# Hyperledger Fabric – A Platform for Enterprise Blockchain Solutions: Architecture and Transaction Flow

Tinu N.S[#1]

[#1]*Department of Computer Science and Engineering , New Horizon College of Engineering*

[1]tinuns@newhorizonindia.edu

***Abstract*— Hyperledger is an open source, collaborative forum of diverse industries under the supervision of the Linux Foundation, started in the year 2015. The reason behind such an enriched association came with the foresight that working together as a team can bring more innovation than being standalone. Hyperledger has crossed more than 250 organizations in multidisciplinary domains as members- from Airbus to IBM and they have been working on many interoperable enterprise blockchain projects. Hyperledger Fabric is a hyperledger framework initially contributed by IBM and Digital Asset. It became the first modular and extensible permissioned blockchain system primarily for enterprise distributed applications. Fabric could support diverse industry use cases through its pluggable-modular order-execute-validate architecture for components such as membership services, consensus, privacy and chaincodes, thus assuring a new era of trust, interoperability, privacy, transparency and accountability for the collaborative enterprise projects.**

***Keywords*— Hyperledger, Hyperledger-fabric, order-execute-validate architecture, pluggable modules, transaction flow**

## I. INTRODUCTION

Blockchain has now extended its arms to engulf the business world, rather than merely staying in the technical websites as a buzzword. The progress in blockchain technology is so immense and rapid. The major reason is, being an open source technology, in which any one can contribute to (Bitcoin by Satoshi Nakamoto 2008and ethereum by Vitalik Buterin 2013) and also the involvement of major tech gaints like IBM, Microsoft, Facebook and other market boomers like the Walmart, De Beers, to name a few, who have already adopted this latest technology. Hyperledger- an open source blockchain collaboration and Hyperledger fabric- a highly modular, permissioned blockchain solution from IBM, Microsoft Azure, the cloud infrastructure of Microsoft, which has embraced the blockchain technology, are few examples.

Blockchain, the immutable distributed Ledger Technology (DLT), has already chained into the network of globe of enterprises which involves mutually untrusting peers, with its unique features of being shared, private, secure, immutable, auditability and consensus[1]. Being a new technology with unique characteristics does not mean that all industry or business applications must adopt blockchain. Clear guidelines on the need and who all can adopt the technology to bring potential improvement in their business are already set[2]. The technology has got wide acceptance in various finance, e-commerce and product manufacturing domain, where multi-party-dependancy (MPD) is unavoidable. Banking, supplychain , IOT, Healthcare, Bigdata analytics are few sectors which blockchain is closely coupled with. As it has to satisfy variety of requirements from variant application domains, blockchain is mainly classified into, permissionless or public- anonymous participants, permissioned or private- participants are known and consortium (combination of public and private) with selected privacy, networks. Blockchain having this broad distribution of peer nodes, need some accord to bring trust into the network, which is achieved using the consensus protocol/algorithms. Consensus is mainly used to validate the transactions with in the new block. The valid transactions are ordered for consistency and added to the hash chain.

Public or Permissionless blockchain, in which the peers are totally unknown, generally includes a natal cryptocurrency and mostly uses Proof of Work(PoW) as the consensus mechanism with a motivating incentive strategy to run the network. The notion of permissioned blockchain was to bring together the global enterprise communities, that do not trust each other fully but that can interoperate and solve enterprise blockchain challenges, one such network is the hyperledger fabric. The identity property allows the permissioned blockchain to cope up with non- determinism, performance attacks and use Byzantine-fault tolerant(BFT) consensus.

TABLE I
COMPARISON OF FEATURES - PERMISSIONLESS AND PERMISSIONED BLOCKCHAINS

| **Blockchain platforms- Vs - Features** | **Governance** | **Ledger Type** | **Cryptocurrency** | **Consensus** | **Smart Contract support** |
|---|---|---|---|---|---|
| **Bitcoin** | Satoshi Nakamoto | Permissionless | bitcoin | PoW | No |
| **Ethereum** | Ethereum Developers | Permissionless | ether | PoW,PoS | Yes |
| | | Permissionless | | | |
| **Quorum** | JPMorgan Chase | Permissioned | JPM Coin | Raft, BFT | Yes |
| **Hyperledger fabric** | Linux Foundation | Permissioned | None | Pluggable | Yes |
| **R3 Corda** | R3 LLC | Permissioned | None | Pluggable | Yes |

This paper focuses mainly on Hyperledger Fabric, its components and architecture, features of fabric and the execute-order-validate transaction flow adopted in fabric that contributed to the increased scalability, flexibility, confidentiality, resilience and most importantly end-to-end throughput[3]. Fabric should be a very good option for businesses that involves multi-party dependency(MPD) with the need for trust, privacy, transparency, accountability due to its pluggable modules, support of general purpose programming languages for distributed applications and lack of dependency on a natal cryptocurrency.

## II. BACKGROUND

In a multi-party network, the most widely used fault tolerance mechanism, whether it be byzantine fault tolerance(BFT) is active replication or state machine replication(SMR)[4]. Most of the existing blockchains with chaincode support, implements this same SMR, ie all replicas execute the operations and sent replies. In traditional blockchain systems, that range from permissionless Ethereum to permissioned quorum, Tendermint, it follows order-execute architecture, where it requires all peers in the network to execute every transaction first and later ordered. This require the transactions to be deterministic also. It affects the scalability feature of the entire blockchain system as the transactions need to be executed sequentially and endorsed by all peers. The further limitations of the order-execute architecture are

- *Affects performance:*
  - as it needs to encounter denial-of-service attacks, originating from untrusted smart contract functions.
  - also due to the sequential execution order of transactions.
  - Non deterministic transactions further escalates the trouble, by forking the ledger.

- *Lack of flexibility and adoption by general crowd*
  - due to the dependency of smart contracts written in domain specific programming language like solidity in Ethereum.
  - Consensus is hard-coded, which contradicts the BFT notions.

- *Deteriorates confidentiality and trust model*
  - All peers execute every smart contract, that hinders privacy within the parties.
  - Consensus protocol for transaction validation gets affected by the smart contract.

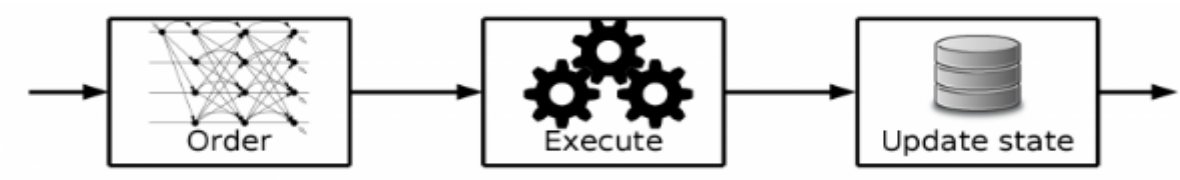



Fig. 1 Traditional Order-execute architecture

## III. HYPERLEDGER FABRIC COMPONENTS AND ARCHITECTURE

Hyperledger fabric is the first blockchain system that supports implementation of distributed applications or chaincodes written in general purpose languages like Go, Java, JavaScript, Python[5]. The implementation is in such a way that chaincodes are pluggable and can be executed consistently across many nodes, thus making it the first distributed operating system[6] for permissioned blockchains. The new architecture of execute-order-validate uses a hybrid replication design that combines both passive (few peers perform replication) and active replications.

The fabric architecture and its transaction flow would be better explicable with a brief introduction to its key components.

a) Membership Service Provider (MSP):- Provides identity required for the participants in the blockchain network (peers-endorser, committer, orderers', client applications, administrators etc). Identity is nothing but the digital certificate, required to provide authentication and access control permissions, for the transactions performed by the participants. The certificate obtained by the participating nodes will have information regarding their privileges and attributes, using which they transact in the blockchain. Fabric has got two certification authorities(CA's), one is the external CA and other is the stand-alone CA called fabric-CA. Certification authority is a pluggable module in fabric and as long as the root CA is recognized by the network, any standard CA could be used[7]. Fabric uses X.509 certificates MSP implementation by default as identities, that adopts the traditional Public Key Infrastructure (PKI) hierarchical model.

b) Hyperledger Fabric Client (HFC) SDK: Which is the client interface developed in languages like Node.js, Java, Python etc..Client applications use set of SDK's to submit transactions on the blockchain.

c) Types of Peers: Fabric is the permissionless blockchain where multiple organizations are involved and in turn, multiple peers from an organization can also participate. Peer nodes maintain the ledger, chaincode and events in blockchain. Multiple peers exists as an aid for fault tolerance and peers might also have specific roles associated with, in the network. Peer nodes can serve as

   i) Endorsing peer
   ii) Committing peer
   iii) Both endorser and committer

Endorsing Peer or Endorser: The transactions submitted by the client SDK will be received by multiple endorsing peers(depending on the endorsement policy set). These endorsing peers will have a copy of the chaincode which it executes (unlike all peers in order-execute architecture), and the output of that transaction would be signed or endorsed by the endorsing peer/peers using its certificate. Multiple clients can sent transaction proposals asynchronously to the endorsing peers. After executing transactions, these endorsing peers would endorse or sign the transactions. Few peers act as both endorser and committer for scalability factor.The executed transactions would never be committed to the blockchain yet.

Endorsement Policy- Determines, how many peers that particular client should send transaction proposals to. Usually this information can be defined along with smart contract.

d) Ordering Service: Receives transactions from different peers asynchronously, across the network and provides a totally ordered set of transactions. These ordered set of transactions are communicated to all the peers, so that they can commit all transactions in the same order, which will maintain consistency across the entire network. Ordering node neither maintain smart contracts nor copy of ledger, but just perform ordering. Ordering service is a decentralized module that might be owned by some organization that already have peer nodes in the network.

e) Committing peer: Maintains ledger and state. Once transactions are ordered by the ordering service module, committing peer would commit those transactions in the same order. These peers may or may not hold chaincodes.

f) Channels: Are the communication paths that provide privacy among different ledgers. Channels enable the peer nodes to be part of multiple blockchain ledgers. For example, consider E0,E1,E2,E3 are the four endorsing peers in permissioned blockchain system, now E0,E1,E2 can be a part of one blockchain ledger and E0,E2,E3 can be part of another ledger. Thus it forms two different chains of block, which can have either the same ordering service or different ordering service, but with different scope of the ledgers. Channels thus provide multitenancy, which means, different applications incorporate different set of peers, which executes on each channel, thus can form consortiums with in the network of organizations.

## IV. FABRIC ARCHITECTURE

Fabric with its plug-n-play modules became the first distributed operating system for permissioned blockchains, written in general purpose programming langauages. Smart contract, also called chaincodes are the program code that holds the business logic, which gets invoked during the execution phase. Chaincodes is pluggable module that can be written by an untrusted developer. There are mainly two types of chaincodes: one for maintaining the blockchain network system called the system chaincodes and the other is the application chaincodes. Hyperledger fabric do not have built-in cryptocurrency.

The new architecture introduced by the fabric, execute-order-validate , as depicted in the fig.2.It consists of mainly three phases:

- Execution Phase
- Ordering Phase
- Validation Phase

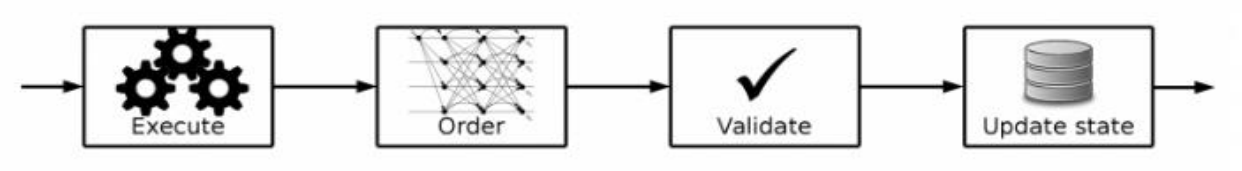



Fig.2 Hyperledger fabric architecture

- Execution Phase- In this phase, the transaction proposals submitted by the client is executed by the endorsement peers by invoking the chaincode's function. Multiple clients can submit proposals asynchronously The function is executed against the current state database and the transaction result is obtained. The result includes a response value and read-write set (key/value pairs whether to create or update an asset). This output is then signed by the corresponding peer for authentication. The endorsement peers involved in the process must attain same output for the proposal, otherwise is an indication of invalid transaction, which would be identified during validation stage. The ledger is not updated in this phase.
- Order Phase- The ordering service collects results of endorsed transaction proposals coming to it asynchronously from multiple peers and make sure that all transactions are ordered and all peers view these transactions in the same order. This ordered transactions forms a block. These blocks are delivered to peers either through *gossip* or by ordering service itself.
- Validation Phase- Validation consists of two main steps, after which the ledger gets updated. Those steps are the *endorsement policy evaluation* and *read-write conflict check.*
  Update state: Each peer appends the block to its corresponding channel's chain and the valid transaction's write sets are committed to the ledger and events are generated in order to notify the client application regarding whether the transaction has been valid and committed, or invalid and rejected.

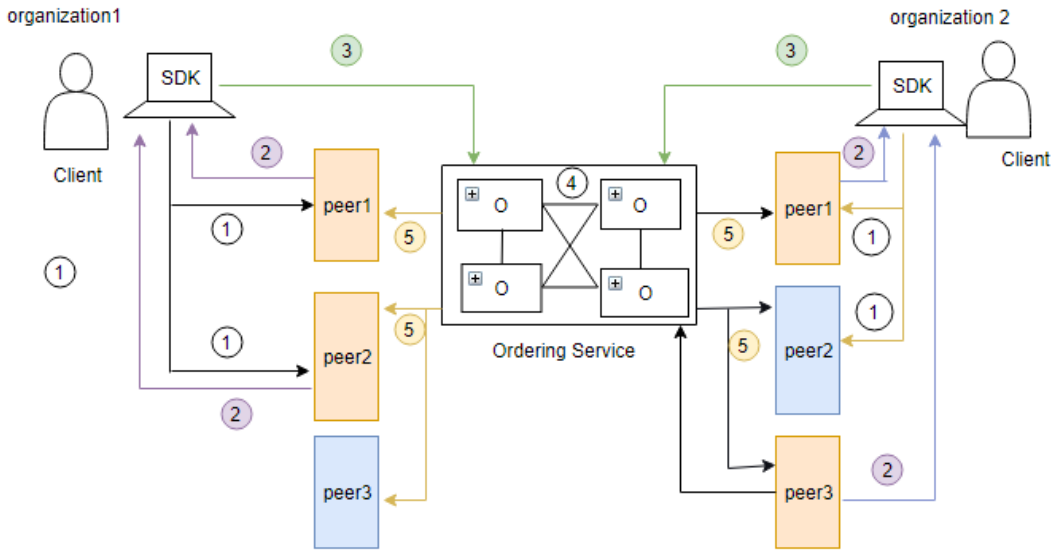
## V. FABRIC TRANSACTION FLOW

Consensus in hyperledger fabric is achieved through the transaction flow mechanism described in the following seven steps and depicted in fig.3 which excludes the last two steps, of validation and updation of ledger which happens in every peers in the network.

1) Propose Transaction:- Client applications would submit the transactions to a few peers(endorsing peers).
2) Execute proposed transaction:-These endorsing peers would execute the transactions and the outputs received are signed by corresponding endorser, so that all outputs of these transactions would be the same, for it to be deterministic. The endorsement response contains the cryptographic materials as well as the read-write set.
3) Proposal response:-Client application is required to collect endorsements from multiple endorsing peers, depending on the endorsement policy defined along with the smart contract.
4) Order transaction:-After collecting required endorsements, those transactions are given to the ordering service. The ordering service would receive transactions from multiple applications.

5) Deliver ordered transaction:-Ordering service ensures that all peers would see these transactions in the same order, and it would be known as a block. Thus the ordering service forms the block and communicates it to all peers. Ordering service can follow any ordering algorithms developed from the past 30 years of enriched literatures. The few ordering service algorithms implemented by hyperledger fabric are
    a. SOLO- A single node or a dictator decides what the order should be. The order could be FIFO for example. As the name indicates, it has only a single ordering node and thus doesn't offer fault tolerance. Thus SOLO is not recommended for production but could be used for test case applications.
    b. Kafka- It is a popular event management service by Apache and is open source[8]. Kafka offers notion of consensus that is crash fault tolerant (CFT). CFT can tolerate some ordering node failures. Even if few ordering nodes fails (typically less than 50%), it can still achieve consistent order across all the nodes. Like Raft, Kafka follows "leader and follower" model. The Kafka based ordering service is available since Fabric v1.0.

   The ordering service can perform access control permissions of the client nodes, to check whether the node has permission to broadcast or receive blocks on a particular channel. Once the block is send to all peer nodes through either ordering node or the gossip protocol (based on gRPC), the transactions must be validated.
6) Validate transaction:-Not every transactions in the block will be valid ones. Validation phase has to identify and reject the invalid transactions and only the valid ones will be committed and updated in the ledger and the world state(stores the recent state of blockchain ledger). Validation phase has two consecutive steps:
    a. Endorsement policy evaluation: Is the task of validation system chaincode(VSCC). VSCC is a part of the static library of the blockchain configuration in the genesis block (the very first block).
    b. Read-write conflict check: The read-write set provided by endorser will have a read set which contains list of unique keys along with its committed version number. The write set contains list of unique keys along with its new value written by the transaction during its execution with endorsing peer. The version of the read set is checked with the current version in the world state of the blockchain. If there is mismatch in versions, the transactions are marked invalid and its effects will be disregarded. The transactions that match the read set versions would be the valid ones.
7) Ledger update and notify Transaction- The valid transactions identified will be committed by all the peers in the blockchain and its write set would be updated in the world state. Every peer need to commit all transactions in the same order. The peers can emit events to notify the client applications. Events can be of multiple types,
    a. Block level events- Notifies the block of transactions that get committed.
    b. Transaction level events- Notifies the valid and invalid transactions within a block
    c. Chaincode events- To notify results of any business logic related events.



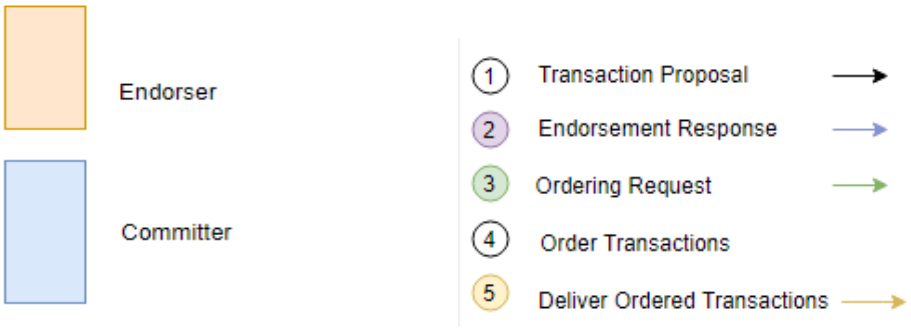



Fig. 3 Fabric transaction flow

Fabric Features

- Security and membership services: Good support through the Certification Authorities (CA's) for TLS certificates, enrolment certificates, and transaction certificates. Easy to trace back transactions and to verify its authenticity.
- Selected endorsers: Improves the efficiency and throughput.
- Pluggable consensus and chaincodes: improves scalability and performance.
- Channels: Provide privacy and multitenancy with in the blockchain network.

## VI. CONCLUSIONS

Hyperledger fabric developed by Linux Foundation, is a truly modular and pluggable, thus extensible distributed operating system to implement permissioned blockchains through its novel architecture of execute-order-validate. The new architecture enables to separate execution from consensus thus improving the performance and scalability of the permissioned network. The blockchain system related configurations will be provided along with the genesis block. The pluggable modules in fabric enables it to pave path for many future endeavors, as the fabric provides highly secure private channels and multi-party ownership towards transactions.

## REFERENCES

[1] Zibin Zheng1, Shaoan Xie1, Hongning Dai2, Xiangping Chen4, and Huaimin Wang3,"An Overview of Blockchain Technology:Architecture,Consensus, and Future Trends",IEEE 6th International Congress On Big Data, pp. 557-564
[2] Tinu N.S*1 , Soja Rani.S 2,"Blockchain Technology- An Extensive Study On Its Users And Applications", GJESR, vol. 5, Issue 6.2018
[3] C.Cachin, C. Ferris Hyperledger Fabric: A Distributed Oprerating system for Permissioned Blockchains,arxiv.org/pdf/1801, Apr 2018.
[4] F. B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. ACM Comput. Surv., 22(4):299–319, 1990J. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
[5] Hyperledger. http://www.hyperledger.org..
[6] A. S. Tanenbaum. Distributed operating systems anno 1992. what have we learned so far? Distributed Systems Engineering, 1(1):3–10, 1993.
[7] J. Camenisch and E. V. Herreweghen. Design and implementation of the idemix anonymous credential system. In ACM Conference on Computer and Communications Security (CCS), pages 21–30, 2002.
[8] Apache Kafka. http://kafka.apache.org.