Blockchain Platforms: A Compendium

Chinmay Saraf
Department of Computer Engineering
Pune Institute of Computer Technology, Pune, India
Email: sarafc@acm.org

Siddharth Sabadra
Department of Computer Engineering
Pune Institute of Computer Technology, Pune, India
Email: sabadrasiddharth@gmail.com

***Abstract*—In recent years, cryptocurrencies gained popularity with Bitcoin. The main promising technology behind Bitcoin was 'Blockchain'. Blockchain provided unique features like transactional privacy, system transparency, immutability of data, security with cryptography, etc. These features paved way for Blockchain in advancing many technologies like voting systems, IOT applications, supply chain management, banking, healthcare, insurance, etc. Blockchain development was boosted with the increasing demand of the technological update. Many blockchain platforms are available like hyperledger fabric, ethereum, corda, etc. We always end up with perplexity while choosing a platform for blockchain development. Through our survey, we provide a comparative analysis of all the hyperledger platforms, ethereum, corda to make a choice of the platform easily according to the requirement.**

***Keywords*—Blockchain, Ethereum, Hyperledger, Corda, Ether, Sawtooth, Iroha, Burrow, Indy, Fabric**

I. INTRODUCTION

Transactions on the internet generally have to rely on the third parties for the its processing. A part of the transaction is paid to the third party which increases the transaction cost[1]. It also induces limitation on the minimum cost of the transaction that can be carried out. A peer to peer electronic cash system as Bitcoin enabled us to do financial transactions without trusting any third party. Here we rely on cryptography for the transactions instead of trusting the third party. With Bitcoin, digital or virtual currency called cryptocurrencies [2] gained popularity. Many cryptocurrencies like Litecoin, Peercoin, Swiftcoin, Peercoin, Ripple, etc were formed after Bitcoin. These cryptocurrencies boosted the blockchain development.

Blockchain is a distributed ledger which means all the parties taking part in the transaction or the parties present on the blockchain has the copy of the ledger and there is no centralised database. In case of any failure of the centralized database it may lead to data loss but with blockchain this problem is solved. Another important advantage provided is the transparency of the transactions.

Blockchain has found application in many new technologies with its promising features. Digital Identity management can be done by blockchain where we can control our identity without depending on any central authority[3]. It enables us to share our identity according to the need and can protect user's consent. OneName is one of the companies providing such digital authentication. In 2015 Bitcoin foundation started a new project on the blockchain based voting system to ensure transparency in the voting system with every vote being recorded on the blockchain. Immutability, cryptographic hash, and transparency are the advanced features provided for the blockchain based voting system. In Supply chain Management Blockchain enabled us to keep the track of the origin of the products. We can ensure the quality of the end products by maintaining the immutable record on the blockchain. Further Blockchain is also used with IOT, medical services, baking, real Estate, etc.

While using Blockchain in any of the applications we face common questions like-

1) Which blockchain platform has to be used?
2) What are the advantages and disadvantages of different platforms?
3) Which consensus is used by which platform?

Through our survey, we provide a overview of blockchain platforms like Hyperledger fabric, corda, and ethereum which are the major platforms for blockchain development. Further, we provide the difference between different blockchain platform hosted under Linux foundation.

II. ETHEREUM

A. What is Ethereum?

Ethereum is a project which is used to build the technology on which all transaction-based concepts can be built. It provides a system to end-developer for building software on formerly unexplored computer models in the mainstream. The key goal of this project is to facilitate transactions between individuals who would have no means to trust one another[4].

Ethereum is a transaction based state machine. It begins with a genesis state[5] and incrementally execute a transaction to modify it into a final state. The state can contain various data as, balances, data pertaining, etc. There can be valid or invalid state changes. Invalid state changes can be due to invalid account balance modifications i.e. increase in receiver's account balance without simultaneous reduction in the sender's account balance. Formally,

$$\sigma_{t+1} \equiv Y(\sigma_t, T)$$

where, Y is Ethereum state transition function, which allows components to carry out computation, σ allows components to store arbitrary state between transactions.

Many transactions are combined in a block using Merkel tree [6] and blocks are chained together using

978-1-5386-5283-1/18/$31.00 ©2018 IEEE

the cryptographic hash [7]. Blocks function as a journal, recording a series of transactions together with previous block and identifier for final state.

B. Currency in Ethereum

In order to encourage computation within a network, there should be an agreed method for transmitting data. To address this issue ethereum has an intrinsic currency called "Ether" or ETH, smallest sub-denomination of Ether is Wei[4].

TABLE I: Ethers and various multipliers [4]

Mutiplier	Name
10^0	Wei
10^{12}	Szabo
10^{15}	finney
10^{18}	Ether

C. Account in Ethereum

In Ethereum, state is made up of objects called as Accounts [8]. Each account has 20-bytes address. State-transition is transfer of value and information. Each account consist of four fields:

1) Nonce- Counter used to make sure that transaction is processed once
2) Ether balance
3) Contract Code
4) Storage

There are two types of accounts in ethereum:

1) Externally Owned Accounts (EOAs), which are controlled by private keys.
2) Contract Accounts, which are controlled by their contract code and can only be activated by an EOA.

D. Transaction in Ethereum

The term Transaction is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Message call and account creation are the two types of transaction. Transactions contain the recipient of the message, a signature identifying the sender, the amount of ether and the data to send, as well as two values called Start (Gaslimit) and Gas price[8].

- Gas price- A scalar value equal to the number of Wei to be paid per unit of gas for all computation costs incurred as a result of the execution of the transaction.
- Gaslimit- Every transaction has a specific amount of gas associated with it called as GasLimit. Gaslimit is the amount of gas which is implicitly purchased from the sender's account balance. We can refer gas limit as, the total amount of gas a sender is willing to purchase for the transaction to be executed.

The Transaction Process is explained below[4]:

1) Check if the transaction is well-formed (i.e. has the right number of values), the signature is valid, and the transaction nonce matches the nonce in the sender's account. If not, return an error.
2) Calculate the transaction fee as START GAS * GASPRICE and determine the sender's address from the signature. Subtract the fee from the sender's account balance and increment the sender's nonce. If there is not enough balance to spend, return an error.
3) Initialize GAS = START GAS, and take off a certain quantity of gas per byte to pay for the bytes in the transaction.
4) Transfer the transaction value from the sender's account to the receiver's account. If the receiving account does not yet exist, create it. If the receiving account is a contract type account, run the contract code either to complete the transaction or until the execution runs out of gas.
5) If the value transfer fails due to the sender's low balance or due to unavailability of gas, revert all state changes except the payment of the fees, and add the fees to the miner's account.
6) Otherwise, refund the fees for all remaining gas to the sender, and send the fees for gas consumed by the miner.

E. Technologies used in Ethereum

1) **Data Storage** -Swarm[9] is a distributed storage platform and content distribution service, a native base layer service of the ethereum web 3.0 stack. The primary objective of Swarm is to provide a decentralized and redundant store for ethereum's public record, in particular, to store and distribute dapp code and data as well as blockchain data.
2) **Web Technology** - Web 3.0 is an internet where core services like DNS and digital identity are decentralized. Ethereum is perfectly suited to serve as shared back-end to a decentralized and secure internet. Ethereum uses Web 3.0 as a platform for a decentralized application (DApp).
3) **Client/Node implementation**- Ethereum clients are used for sending and receiving data. Wallet or custom applications connect to a client for transactions, may be for sending or receiving ethers, deploying a contract, executing a contract, initiating mining, etc. Ethereum has many client/ node implementation as go-ethereum, parity, cpp-ethereum, pyethapp, etc. Smart Contracts in ethereum are implemented using languages like Solidity[10], Lisp, etc.

F. Consensus Algorithms in Ethereum

A fundamental problem in distributed computing and multi-agent systems is to achieve overall system reliability in the presence of a number of faulty processes. This often requires processes to agree on some data value that is needed during computation. The algorithm involved in such processes is Consensus algorithm. There are mainly 3 types of consensus algorithms.

1) Proof of Work (PoW) [11]: Mining is the process of dedicating efforts to strengthen the series of transactions in one block over other competitor blocks. Any node on public blockchain can act as a miner. The only requirement is that it should support resource requirement for mining in PoW. In Proof of Work (PoW), miner has to solve the puzzle for validating the new block to be added. The first miner who gets the solution publishes it to the network. The difficulty level of puzzle changes for different blocks. Hence, the amount of computations differs for different blocks.

Hashing function takes input data and converts to Hash value also known as the message digest which is unique. The hash value cannot be used for recreating the original data. Hence, the hashes used in such case are called as One Way Hashes. The puzzle is:

$$Hash(Data + X) = HashValue$$

Guess the value of **X** such that there are **N** leading zeroes in the hash value. X is referred as Nonce which is some random value and N decides the difficulty of the puzzle. The difficulty is directly proportional to N. Hence, this puzzle can only be solved by Brute-force algorithm i.e. iteratively checking the X. PoW uses GHOST protocol and ETHash algorithm. But, PoW is environmentally unfriendly due to its high power consumption. Presently, Ethereum is using PoW as Consensus model.

2) Proof of Stake (PoS) [4]: This algorithm is similar to PoW, the only difference is that here there is no competition as in PoW. Here, the network itself chooses the node which would validate the transaction known as the validator (not a miner). If node selected as validator does not validate the transaction then network selects next validator and process goes on untill the transaction is validated by any node. PoS uses CASPER protocol. In future, Ethereum will adopt PoS.

3) Proof of Authority (PoA) [4]: In this consensus algorithm, network allows for only those nodes which are authorized for validating the transaction. The chain has to be signed off by the majority of authorities, in which case it becomes a part of the permanent record. This makes it easier to maintain a private chain and keep the block issuers accountable.

III. HYPERLEDGER

Hyperledger project was started in December 2015 under Linux foundation for open source blockchain development with an aim to improve the performance and reliability of these systems. There are 5 types of the hyperledger frameworks under the hyperledger project for the blockchain development.

A. Burrow

Ethereum is permissionless and public blockchain. Ethereum platforms for the blockchain deployment does not provide a functionality to make permissioned blockchain. Ethereum blockchain can be deployed in permissioned way using Hyperledger Burrow [12] [13]. It originated from Manox, the ecosystem application platform and was co-sponsored by Intel and further hosted by the Linux foundation. The smart contract development functionality is provided by manox chains.

Burrow is a multi-chain universe having application specification formulated earlier. The Burrow node has three main components like the consensus engine, the permissioned ethereum virtual machine, and the RPC gateway. The consensus algorithm used is Byzantine fault tolerant Tendermint which provides a high throughput with the known set of validators. The malicious node can be easily tracked with the tendermint. Each block in the blockchain has a height assigned to it such that all the validators have the latest added block with the same height. This height is included in the header of the block.

The consensus algorithm consists of three parts for adding the block to the blockchain. Those are-

1) Proposal- The proposer should propose the transaction and it should be visible to all the validators in the stipulated amount of time and if not, the proposal is discarded.
2) Voting- There are 2 phases of voting called pre-vote and pre-commit. If the transaction is supported by two-third of the validators in both phases then it is called to receive a Polka for a block and then it can be committed.
3) Locks- To ensure that, no 2 validators commit different blocks at the same height it uses locking mechanism. This mechanism determines which validator can commit.

One of the disadvantage of this mechanism is that the system may halt if one third or more than one-third of the validators are offline. An advantage provided is that consensus through proof of work using mining is avoided which requires a lot of computations and leads to loss of resources.

B. Fabric

Hyperledger fabric [14][13] is a platform to create a private and permissioned blockchain platform hosted by Linux. The members of the blockchain network have to register through a Membership Service Provider. We can store the ledger in different formats, the consensus mechanism can be changed according to the requirement and the different MSPs are supported. Sometimes in blockchain network, we want to avoid disclosing a transaction to all the participants on the network In hyperledger fabric we can create channels with a group of members which will have a separate ledger disclosed only to the members of the group. The different functionalities provided by the hyperledger fabric are-

1) **Identity management**- As it is a private blockchain the members have to register to the network and a user id is provided to all the members.
2) **Privacy and confidentiality**- The functionality of private chains is provided as discussed earlier.
3) **Efficient processing**- Parallelism increases the system's efficiency as the transaction execution and transaction commitment and ordering are all

TABLE II: Difference between hyperledger platforms

No.	Property	Burrow	fabric	Indy	Iroha	Sawtooth
1	Hosted by	Linux foundation but initiated by Sovrin foundation	Linux foundation	Linux foundation	Linux foundation but started by japanese fintech company Soramitsu Co.,ltd.	Linux foundation but started by Intel
2	Status	Incubation	Active	Incubation	Active	Active
3	Consensus Algorithm	Proof of Stake protocol called tendermint	Any consensus can be plugged some are SOLO,KAFKA , SBFT	RBFT	Byzantine fault Tolerant Algorithm called sumeragi	It supports different types of consensus on the same blockchain like PoET, PoET simulator, dev mode, RAFT
4	REST API	Available	Available	Not Available	Available	Available
5	Interoperability	Ethereum blockchain can be deployed in permissioned manner	None	None	None	Ethereum blockchain can be deployed using seth
6	Cryptography	ED25519/SHA512	ECDSA and SHA256(abstracted)	ED25519	ED25519/SHA512	SECP256K1 ECSA
7	Types of Consensus Algorithm	Voting Based	Application Dependent	Voting Based	Voting Based	1.Poet -Lottery Based 2.poet simulator voting based 3.RAFT-voting based

separately performed. Multiple transactions can be executed simultaneously.

4) **Chaincode functionality**- Through chain code we can apply different transaction rules for validation and endorsement to different groups in the blockchain.
5) **Modular design**- The modular design facilitates the designers to plug any algorithm for identification, encryption, and consensus in the blockchain network.
6) **State database**- State database stores the current state of the key-value pairs from the ledger. The state database helps us to increase the efficiency of the transaction as we can extract the latest key-value pairs easily. The state database can be implemented using CouchDB and embedded LevelDB.

Consensus in hyperledger fabric consists of three phases namely Endorsement, ordering and Validation-

1) Endorsement- For endorsing any transaction the user has to follow an endorsement policy.
2) Ordering- In ordering phase, the transactions endorsed are accepted and placed in order chronologically by channel and places them in the block according to the channel.
3) Validation- In validation the transactions are again validated and the endorsement policy is checked again and the transactions in the block are tagged valid or invalid accordingly.

Hyperledger fabric supports pluggable consensus according to the requirement for all these phases. Some consensus protocols are provided like SOLO, KAFKA.

Advantages of fabric are:

1) We can query the information from the ledger just like the SQL queries.
2) Private channels.
3) Tangible as well as intangible assets can be created.

C. Indy

Joining or participating in any of the organization or institute requires identification and verification of the identity. This same procedure is repeated for all the institutions and is a time-consuming process. Digital identity paves a way to avoid reliance on physical documents and manual processes for the identity management. Digital identity can be given access to all these institutions easily with the user's explicit consent. By maintaining a digital identity on the distributed ledger all the institution can easily share a common identity. Hyperledger Indy [15][13] was built with this ideology. It was a brainchild of the Sovrin foundation.

As the organizations keep our identity in a centralized database, we rely on them for the security of our information, but with Indy which uses the distributed ledger the user can manage its privacy and decide the disclosure of their identity without relying on any organization.

The consensus protocol used is RBFT [16]. Byzantine fault tolerance protocols are not efficient when a fault is encountered. The current BFTs rely on a special replica called as primary to order the requests. If this primary is a malicious node then it can delay the requests of the transactions and can cause different problems. The redundant BFT is a new approach for BFTs inspired by plentum BFT. It runs multiple instances of plentum protocol to ensure that the system is fault tolerant. By doing this we can detect the malicious node.

D. Iroha

Fintech is the new technology having the capability to disrupt the current financial service sector by providing a better platform for these services. Soramitsu, the fintech company in Japan open-sourced the code for the project named IROHA [17][13]. This project was hosted by the Linux foundation under the Hyperledger project. It is developed in C++. This system is developed to share the information between the untrusted parties which uses a private blockchain. The main difference between IROHA and other blockchain is that every participant is not allowed to store all the data history. The users can only query the data

if they have been authenticated and have the permission. This system is developed for the clients having diverse application and peer hardware from embedded system to enterprise-class servers. The consensus algorithm used is Byzantine fault Tolerant consensus Algorithm called sumeragi which is highly influenced by the B-Chain algorithm.

The goal of Iroha is to provide C++ libraries to the hyperledger projects. These libraries are Summeragi consensus library, SHA-3 hashing library, API server library, Javascript library, etc. The cryptographic method used by IROHA is SHA256. Chaincode is used to apply the transaction rules and validations. Java based chain code is supported.

E. Sawtooth

Hyperledger Sawtooth framework is the distributed ledger or blockchain which is aimed to set the business enterprise by setting all the parties on the blockchain and maintaining the track of all the parties related to the business [18][13].

For setting the business on the blockchain, the business rules can be set easily without knowing much about the design of the core system. The consensus algorithm, transparency rules, and the permissions can be designed according to the requirement of the application. Different SDKs are available in different languages like Python, Go, Javascript, Rust, C++, and Java. With Sawtooth, we can write a smart contract in any language of our choice. A REST API is also provided for the development. Many blockchains have serial execution of the transactions but the sawtooth framework provides an advantage of a parallel execution of the transactions which increases the performance time. We can implement ethereum blockchain using the Sawtooth-Ethereum integration project, Seth. We also have an option to write smart contracts in solidity with Seth. Thus interoperability between ethereum and hyperledger is provided using Seth.

The consensus in a mechanism which has to be executed before a block is added to the blockchain or a transaction is to be performed. In sawtooth, we can have multiple consensuses on the same blockchain. The consensus used in sawtooth are-

1) Proof of elapsed time protocol [19]- It is designed to handle a large network population. It is also a lottery based algorithm just like the ethereum's proof of work(PoW). It was designed by Intel. This protocol solves the Byzantine generals problem.
2) Proof of elapsed time simulator- It provides a poet-style consensus on any type of hardware, including a virtualized cloud environment.
3) Dev mode- It a simplified random leader algorithm that is useful for developing and testing.
4) RAFT- It is a voting-based protocol. The features provided by this protocol are fault tolerance, provides high throughput and low latency transactions.

IV. CORDA

A. What is Corda?

Corda [21] is a distributed ledger platform for recording and processing financial agreements. It is specialized for use with regulated financial institutions. It is inspired by blockchain systems, but without the design choices that make traditional blockchains inappropriate for many financial scenarios.

Some features of CORDA are-

- Recording and managing the evolution of financial agreements and other shared data between two or more identifiable parties in a way that is grounded in existing legal constructs and compatible with existing and emerging regulation.
- Validating transactions between parties.
- Coordinating work-flow between different firms without central authority/system.
- Supporting various Consensus algorithms.
- Supporting consensus between firms without involving firms which are not part of the transaction.
- Using industry-standard tools.

B. State Object in Corda

The basic concept of an object in corda [21][22] is state object, which is a digital document which records the existence, content and current state of an agreement between two or more parties. Only privileged parties can access the state object of agreement between parties. The ledger is defined as a set of immutable state objects.

C. Consensus in Corda

1) Transaction Validity- Parties can be certain that the transaction which is updated is valid by checking the contract code, required signatures, etc.
2) Transaction Uniqueness- Parties can be certain that the transaction is unique as they haven't reached consensus(validity and uniqueness) that consumes any of the same states.

In corda, some data is "on-ledger" only if at least two actors are in consensus i.e. at least two actors should be in existence for consensus process of the data. Data with a single actor on consensus is "off-ledger".

Nodes are arranged in an authenticated peer to peer network. All communication is direct. There is no blockchain. Transaction races are deconflicted using pluggable notaries. A single Corda network may contain multiple notaries that provide their guarantees using a variety of different algorithms. Thus, Corda is not tied to any particular consensus algorithm.

D. Business Logic in Corda

Corda forces business logic that either accepts or rejects the transaction. The business logic [21] is implemented using smart contract code which can be composed from

TABLE III: Difference between Ethereum, Hyperledger and Corda [20]

No.	Property	Ethereum	Hyperledger Fabric	Corda
1	Platform type	Generic blockchain platform	Modular blockchain platform	For financial industry
2	Governance	Ethereum developers	Linux foundation	R3
3	Mode of operation	permissionless, public or private	permissioned, private	permissioned, private
4	Consensus	1.Mining based on proof-of-work (PoW) 2.Ledger level	1.Broad understanding of consensus that allows multiple approaches 2.Transaction level	1.Specific understanding of consensus(i.e., notary nodes) 2.Transaction level
5	Currency	Ether	None	None
6	Contract language used	Solidity	Go, Java	Kotlin, Java
7	Data Storage	Swarm	CouchDB, LevelDB	—
8	Access to data	1. All (Public Network) 2.Authorized (Private Network)	Authorized (Private Network)	Only to relevant parties.

simpler, reusable functions. Contracts define part of business logic, which are mobile. Nodes have to download it and execute it on their sandbox. It uses JVM for contract execution and validation. The reason behind using JVM is to reuse the older codes of specific party, to connect with the corda. Also, JVM has huge set of functions, libraries which makes it easier to develop contract code.

V. CONCLUSION

Blockchain has the potential to disrupt the existing technologies with its features like immutability, transparency, cryptographic hashing techniques, etc. In this survey, we have analyzed various blockchain platforms with their protocols and consensus algorithms. We have compared various platforms under hyperledger project and we have also provided comparison between hyperledger fabric, etherum and corda. Thus, we conclude that Hyperledger can be used in applications where security and privacy is preferred, Etherum can be used where miners are required and Corda can be used where financial agreements are essential.

We hope that our survey can be used for referring the advantages and disadvantages of different platforms and can pave the way for fruitful application development as per the application's need.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] *List of cryptocurrencies*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_%20cryptocurrencies (visited on 02/02/2018).

[3] M. Z. F. Xavier Olleros, "Research handbook on digital transformations," 2016.

[4] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger. ethereum project yellow paper 151 (2014)," 2014.

[5] *Genesis Block*. [Online]. Available: https://en.bitcoin.it/wiki/Genesis_block (visited on 11/01/2018).

[6] *Merkle Tree*. [Online]. Available: https://en.wikipedia.org/wiki/Merkle_tree (visited on 05/02/2018).

[7] *List of cryptographic hash function*. [Online]. Available: https://en.wikipedia.org/wiki/Cryptographic_hash_function (visited on 03/02/2018).

[8] V. Buterin, "Ethereum white paper, 2014.," 2014.

[9] *What is Swarm and what is it used for?* [Online]. Available: https://ethereum.stackexchange.com/questions/375/what-is-swarm-and-what-is-it-used-for (visited on 05/02/2018).

[10] *Solidity*. [Online]. Available: http://solidity.readthedocs.io/en/latest/ (visited on 06/02/2018).

[11] *Proof-of-Work System*. [Online]. Available: https://en.wikipedia.org/wiki/Proof-of-work_system (visited on 06/03/2018).

[12] *Hyperledger Burrow*. [Online]. Available: https://github.com/hyperledger/burrow/blob/master/README.md (visited on 05/02/2018).

[13] *Hyperledger - Blockchain Technologies for Business*. [Online]. Available: http://www.hyperledger.org (visited on 05/02/2018).

[14] *Hyperledger Fabric*. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.1/ (visited on 05/01/2018).

[15] *Hyperledger Indy*. [Online]. Available: https://github.com/hyperledger/indy-node/blob/stable/getting-started.md (visited on 04/02/2018).

[16] V. Q. Pierre-Louis Aublin Sonia Ben Mokhtar, "Rbft: Redundant byzantine fault tolerance," 2013.

[17] *Iroha Distributed Ledger Technology*. [Online]. Available: https://www.hyperledger.org/projects/iroha/resources (visited on 05/03/2018).

[18] *Hyperledger Sawtooth*. [Online]. Available: https://www.hyperledger.org/projects/sawtooth (visited on 05/02/2018).

[19] *PoET Specification*. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture/poet.html (visited on 11/03/2018).

[20] V. M. and S. P., "Comparison of ethereum, hyperledger fabric and corda. fsbc working paper," 2017.

[21] M. Hearn, "Corda: An introduction," 2016.

[22] M. Hearn, "Corda: A distributed ledger," November 29, 2016.