns in the business community. Hyperledger fabric expects some failure transactions and also imposes constraints on the way the request being submitted to the system.

5.2 Corda

Corda [2] is a blockchain-inspired distributed ledger technology that currently targets finance use cases. Like Hyperledger Fabric, it is a permission network where all participants have verifiable identities using public-key infrastructure. Corda allows parties to transact directly, with value and does not use a blockchain to record transactions.In general the blockchain network replicates the ledger across the members of the network. Corda views this approach as privacy concern and restricts the member to maintain their states with themselves. The only entities that should be aware of a transaction are the parties directly involved in a transaction.

In Corda a network map service publishes information about how to reach every identity in the network. This allows any entity to specifically contact any other entity directly. Corda uses point to point communication,a transaction is sent directly to the involved parties which is very quick to process. However, because transactions are not broadcasted to the entire network there is need for block creation and no entity has an entire history of all the transactions in the network.

5.2.1 Role of Members Corda assigns the roles/identities to its members in the initial stage.

1. Client
2. Doorman
3. Notary

Client: The Client is a participant of the network which would like to perform a transaction of assets. In Corda the assets are defined based on the digital representation called Unspent Transaction Output (UTXO). Each client obtains these UTXOs either from an initial transaction performed by central issuing authority or by the involving in a valid transaction.

Doorman: The Doorman is responsible to registering the members in the network and providing the necessary the network services to interact with the other members. Doorman also serves as the authentication service to authenticate the members in each session.

Notary: A notary is a network service that provides uniqueness consensus by attesting that, for a given transaction, it has not already signed other transactions that consumes any of the proposed transactions input states.

5.2.2 Transaction Flow

1. Both parties of an individual transaction initiates the transaction with a set of input state references that will be consumed by the final accepted transaction and their respective output references.
2. Upon agreeing on the output state of the reference objects each participant signs on the transaction.
3. The group of transactions upon agreeing on the output state will be forwarded to the notary service for transaction validation.
4. Notary service validates the input state references of each transaction. Basically it verify the below two conditions.
   i. Check if the input state references provided are valid UTXOs.
   ii. Check if the given UTXOs is not used in another transaction.
5. Based on the outcomes of the conditions Notary updates its state which is a collection of UTXOs in the system .
6. Update by the notary can be referred as the commit to the ledger of the system
7. The clients upon receiving the success response from the notary makes their input reference as history and the output references at their current state.

The history of the state references (UTXOs) of each individual client is stored in their respective vaults. It can be viewed that the architecture of corda is completely different from other blockchain platforms butt provides blockchain kind of environment where the electronic assets can be used to perform a transaction.

5.3 Quorum

Quorum is an Ethereum-based distributed ledger protocol that support transaction and contract privacy. Quorum system is developed to address features like transaction and contract privacy, voting-based consensus mechanism network and peer permissions management and also higher performance.

5.3.1 Roles and Responsibilites of Participants In quorum, the roles and responsibities are predefined. The roles are assigned based on the predefined policies which helps clients to perform business transactions and append to distributed ledger.

1. Quorum Node
2. Transaction Manager
3. Encalve

Quorum Node: Quorum Node is a simple node whose major responsibility is to maintain two databases public and private. Quorum node accepts the transaction requests from the clients/dapps and then forwards the same to the network.

Transaction Manager: Quorums Transaction Manager is responsible for Transaction privacy. It stores and allows access to encrypted transaction data, exchanges encrypted payloads with other participant's Transaction Managers but

does not have access to any sensitive private keys. The Transaction Manager is restful/stateless and can be load balanced easily.

Enclave: The Enclave works hand in hand with the Transaction Manager to strengthen privacy by managing the encryption/decryption in an isolated way. It holds private keys and is essentially a virtual HSM isolated from other components.

5.3.2 Transaction Flow The transaction includes the private transaction happening between A and B. The third member C belongs to the network but will be unable to view the transactions between A and B.

1. As Quorum node passes the transaction on to its paired transaction manager (Transaction Manager A), requesting for it to store the transaction payload.
2. As Transaction manager makes a call to its associated enclave to validate the sender and encrypt the payload.
3. As enclave checks the private key for Party A and, once validated, performs the transaction conversion.
4. Party As transaction manager then stores the (encrypted payload , encrypted symmetric key). And then securely transfers the (hash, encrypted payload, encrypted symmetric key) that has been encrypted with Party Bs public key to Party Bs Transaction Manager.
5. Party Bs Transaction Manager responds with an Ack/Nack response.
6. As Transaction Manager returns the hash to the Quorum Node, which then replaces the Transaction's original payload with that hash.
7. Then transaction is propagated to the rest of the network using the standard Ethereum P2P protocol. A block containing Transaction AB is created and distributed to each party on the network.
8. All the parties attempt to process the transaction. A and B make a call to its enclave, passing in the encrypted payload, encrypted symmetric key, and signature. However the member C will receive a NotARecipient message.
9. The respective enclaves tries to validate the signature and then decrypts the symmetric key using the partys private key that is held in the enclave. The transaction is then decrypted to payload using the now-revealed symmetric key, and returns the decrypted payload to the transaction manager.
10. The transaction managers for parties A and B then send the decrypted payload to the EVM for contract code execution. This execution will update the state in the Quorum node's private StateDB only.

5.4 MultiChain

MultiChain is a platform for the creation and deployment of private blockchains, either within or between organizations [3]. It aims to overcome a key obstacle to the deployment of blockchain technology in the institutional financial sector, by providing the privacy and control required in an easy to use package. MultiChain is aimed to solve the related problems of mining, privacy and openness via integrated management of user permissions.

MultiChain uses cryptography to restrict blockchain access to a list of permitted users, by expanding the "handshaking" process that occurs when two blockchain nodes connect:

1. Each node presents its identity as a public address on the permitted list.
2. Each node verifies that the others address is on its own version of the permitted list.
3. Each node sends a challenge message to the other party.
4. Each node sends back a signature of the challenge message, proving their ownership of the private key corresponding to the public address they presented.

If either of the nodes is not satisfied with the results, it aborts the peer to peer connection. Since multichain is based on the blockchain technology but provides the execution of transactions in a private mode. Since it's a private Blockchain, there is a lot more control about the miners in the chain. This allows having specific mining settings for the chain like mining diversity, blocking size and frequency and as such solving the capacity problems.

A transaction in multichain is similar to that of blockchain, but the only difference is that each node performing a transaction should have the necessary permissions in subscribed private blockchain. Private Blockchain can be informally defined as the set of nodes performing transactions with necessary permissions in that chain.

Permissions Management: In MultiChain, all privileges are granted and revoked using network transactions containing special metadata. The miner of the first genesis block automatically receives all privileges, including administrator rights to manage the privileges of other users. This administrator grants privileges to other users in transactions whose outputs contain those users addresses together with metadata denoting the privileges conferred. When changing the administration and mining privileges of other users, an additional constraint is introduced, in which a minimum proportion of the existing administrators must vote to make a change. These votes are registered by each administrator in a separate transaction, with the change applied once sufficient consensus is reached. The first few blocks of a chain define a setup phase, in which a single administrator is able to bypass this voting process. Future versions of MultiChain could also introduce super administrators who can assign and revoke privileges on their own.

Block Creation: To create a new block, the user has to perform two simple steps.

1. The user chooses a name for the chain, upon which MultiChain creates a configuration file containing the default settings. This file can be modified by the user, although the defaults will be suitable for common use cases.
2. The user launches the blockchain, upon which the genesis block is mined by MultiChain, granting its creator all user privileges.

5.4.1 Transaction flow A message is sent from the originator to the recipient as follows:

1. The originating MultiChain node sends a message transaction to the recipient with metadata containing its IP address and a hash of the messages content.
2. The receiving node receives the transaction and decodes this metadata.
3. The receiving node contacts the originating node via its IP address to retrieve the message, signing the request in order to prove its identity as the intended recipient. This communication takes place using the blockchains existing peertopeer protocol.
4. The receiving node verifies the messages validity by checking its hash against the hash embedded in the original transaction.
5. If the message is valid, the receiving node completes the loop by sending back a second transaction to the sender containing the same message hash.

Once the first transaction is confirmed on the blockchain, the recipient can prove the below points.

- Who sent the message, since the sender reveals their address when signing the transaction,
- The time the message was sent, since the transaction is embedded in a time stamped block, and
- The messages content, since the hash is part of the transaction that was signed.

Ensuring privacy: Multichain addresses the problem of privacy of a transaction and user data by restricting users not to view even the hash of the public keys of an unregistered user. The users of same private blockchain will be able to view the transactions. This gives the users a medium level of privacy compared to the public blockchains. Thus the multichain technology provides the private blockchain to perform the business level transaction privately in a distributed environment.

6 Comparison of Blockchain Platforms

Blockchain platforms discussed in the previous section have varying characteristics and each is best suited for specific kinds of usecases or applications. In this section, we study the suitability of these permissioned blockchain platforms from the perspective of the business requirements identified in Section 2. For the sake of comparison we also include the Bitcion and Ethereum blockchain platforms in the discussion below.

6.1 Guarantees on message delivery

Bitcoin/Ethereum: Bitcoin and Ethereum use peer to peer communication. A message could take multiple hops before reaching the intented reciever and an

error at any of the hops drastically reduces the chances of delivery. This results in a low gaurantee on message delivery.

Hyperledger Fabric: A hybrid method of communication is used in hyperledger fabric where communication between some peers is point to point and some peers is peer to peer. However, due to the permissioned nature where participants are well known and identified, message delivery is guaranteed in hyperledger fabric.

Mutichain/Qourum: Multichain and Qourum both use hybrid communication method. The communication beteen the designated node is performed using point to point method. Multichain and Qourum both provide high gaurantees on message delivery.

Corda: Corda uses point to point communication method which provide high gaurantees on message delivery.

6.2 Guarantees on results of business processes

Bitcoin/Ethereum: Bitcoin has only a single business process to perform fund transfer. The validity of transaction and the result of its execution are performed by scripts that are available in public. Thus, Bitcoin blockchain provides a high level of gaurantee on the results of business processes. Ethereum, generalized the notion of scripts used in Bitcoin to define smart contracts that capture the business logic.

Hyperledger Fabric: Since Hyperledger fabric uses the endorsers and commiting peers, the transaction is executed at single place. This gives high level of gaurantee on business results.

Multichain/Quorum: Multichain and Quorum also use the concept of smart contracts for the execution of a transaction, thereby providing high level of guarantee on business results.

Corda: In Corda, before a transaction is sent to the notary service for validation, each party has to accept on their output values after executing the smart contract. This makes the peers not only having prior knowledge of the outcome of each transaction it also gaurantees on results using the smart contracts.

6.3 Business Continuity

Bitcoin/Ethereum Multiple copies of data exist throughout the network and it is easy to verify the trustworthiness of the information received from peers. Thus a peer that lost its copy could recover easily. Thus these systems provide high business continuity.

Hyperledger Fabric: In Hyperledger fabric peers are gourped into channels, if a node fails it can be recovered by updating the state from commiting peers. This gaurantees a high level of business continuity similar to Bitcoin/Ethereum.

Multichain/Quorum: In Multichain and Quorum, the blockchain shared only contains the summary of the transactions, and the actual payload is available only with the respective nodes. Thus, the onus of protecting data lies with individual nodes, and in case of a failure, it becomes nearly impossible to fully recover.

Corda: Like in Multichain/Quorum, Corda also requires the user to maintaining his data correctly.

6.4 End-to-end security

Bitcoin/Ethereum: End-to-End security of a transaction is high in Bitcoin and Ethereum due to the public availability of the transaction data.

Hyperledger Fabric: In hyperledger fabric, transactions are allowed only between peers in the same channel, and since the same copy of blockchain is available to all the peers in a channel, hyperledger fabric provides high end-to-end security.

Multichain/Quorum/Corda: In Multichain/Quorum/Corda, only the entities involved in a transaction have access to the full data. However, typically several complex business processes require multiple transactions between multiple entities. The connection between the different steps in the business process is ad hoc, and provides only low level of end-to-end security.

6.5 Privacy

Bitcoin/Ethereum: Due to permissinless architecture of Bitcoin and Ethereum, the level of privacy provided is low. However, the identities of the parties of a transaction being psuedonymous, restores some comfort for privacy, which is subject to linkability attacks.

Hyperledger Fabric: The blockchain in hyperledger fabric is shared within the channels and no data is shared between the channels. This achieves high level of privacy between channels but provides no privacy within the channel.

Multichain/Quorum/Corda: In Multichain/Quorum/Corda, although transaction summary (hash) is shared with all the entities, transaction data is shared only within the transacting parties. This provides high privacy.

6.6 Data integrity

All the systems provide a very high guarantee on data integrity because of the use of the blockchain data structure.

6.7 Latency

Bitcoin/Ethereum: In Bitcoin/Ethereum the time consumed for consensus is pretty high which results in high latency.

Hyperledger Fabric: In Hyperledger fabric, the ordering and execution of a transaction are done centrally, thereby resulting in low latency.

Corda: A notary service is used to decide on execution of each transaction where the notary just verifies the validity of the unspent transaction outputs (UTXOs) and updates its state. This makes the latency very low.

Multichain/Quorum: In Multichain and Quorum the latency is medium due to the fact that consensus takes a little bit of time.

6.8 Asynchronous processing

Bitcoin/Ethereum: Bitcoin supports highly asynchronous behaviour because of the use of UTXO data type, which can be thought of as a named entity. The same is not true of Ethereum because of its use of accounts and balances model.

Hyperledger Fabric: In Hyperledger fabric, because of the separation between the ordering and execution, support for asynchronous behaviour is low.

Multichain/Corda: Both Mutlichain and Corda support a high level of asynchronous processing because transaction execution is a simple update to the UTXOs - a trivial task.

Quorum: Due its architecture, a transaction can be executed only in sequence. This makes the quorum systems more sequential in nature.

6.9 Summary of blockchain platforms

1. Bitcoin/Ethereum blockchains are best suited for applications that demand high business continuity and end-to-end security, and can tolerate a slight loss of privacy and throughput.
2. Hyperledger fabric is best suited for applications that demand high business continuity and end-to-end security, and can tolerate slight loss of privacy - restricted to within a channel.
3. Corda is best suited for applications that demand high privacy and throughput. Business continuity and end-to-end security are adversly effected due to the design choice of Corda.

4. Multichain/Quorum are best suited for applications that demand high privacy and moderate throughput. Business continuity and end-to-end security in these systems are adversly effected.

7 Conclusions

In the literature on blockchain technology, different blockchains have been studied and compared based on their design, technical features and architecture. In this paper, we have listed the important technical requirements for businesses, and identified gaps in the current architectures. Further, we studied prominent permissioned blockchains and compared them against the desired business requirements. Hyperledger fabric, Multichain, Quorum and Corda are discussed in detail along with the roles, responsibilities and transaction flows in those platforms. The existing blockchains address some gaps in centralized systems, but need more research. In this paper, we have also provided the suitability of blockchains based on the business priorities. It is important to note that none of the existing blockchains sufficiently address all the business requirements. This paper, makes an attempt to unveil the necessity of further research required in the promising direction of distributed ledger technologies.

References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. pp. 30:1–30:15. EuroSys '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3190508.3190538, http://doi.acm.org/10.1145/3190508.3190538
2. Brown, R.G., Carlyle, J., Grigg, I., Hearn, M.: Corda: An introduction (2016)
3. Dr Gideon Greenspan, F., CEO, C.S.L.: Multichain private blockchain white paper
4. Macdonald, M., Liu-Thorrold, L., Julien, R.: The blockchain: a comparison of platforms and their uses beyond bitcoin. Work. Pap pp. 1–18 (2017)
5. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf
6. Niforos, M., Ramachandran, V.: Blockchain: Opportunities for private enterprises in emerging markets. Washington, DC: International Finance Corporation (2017)