I.J. Information Engineering and Electronic Business, **2021, 4, 30-46**
Published Online August 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2021.04.03

A Blockchain based Framework for Property Registration System in E-Governance

Siddhartha Sen
National Informatics Centre, 18 Rawdon Street, Kolkata-700017, India
Email: sen141@gmail.com

Sripati Mukhopadhyay
Academy of Technology, Adi Saptagram, Hooghly-712121, India
Email: dr.sripatim@gmail.com

Sunil Karforma
University of Burdwan, Golapbag, Burdwan-713104, India
Email: sunilkarforma@yahoo.com

Received: 18 April 2021; Revised: 01 May 2021; Accepted: 15 May 2021; Published: 08 August 2021

Abstract: In recent years, the most cutting edges and promising technology emerged is Blockchain. It has huge potential to impact various industries. The append-only distributed ledger technology (DLT) and the consensus mechanism of Blockchain can also change the dimension of E-Governance. The Electronic Property Record (EPR) systems of government have challenges like data security, integrity, secure storage of data and automated service delivery. In this paper, we discuss how Smart Contract based Blockchain technology can effectively be used to address the challenges of EPR System over the existing available systems. We propose a Smart Contract based permissioned blockchain framework which is an innovative approach, especially in the Electronic Property Registration domain of E-Governance in India. The objective of our proposed framework is firstly to implement Smart Contract solving the security problems like confidentiality, integrity, authentication, and secondly to ensure secure storage of electronic records by defining access rules for the stakeholders of the proposed framework. Moreover, we address the issues of single-point-of-failure, data inter-operability between the organizations involved for sharing and verification of property information among various stakeholders.

Index Terms: Blockchain, smart contract, transaction ledger, decentralization, framework, EPR

1. Introduction

With advancement of new technologies, people are experiencing better services and values in every aspect of life. Changes in technologies are providing much better options to solve problems and thus improving the existing systems. At the same time, technological advancements are also creating new threats in the security perspective. Over the time Property Registration System of government had improved quite a lot by moving to EPR system from the older manual system. Nowadays information is easily searched and services are delivered much faster. However, the system still has scope of improvement in the area of security, integrity, transparency and ownership of data. These issues may be better addressed by the use of emerging blockchain technology. This technology offers to provide a secure, temper-proof, transparent platform for storing records in de-centralized manner and delivering e-governance services better than the existing system. The ledger of a blockchain is immutable as every block header holds hash of previous block and Merkle root which is hash of all the hashes of all the transactions. As a result, any tampering of data in any block makes all the following blocks invalid. This immutable property makes the data tamper-proof in any blockchain.

In the existing traditional Property Registration systems, property registration process involves verification of the title of the property, estimation of property value, stamp duty and fees, payment of fees and stamp duty, digital signature of the deed, biometric data capture for buyer and seller and finally handing over of the deed. This requires involvement of buyer, seller, registration authority, Land Authority and/or municipal civic body. For registering the execution of any property, firstly the title of the property is required to be verified by registration authority and the land on which property is existing is required to be verified from Land Authority. Sometime verification is required from local municipal civic body. Secondly, it is required to be verified whether the seller is possessing the amount of property he or she sets up for sale or transfer. It is also required to be verified whether the seller is really selling the property or some fraud element is trying to sell the property. The system requires a strict check on whether the same property is being sold or transferred to multiple persons. Thirdly, payment is made for registration fee and stamp duty. Finally, property is debited from the seller and credited in the name of the buyer and transaction is executed by changing the ownership to the buyer and digitally signed deed is handed over to buyer. In the existing system, most the above-mentioned verification is done manually on the basis of submitted supportive documents. This may not rule out any case of submitted forged document. There is no mechanism available to verify documents digitally in trusted manner in the exiting EPR system. The system involves multiple parties and authorities to verify and trust each other which may be case in practical world. The existing manual system may take considerable time to truly verify the submitted documents and buyer or seller has to be dependent on the authorities to execute the transaction. Ideally, once verification is done and payment is completed, transaction should be executed immediately, which always do not happen in practice.

Electronic property registration system is implemented in many state governments in different forms. Different state government may have different rules, they may have a different setup involving different government agencies to provide the E-Governance services to the citizen. Despite providing E-Governance services to the citizens, the existing EPR systems faces some problems and unable to meet expectation associated with them. The problems of the existing EPR systems are as follows:

i. Multi-agency data verification and inter-operability: Multiple independent agencies like Registration Authority, Land Authority, Municipal Authority are involved in verification, approving and finalizing transaction on any property like selling, buying, ownership transferring. Land or Municipality related documents are submitted by a buyer or seller before Registration Authority for verification and processing further. The cost of verification of submitted physical documents are high and time taking matter as those agencies are independently built on different technology platform and there is no inter-operability of data. Often exchanges of data between them are a problem. A platform is required where each other's data are easily available and can be trusted without requiring further verification but at the same time data should not be public. Even agencies like bank, insurance company, mortgage agency seek data verification for sanctioning approval for issuing loan or insurance cover etc.

ii. Centralized data with single agency control: In existing conventional system, all data are stored centrally in agency's server. Registration Authority has all control over the whole EPR data. Any adversary within the agencies would be able to alter data and also the data is subject to single point of failure as stored centrally. No mechanism is employed to prevent any possible manipulation of existing transaction data. A system is required which can ensure that nobody can alter any data. There should be a system to ensure that all real time data are stored in different locations preventing single-point-of-failure.

iii. Data confidentiality and access control: Government data have become a target of cyber-attacks and an increasing trend has been witnessed. Always there are chances of compromise of the existing system and thus may even loose data. The Government data are required to be kept confidential and unauthorized entities should not be able to access the EPR data. Only authenticated users should be allowed to access data with clearly stated access privilege.

The problem of data verification can be done by using digital signature by the issuing agencies. But all agencies are always required to maintain an updated Certificate Revocation List. Any compromised DSC can destroy the whole system and still there are dependencies on trusted third party agency. The problem of inter-operability of data can be resolved using web service. All agencies required to employ web service to make their data consumable by other agencies. However, web services are also subject to cyber-attack. It requires a complete maneuver of the existing technology platform. In order to solve the single-point-of-failure, all agencies are required to deploy multiple redundant storage at various locations and provisioning synchronization of those data in real time basis. This is a very costly matter to implement and still it may end up with storing of non-updated data. And centralized data with control of a single agency are always vulnerable to tampering by internal adversaries and cannot be absolutely ruled out. Regarding confidentiality and access control of data, any agency is vulnerable to this problem even after implementing good security solution. There is always a need to maintain an updated security solution to keep data confidential.

These problems make it reasonable to find a citizen centric tamper proof resilient distributed redundant interoperable platform. Blockchain technology which is based on secured cryptographic one-way hash functions provides security, transparency, data integrity to the government records of the citizen. The consensus protocol of blockchain eliminates involvement of trusted third party for legitimacy of any transaction. The capability of Blockchain nodes on different technology platform to transact among themselves can solve the data inter-operability problem of EPR systems. In one hand, the immutable ledger of blockchain rules out all possibility of any data tampering even by any adversaries within and also solves the problem of single-point-of-failure due to its de-centralized platform. The membership services of blockchain solve the problem of confidentiality and access control. This paper proposes a framework that creates such a decentralized platform that would store citizen's property and land related records in different locations and give access of those records to owner of the data or concerned individual client citizen with facilitating confidentiality, integrity, inter-operability of data between stakeholders. Apart from solving the cited problems of EPR systems, our research work also evaluates the performance of blockchain implementation.

Contributions. The main contribution to this paper is to conceptualize a framework for creating a tamperproof secured e-Governance transaction platform where multiple agencies are involved in approving and verifying transaction. Smart Contract built for verifying key information from multiple agencies and executing the business logic over different fabric channels. EPR Blockchain is a framework aimed to address the challenges faced in existing available system and its related technology. The performance of the proposed blockchain framework is evaluated on the basis of on the basis of latency and throughput by experimenting single-host and multi-host deployment and it is established that the proposed model can handle the transaction load of existing property registration offices of India.

In this paper we intend to achieve information security, privacy and integrity using cryptographic techniques existing in blockchain technology during e-Governance transactions. Section-2, summarizes basics of Blockchain technology and Hyperledger. Section-3, discusses about the related work done on EPR domain. The proposed EPR Blockchain framework and processes involved are described in Section-4. Conclusions are drawn from the entire discussion and performance evaluation are mentioned in Section-5. References are listed at the last part of this paper.

2. Blockchain Technology and its effectiveness:

A. Blockchain

A blockchain can be defined as an immutable ledger for recording transactions in a verifiable and permanent way, maintained within a distributed network of mutually untrusting peers. Every peer maintains a copy of the ledger. The peers execute a consensus protocol to validate transactions, group them into blocks, and build a hash chain over the blocks. This process forms the ledger by ordering the transactions, as is necessary for consistency. Blockchain technology was introduced by Satoshi Nakamoto [1], for his popular work of digital currency or crypto-currency, i.e., bitcoin. Nakamoto used blockchain technology to solve the double spending problem of bitcoin but soon this novel technology was being used in many other applications.

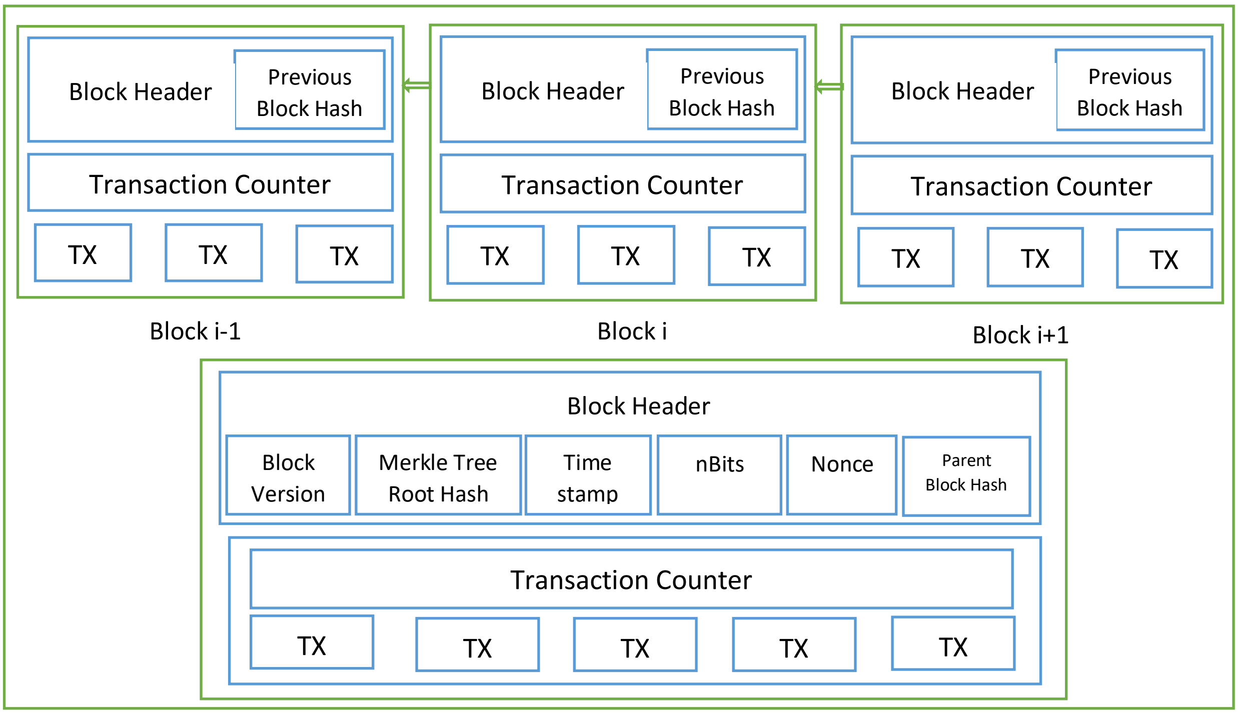

Fig. 1. Sequence of Blocks in Blockchain and Block Structure

Blockchain is a chain of blocks that are connected together and are continuously growing by storing transactions on the blocks. Legitimate data or transactions are permanently recorded in the blockchain, and the Merkle root [2,3] of the transaction can verify whether the transaction data in the block header and block has been tampered with. The hash value of the former block can be used to verify whether all blocks before the block and up to the Genesis block have been tampered with. Relying on the hash of the previous block, all blocks are interlinked. If any block is tampered, all subsequent block hash changes will be triggered. This platform uses a decentralized approach and managed by peer-to-peer networks. All peers hold a copy of the ledger. Blockchain can implement trusted transactions in untrusted distributed systems through cryptographic algorithms, times-tamps, and distributed consensus. A blockchain has certain benefits such as security, anonymity, and integrity of data with no third-party intervention. These benefits make it a reasonable choice to store citizen's property ownership records on it as records on blockchain are immutable. A number of researchers have also identified that using blockchain technology in similar sector would be a feasible solution [4,5].

Blockchain adopts the P2P protocol that can tolerate single point of failure. The consensus mechanism ensures a common, unambiguous ordering of transactions and blocks, and guarantees the integrity and consistency of the blockchain across geographically distributed nodes. By design, blockchain has such characteristics as decentralization, integrity, and auditability [6]. According to Xu et al. [7], blockchain can serve as a novel kind of software connector, which should be considered as a possible decentralized alternative to the existing centralized shared data storage.

As the blockchain technology continues to evolve with respect to the ways of how blockchains are constructed, accessed, and verified, they are being classified into three broad categories: (1)Public blockchain (such as Bitcoin and Ethereum) [18], which is open for anyone to read, send, or receive transactions, and allows any participant to join the consensus procedure of making the decision on which blocks contain correct transactions and get added to the blockchain. (2) Consortium blockchain (such as Hyperledger5 and Ripple), which has placed certain constraints on write permissions such that only a pre-selected set of participants in the network can influence and control the consensus process, even though read is open to any participant in the network, and (3) Private blockchain, whose write permissions are restricted strictly to a single participant (or organization), even though its read permissions are open to the public or constrained to a subset of participants in the network.

B. Architecture and Key Features of Blockchain

The blockchain platform can be seen as six layers, application layer, contract layer, incentive layer, consensus layer, network layer and data layer.

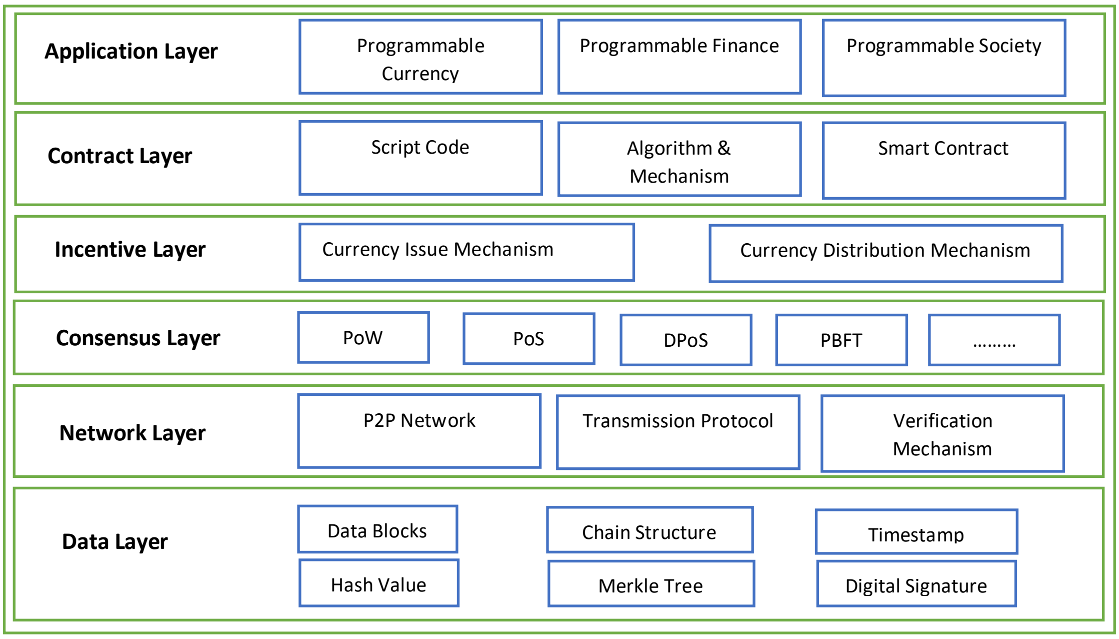

Fig. 2. Architecture of Blockchain

1) Single Point Failure Tolerance

In blockchain the data is distributed across the network to its peers rather than at one central point. The data is shared to all nodes connected on the network. The data that was before concentrated at one central point is now handled by many trusted entities. In case of crashing of any node, data would be available to other peers and it is tolerant to failure of any single node. Data can not be controlled by any single entity.

2) Data transparency

Every node or participant are part of the blockchain. Same version of data is available to all nodes and participant. All nodes can see the transaction data in the block making the system completely transparent. Trust is based on the cryptographic algorithm on which the blockchain technology is built.

3) Security and privacy

Blockchain technology is based on cryptographic functions to provide security to the nodes connected on its network. SHA-256 cryptographic algorithm is used to store hash of previous block in the block header. Cryptographic hashes are strong one-way functions that generate checksum for digital data and it can not be decoded. These cryptographic approaches make blockchain secured and its data private.

4) Immutable

Block header of every block of a blockchain holds hash of previous block and Merkle root which is hash of all the hashes of all the transactions. Thus, any tampering of data in any block makes all the following blocks invalid. This property of blockchain ensures data integrity and makes data in blockchain as immutable.

C. Smart Contract

The term “smart contract” was first coined in mid-1990s by computer scientist and cryptographer Szabo, who defined a smart contract as “a set of promises, specified in digital form, including protocols within which the parties perform on these promises” [8]. In his famous example, Szabo also expected that through clear logic, verification and enforcement of cryptographic protocols, smart contracts could be far more functional than their inanimate paper-based ancestors. However, the idea of smart contracts did not see the light till the emergence of blockchain technology, in which the public and append-only distributed

ledger technology (DLT) and the consensus mechanism make it possible to implement smart contract in its true sense.

Generally speaking, smart contracts can be defined as the computer protocols that digitally facilitate, verify, and enforce the contracts made between two or more parties on blockchain. As smart contracts are typically deployed on and secured by blockchain, they have some unique characteristics. First, the program code of a smart contract will be recorded and verified on blockchain, thus making the contract tamper-resistant. Second, the execution of a smart contract is enforced among anonymous, trustless individual nodes without centralized control, and coordination of third-party authorities. Third, a smart contract, like an intelligent agent, might have its own cryptocurrencies or other digital assets, and transfer them when predefined conditions are triggered [9].

3. Related Works

There is not so much research works found in the area of EPR. A. Shahnaz et al. [15] have already done some blockchain implementation in the field of Electronic Health Record. This can further be extended to other area of E-governance like EPR. At the very beginning property record were all maintained in physical paper system. Dale A. Whitman, Arthur R Gaudio [10,11] emphasized to have uniform electronic records for property data to manage it more efficiently and prevent fraud. Maksymilian Ewendt [12] examined the potential impact a blockchain-based real property system and possible methods of implementing such a system. It discussed the potential threat, advantage and feasibility to adopt blockchain technology but did not detail on technology implementation. Balaji S [13] proposes a Bitcoin based property ownership recording system. This model is based on public network which is not acceptable by government to store its sensitive data. Bitcoin requires significant amount computation and power for its mining which is not immensely available in the E-governance arena in India. Joshi S.M., Rajeswari K [14] proposed an Ethereum based system for transferring property. This also requires a public network and each Ethereum transaction has a cost to bear. The precondition of government data to keep the data and network safe. Keeping the requirement of privacy and the way government works, here we propose to use Hyperledger Fabric which is a permissioned blockchain. The transaction is executed by executing Smart Contract (Chaincode) and proposed to store the title deed document to IPFS (Inter Planetary File System) [15,17], which lets store large amounts of data and have the immutable, permanent link into blockchain ledger. It helps not to put the whole deed document in the ledger for transaction, rather only its timestamp is put on the chain securing privacy of the document.

4. Proposed Framework

The proposed framework would solve many problems in the domain of EPR. As EPR data mostly kept at various government department, we propose Hyperledger Fabric, a permissioned blockchain comprising of Registration Authority, Land Authority and Municipal Authority as peers and seller, buyer, mortgage agency, insurance agency as participants. As the EPR data are subject to privacy policy of government, public blockchain is avoided. We propose to place Smart Contract based solution with Kafka consensus to solve the problem. The seller and buyer are directly involved in participating property transactions like, adding new property, selling partial or complete property, transferring ownership of property, whereas Mortgage agencies, Insurance agencies are involved in querying information on property for title, ownership, property details to fulfill their requirement of verification for approving mortgage or insuring property. We designed two primary smart contacts, one for executing the business logic after verifying key information from Land Authority and Municipal Authority and second one to provide information verification services to other agencies. We also proposed to store the title deed document to IPFS after execution of smart contract and committing to ledger. We have also designed specific fabric channels to instantiate the smart contracts. The proposed model is directed to achieve information security, privacy and integrity using cryptographic techniques inherently existing in blockchain technology during e-Governance transactions. It also solves the problem of single point of failures. The performance analysis in our experiments suggests for deployment of multi-host smart contract for implementation in EPR systems.

4.1 Components and Architecture

We first introduce the three main layers of the framework as depicted in Fig.3.:

(1) Peers (Government agencies custodian of EPR data e.g. Registration Authority, Land Authority, Municipal Authority),
(2) Participants (agencies or individual that transact with government or seeks information, e.g. seller, buyer, Mortgage agency, Loan agency),
(3) Hyperledger Fabric Blockchain.

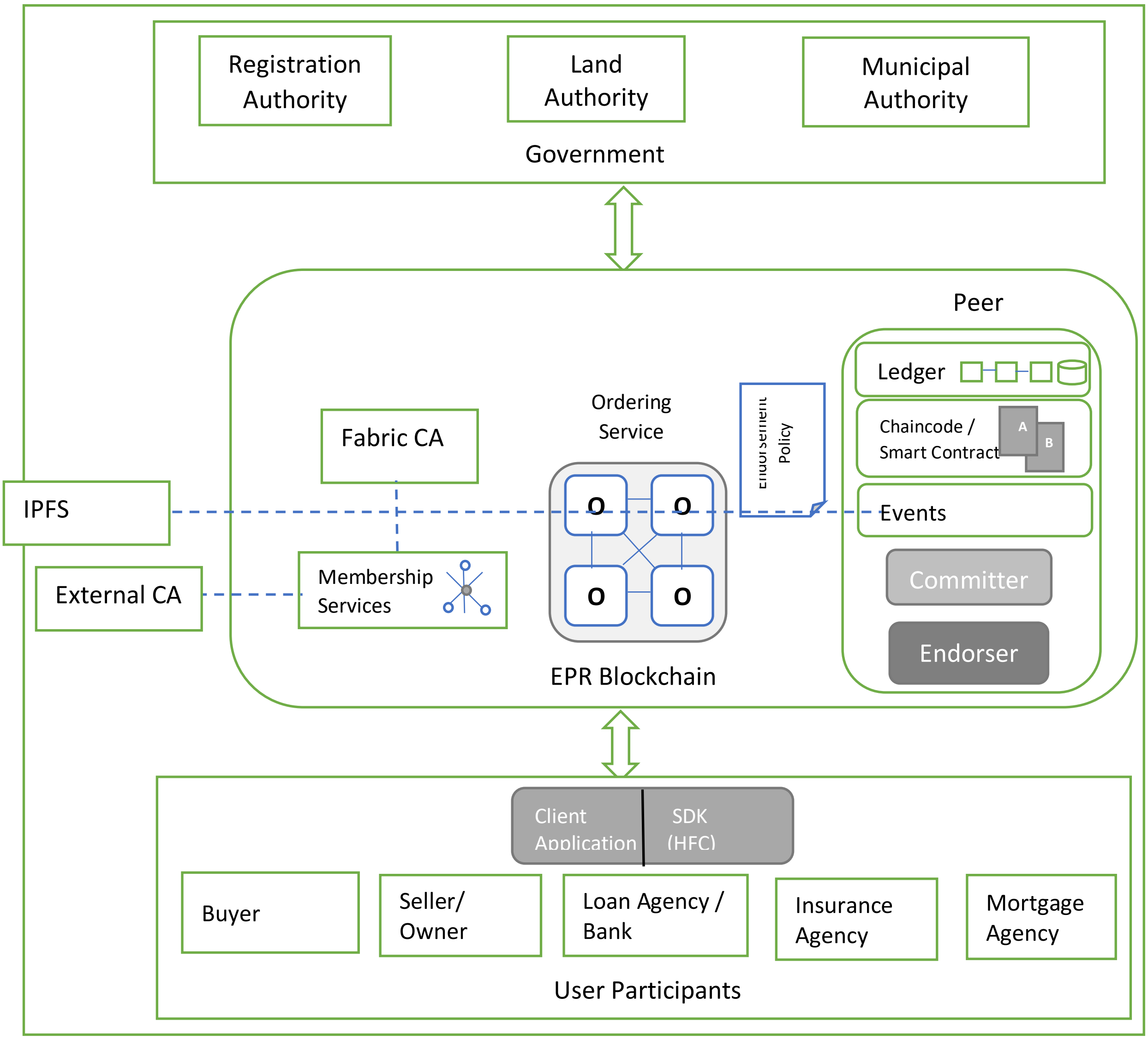

Fig. 3. EPR Blockchain: Framework Architecture

Government Agencies: Here Registration authority is providing the primary services of adding new property registration, transacting property, transferring ownership, issuing deed title, providing information on property details, ownership details. Land Authority is verifying the land details and land ownership details. Municipal authority is verifying the type of property and its status, e.g. in case of registering an apartment, it would verify whether the apartment is complete and ready for registration.

User Participants: Any seller, buyer are direct user participants in the proposed framework. Sometimes the buyer or seller may not be the actual owner of property but may be authorised representative of the actual owner. Other agencies like Loan agencies/banks, Insurance agencies, Mortgage agencies are also user participant as they verify client information on ownership of property, type of property etc for approving loan, insuring property and approving mortgage.

EPR Blockchain: It is permissioned blockchain and restrict the participation in the network. Thus, fulfils government's requirement to have control over the participation the blockchain network. But the control of data is not with any single agencies. Each peer is holding current set of data in their ledger. EPR blockchain is made up of Hyperledger Fabric which executes Smart Contracts (Chaincode) to hold the transaction data of property and is made available to respective users. Title deed is also issued to owners of property by executing specific Chaincode. Chaincode holds business logic to be executed to provide the service.

IPFS: The Inter Planetary File System is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices. As opposed to a centrally located server, IPFS is built around a decentralized system of user-operators who hold a portion of the overall data, creating a resilient system of file storage and sharing. All the data stored on IPFS contains a cryptographically generated hash value. This hash is unique and is used for identification of stored data on it. The data stored in IPFS is separated from transaction and it efficiently reduce communication overhead and computation overhead while ensuring privacy of preserving [16]. The secure storage system under IPFS protocol makes it a favourable choice for storing title deed document.

4.2 Transaction Flow

Here we have three Endorsing Peers, Registration Authority, Land Authority, Municipal Authority and our committing peer is Registration Authority itself is committing and Anchoring Peer. But the role of Registration Authority is different from Land Authority and Municipal Authority. Seller, buyer, loan agency, insurance agency and mortgage agency are the client user-participant. Transaction data is comprising of Client Id, Chaincode Id, payload, timestamp, signature of client (trans< Client Id, Chaincode Id, payload, timestamp, signature >). Payload are the details of transaction data in json data format.

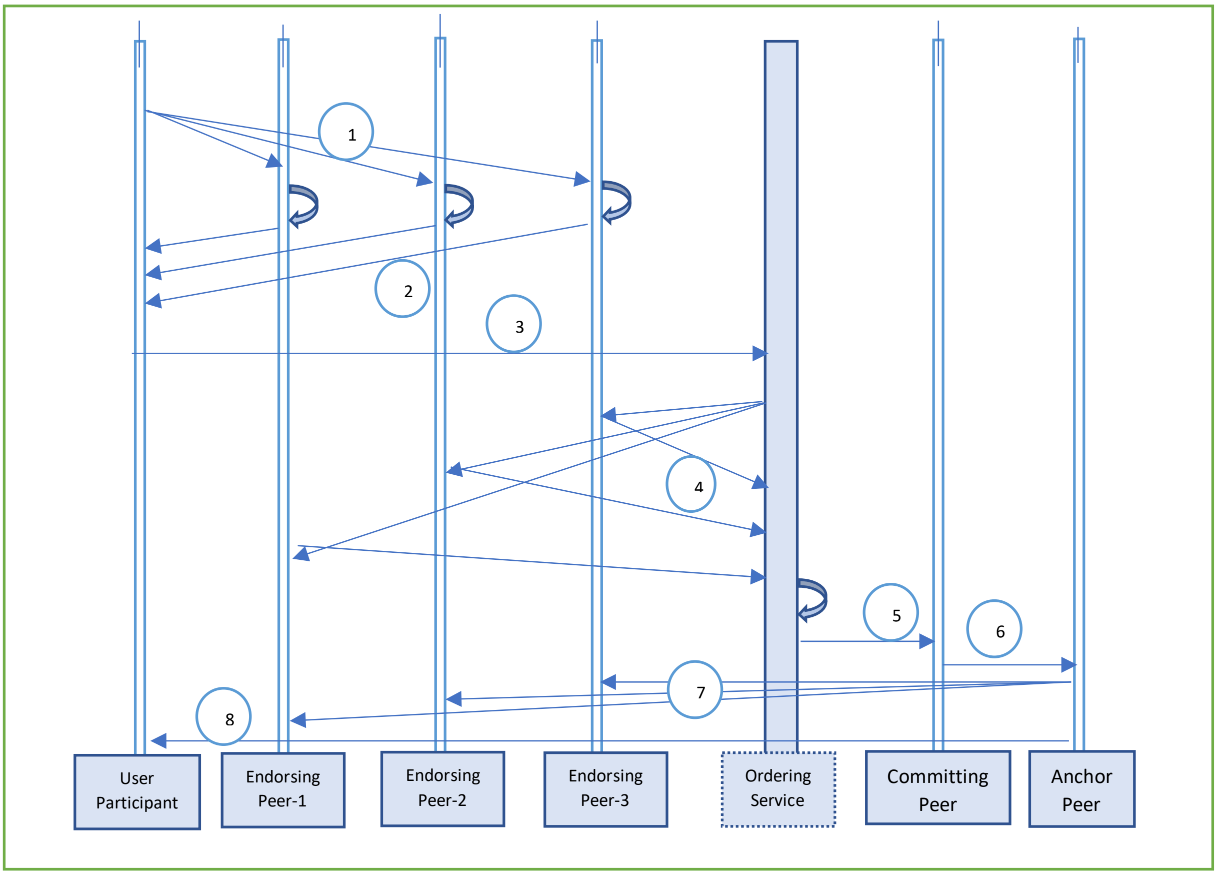

Fig. 4. Transaction Flow

(1) Client application sends signed transaction to all endorsing peers. (2) Endorsing peers checks the identity and permissions of client. Then executes Chaincode to perform simulated transaction and sends back the signed read-write set to the client. (3) Client collects all endorsed transactions, validates with endorsement policy and sends to ordering service. (4) Ordering service verifies all endorsements, read-write set and creates new blocks with transaction data, updates world state and sends to committing peer. (5) Ordering service sends to committing peer. (6) Committing peer checks the transaction & commits new block and sends to anchoring block to broadcast. (7) Anchoring peer broadcasts the updated copy of ledger to all peers to have same version of ledger. (8) Transaction notification is sent to client.

4.3 Activities for Setting up Hyperledger Fabric for transaction:

Following are the preliminaries that are to be completed to set up the hyper ledger fabric and after that transaction can be initiated:

A. Setting up of communication channel

First the Network channels for the fabric is required to be defined. Different Smart Contracts are instantiated on different channel. Here we proposed to have two separate channels.

Channel-1: The first channel comprises of the seller, buyer, Registration Authority, Land Authority and Municipal Authority. The smart contract running on this channel can add new property, buy/sell existing property, transfer ownership.

Channel-2: The second channel is comprised of owner of property, Registering Authority, Insurance Agency, Mortgage Agency, Loan agency. These agencies usually verify ownership, details of property etc but do not participate in any transaction. The smart contract running on this channel will execute different queries with the authorization of the owner of the property.

Channel-3: The third channel is comprising of all three government agencies, Registration Authority, Land Authority and Municipal Authority. Any transaction carried out between them would be transparent to all three organizations only and others can not see those.

Channel policies are defined in configtx.yaml, which we pass as an input during the creation of the channel.

B. Setting up Membership Services

A Membership Service Provider (MSP) manages a set of identities within distributed network. It provides identity for Peers and orderers, participant clients, administrators. Further these identities can be issued by Fabric-CA or by an external CA. MSP issues enrolment certificate which will be used to make future transactions by the users (participants, peers, orderers) in the fabric network. So, MSP provides authentication, validation, signing and issuance. It supports various crypto standards with pluggable interface. A network can have multiple MSPs. Here Registration Authority is running the channel MSP also. It determines which peers or orderer can join which channel. It also regulates the read-write permission of any participant to the channel. It identifies authorized administrators from other peers. It stores public CA certificate and certificate revocation list (CRL) so check the authenticity of the other peer’s users. The channel MSP also provides channel information to all peers and orderers on which channel they are connected to. Land Dept and Municipal Authority is having their own MSP all include their own users as per requirement of their hierarchy. All the communication between peers, participant, orderer etc are secured through TLS encryption.

C. Managing Identities

Here all three government agencies have set of peers from each organization onboard as required fulfilment of their hierarchy. The concerned MSP of the participating organization would manage their identities at their level. Every user in the fabric network is going to be issued an identity and enrolment certificate. Enrolment certificate has private key and signed-certificate. The private key is private to the user and would be used to digitally sign the transaction. The private key is stored in secure KeyStore. Signed-certificate is public x.509. It includes the public key of the user. Everyone in the fabric can verify the identity of the user through this.

D. Setting up of Endorsement policy

Endorsement Policy tells about the list of endorsing peers and about the rule for a number of endorsers required to achieve consensuses. Each endorsing peer has a Chaincode installed and has a copy of ledger which is sync between all peers. Every Chaincode has an endorsement policy which specifies the set of peers on a channel that must execute Chaincode and endorse the execution results in order for the transaction to be considered valid. Here our endorsement policy is that the peers, Registration Authority, Land Authority and Municipal Authority are required to sign the endorsement and they all are supposed to run Smart Contract1 which is built in such a manner that they would run different methods as per requirement of business logic. Land and Municipal Authority would run methods related to verification of land details and construction permission & completion related methods.

E. Deployment of Smart Contract (Chaincode) to all peers

Once the Channel policy, Membership policy, Endorsement policy are complete, Smart contracts (Chaincode) are instantiated over respective channel as applicable and deployed to all peers as per endorsement policy. All peers, Registration Authority, Land Authority and Municipal Authority possess same version of Chaincode.

F. Algorithm for New User Registration and Enrolment to Fabric

Every peer organization require to add the peer nodes with respect to their organization. Here all three government agencies are required to add new users from their organization to satisfy their hierarchy.

Step 1. Any user wants to join the fabric network sends request to the administrator of his/her organization.
Step 2. The administrator registers a new enrolment id to the fabric-CA.
Step 3. The fabric-CA returns some secret to the administrator.
Step 4. Administrator sends the enrolment id and the secret to the user.
Step 5. The user contacts the fabric-CA with enrolment id and secret as received from administrator to obtain an enrolment certificate.
Step 6. Administrator sends fabric enrolment certificate which will be used to make future transactions by the user in the fabric network.

G. The Steps of the Transaction Processing System:

Step1: Client Initiates Transaction

Using Hyperledger Fabric SDK client, the property owner sends transaction proposal in the format trans< Enrolment Id, Chaincode Id, payload, timestamp, signature > to the Registration Authority peer and it is replicated to all other endorsing peers of the network. Our Endorsement Policy require endorsement from all the three peers, Registration, Land and Municipal Authorities to proceed further. Once this policy is satisfied, smart contract would execute.

Step2: Endorsing peers verify signature & execute the transaction

The endorsing peers verify (a) that the transaction proposal is well formed, (b) it has not been submitted already in the past (replay-attack protection), (c) the signature is valid (using the MSP), and (d) that the submitter is properly authorized to perform the proposed operation on that channel.

Each endorsing peer takes the transaction proposal inputs as arguments to invoke the Chaincode function. Chaincode is then executed against the current state of the database to produce transaction results including response value, along with read and write sets, called RW Sets. These RW sets capture what was read from the current world state while simulating the transaction, as well as what would have been written to the world state had the transaction been executed. No updates are made in a ledger at this point. The set of these values, along with the endorsing peer's signature is passed back as a "proposal response" to the SDK which parses the payload for the application to consume.

Step3: Proposal response are inspected

Client receives signed endorsement from all endorsing peer. After receiving the list of endorsements, this list is validated based on the rule set in the endorsement policy and checks if it is satisfied or not. If it requires add/update operation on the ledger, the client checks whether the specified endorsement policy has been fulfilled and then only it prepares it to send to Ordering Service.

Step4: Client assembles endorsements into a transaction

Once all Endorsement responses are validated and found suitable to deliver it to the Ordering service for creation of new block, SDK broadcasts transaction message to ordering services. Among available ordering services, Kafka consensus is recommended as it enables fault tolerance and also provides high-throughput, low-latency for real-time data feeds. The transaction message sent to ordering services contains read/write sets, endorsing peer signature and channel ID. The Ordering service receives all transaction messages through the channels as applicable (here Channel1) and orders them chronologically and creates a block of transactions per channel. The ordering service accepts the endorsed transactions and specifies the order in which those transactions will be committed to the ledger.

Step5: Disseminate the block to leader peers

Once the block is formed by arraigning the transactions in chronological order, the block is now ready to send to all leader peers in the network. Here Registration authority identifies one leader peer which ensures to propagate the block to all peers within that organization. Leader peers also disseminate the block to all committing peers. The block is delivered to all peers on the channel using gossip protocol. The transactions within the block are validated to ensure endorsement policy is fulfilled.

Step6: Ledger update

Committing Peers checks if the endorsement is valid according to the policy of the chaincode and also verifies the dependencies (RW sets) are not been violated. The committing peer validates the transaction by making sure that the RW sets still match the current world state. Specifically, that the Read data that existed when the endorsers simulated the transaction is identical to the current world state. When the committing peer validates the transaction, the transaction is written to the ledger, and the world state is updated with the Write data from the RW Set and each peer appends the block to the channel's blockchain. The created title deed document permanently committed to IPFS.

Step7: Client Notification

Lastly, an event is emitted by each peer to notify the client application that the transaction (invocation) has been immutably appended to the chain, as well as notification of whether the transaction was validated or invalidated. The committing peers asynchronously notify the client application i.e. the property owner of the success or failure of the transaction. The same is notified to all the peers as well.

4.4 Proposed Smart Contract

In the proposed framework we have identified two main smart contracts, one for property registration transaction. This describes us the scenario, when a seller is trying to sole his/her existing property and registering the transaction in the records of Registration authority. Second one describes the scenario when Loan approving authority or Mortgage authority or Insurance authority is seeking information for verification for their internal approval process. The algorithm for the two mentioned are described below:

A. Smart Contracts

```
--------------------------------------------------------------------
Smart Contract Algorithm: 1 for Property Registration Transaction
--------------------------------------------------------------------
Authentication and access privilege of participant:

function authenticate (enrolment-id, signature)
{
if(enrolment-id-exists=False) then
abort request;
end if
if (signature_valid=False) then
abort request;
end if
}
end function

function access_privilege(enrolment-id, chaincode-id)
{
if (enrolment-id having permission to invoke-chaincode=False) then REJECT
end if
}
end function

Checking format and duplicate Transaction:

function check_proposal_format(transaction-proposal)
{
Check_whether_proposal_is_in_acceptable_format;
}
end function

function check_duplicate_transaction(transaction-proposal)
{
Check_whether_transaction_already_submitted_in_past;
```

```
}
end function

Business Logic:

function transaction_approval(owner-information, subjected-property-
details, property-completion-details)
{

if (peer-Is-Land-Authority) then
{ if(owner-and-land-details-is-correct=True) then
  transaction-approved-by-Land-Department=True
  update world state (Key value: Land-Department-
  Validation=True)
  else
  REJECT
  end if
}
else REJECT
end if

if (peer-Is-Municipal-Authority) then
{
  if(property-owner-details-is-correct==False) then
  REJECT
  else
  if (Construction-plan-sanctioned==False) then
  REJECT
  else
  if (property-construction-completion-certificate-issued==True)
  then
  REJECT
  else
  transaction-approved-by-Municipal-Authority=True;
  update world state; (Key value: Municipal-Authority-
  Validation=True)
  end if
  end if
  end if
  }
else REJECT;
end if

if (peer-Is-Registration-Authority) then
{
  if(owner==FALSE) then REJECT
  else
  if(transact-property-amount>owner-property-amount)
  then REJECT
  else
  if(property-valuation==FALSE) then REJECT
  else
  if(fee-paid=True || fee-paid-amount=evaluated-fee)
  then transaction-approved-by-Registration-
  Authority=True
  update world state; (Key value: Registration-
  Authority-
  Validation=True, Fee-paid=True, owner=new-owner)
  Add transaction to the ledger;
  else
  REJECT;
  end if
  end if
  end if
}
else REJECT
end if
}
end function

function Final-Title-Deed-Execution (transaction-id, world state)
{
if (peer-Is-Registration-Authority) then
{
  if key-value (Land-Authority-Validation=True , Municipal-
  Authority-Validation=True, Registration-Authority-Validation=True,
  Fee-paid=True, owner-new-owner) then
  {
   Issue-title-deed and store to IPFS;
   IPFS-address=location of title-deed on IPFS;
   Calculate hash_deed=hash(Issued-title-deed);
   Update world state (key-value: hash_title_deed =
   hash_deed, IPFS_location = IPFS-address);
  }
  else reject
}
else REJECT
end if
}
end function



--------------------------------------------------------------------
Smart Contract Algorithm: 2 for Information Verification
--------------------------------------------------------------------

function authenticate (enrolment-id, signature)
{
if(enrolment-id-exists=False) then
abort request;
end if
if (signature_valid=False) then
abort request;
end if
}
end function

function access_privilege(enrolment-id, chaincode-id)
{
if (enrolment-id having permission to invoke-chaincode=False) then REJECT;
end if
```

```
}

end function

Checking format and duplicate Transaction:

function check_proposal_format(transaction-proposal)
{
Check_whether_proposal_is_in_acceptable_format;
}
end function


function verify_property_details (owner_details, property_details,
owner_consent, information_seeker)
{
if (peer-Is-Registration-Authority) then
{
  if (owner_consent==FALSE) then REJECT
  else
  if (verification_fees_paid=True) then
  provide-information-on-owner-and-property-details;
  update world state (key-value: information-provided-
  to =information_seeker);
  else
  REJECT
  end if
  end if
}
else REJECT
end if
}
end function
```

4.5 Implementation and Evaluation

4.5.1 Deployment

In order to implement the proposed framework and analysis, we built a dummy miniature simplest form of prototype with minimum configuration using some dummy test data. We deployed fabric on sign-host and multi-host environment. In our prototype implementation, we preferred a little bit older version to have stable versions of OS, software, tools etc. Following are the technical specifications we used to deploy the prototype:

1. Hyperledger Fabric: v1.4
2. OS- Ubuntu v16.04.6 LTS

Pre-requisite for Hyperledger Fabric:

3. SDK- Node.js- v12.3.1
4. cURL- v7.47.0
5. Docker- v19.03.0
6. Docker-compose- v1.24.0
7. Go- v1.12
8. NPM- v6.10.3
9. Python-2.7

The Peer, CouchDB, ordering service is packaged into Docker images. We considered two depts, Registration Authority and Land Authority in our experiment and configured the file configtx.yaml accordingly. Ordering Mechanism: SOLO. Only the first chaincode CC1 (for property transaction) is deployed which is executed by both the organization. Depending on their role, respective part would be run. The Hyperledger Fabric SDK allows applications to interact with a Fabric blockchain network. It provides a simple API to submit transactions to a ledger. Following are the peer setup:

Table 1.

Depts	CA	Membership Service	Peers	Port	State (CouchDB) Port
Orderer	--	--	orderer1.ra	10020	--
Registration Authority	CA-RA	MSP-RA	peer1.ra (anchor)	10021	5984
			peer2.ra	10022	6984
			peer3.ra	10023	7984
Land Authority	CA-LA	MSP-LA	peer1.la (anchor)	10024	8984
			peer2.la	10025	9984
			peer3.la	10026	10084

Other Configurations:
Ordering Service: SOLO
REST Server: Express.JS
SDK: Node.JS
Channel: Single

Summary of the data-fields used for asset value in state DB:

Table 2.

Sl. No.	Field
1	Application_No
2	Transaction_date_time
3	Land_details {district, police_stn, mouza, rs_plot_no, land_area, owner_name, last_transaction_date_time}
4	Property_details {district, police_stn, mouza, rs_plot_no, property_type, property_area, owner_name, property_valuation, stamp_duty_amount, stamp_duty_paid_status, last_transaction_date_time }
5	Transaction_status

First, we deployed Hyperledger Fabric in single-host machine. Later we deployed in multi-host environment to compare their performances. In fabric application, isolated docker based containers used to run major components of the network. Docker swarm is used to make communication between containers that are hosted in different hosts. For multi-host setup we have used 11 virtual machines (VM) having Ubuntu 16.04.06 LTS as OS and 4 GB RAM, for one CA, one MSP, three peers with state DB each for the two departments. VM-1 is initialized as docker swarm cluster manager node and rests of the VMS joined as worker node.

The configurations for multi-host experiment are as follows:

Table 3.

Hostname	IP Address	Peer/Service	Port
VM-1	192.168.1.10	Ordering service	7050
VM-2	192.168.1.11	CA-RA	7051
VM-3	192.168.1.12	CA-LA	7051
VM-4	192.168.1.13	MSP-RA	7052
VM-5	192.168.1.14	MSP-LA	7052
VM-6	192.168.1.15	peer1.ra & CouchDB	7053 & 5984
VM-7	192.168.1.16	peer2.ra & CouchDB	7053 & 5984
VM-8	192.168.1.17	peer3.ra & CouchDB	7053 & 5984
VM-9	192.168.1.18	peer1.la & CouchDB	7053 & 5984
VM-10	192.168.1.19	peer2.la & CouchDB	7053 & 5984
VM-11	192.168.1.20	peer3.la & CouchDB	7053 & 5984

4.5.2 Evaluation and Results

The performances are measured using Hyperledger Caliper tool which is a benchmarking tool as used in different Hyperledger frameworks. We measured and analyzed two parameters for measuring performances, (1) Latency and (2) Throughput. These are two commonly used parameters to measure performances in blockchain implementations.

Latency : In terms of blockchain technology, any transaction is said to be confirmed only when it is included in the ledger. Latency of a transaction can be defined as the time taken by a transaction to be added into ledger from the time of submission to the network.

$$Latency = \text{Confirmation Time} - \text{Submission Time} \tag{1}$$

$$Avg\ Latency = \frac{\sum \text{Latency / no.of transaction}}{\text{Number of transactions}} \tag{2}$$

Throughput : This can be defined as the rate of transactions that are committed into the blockchain network. Here only valid transactions are considered. The transactions which are verified correctly without any error are considered as valid. If any error is found in verification, the transaction is considered to be invalid.

$$Throughput = \frac{\text{Valid Transactions Committed}}{\text{Total Time Taken}} \tag{3}$$

Configuration Table 2. is the basis of all results and evaluation herewith. The system is evaluated on single-host (Table 1.) and multi-host (Table 2.) environment with same number of peers in both cases. The results of evaluation metrics of both the environments are compared.

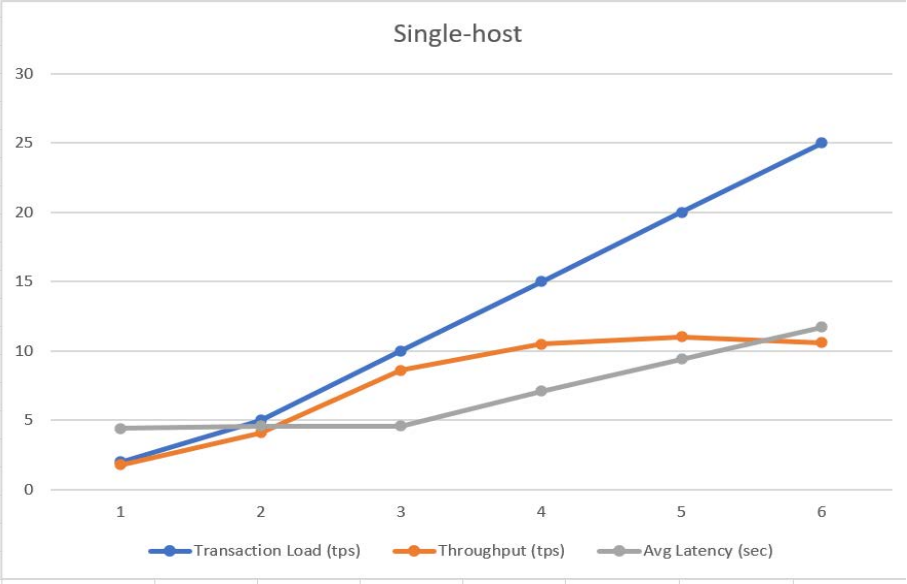

Fig. 5.

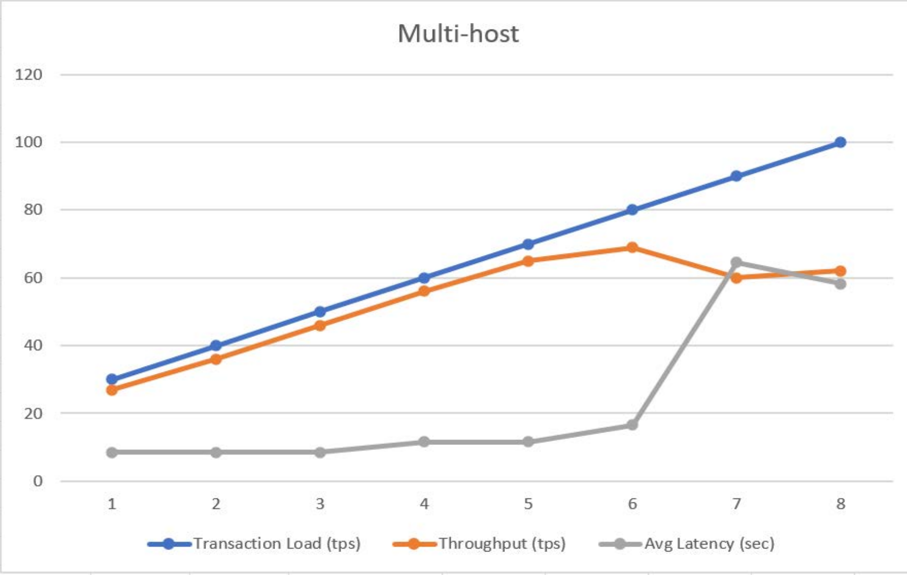

Fig. 6.

Throughput Observation:

1. It is observed that with the increase in transaction load the throughput also increased linearly up to a certain level which is maximum. If transaction load is increased further, throughput dropped but remain near optimal point. This pattern is more or less same for single-host and multi-host.

2. The optimal throughput in multi-host is much higher than single-host architecture. This may be due to the fact that the peers are evenly distributed over the network in multi-host environment.

Latency Observation:

1. It is observed that with lesser transaction load than maximum value, the average latency is significantly lesser. This is due to less waiting time for the transactions. It is also observed that average latency significantly increases, when the transaction load crosses the maximum throughput and the latency gets bigger with the increase of transaction.

It is found that throughput is optimum and latency is moderate around transaction load of 15-20 tps in single-host deployment and throughput is optimum and latency is minimum around transaction load of 60-70 tps in multi-host deployment. In most of the property registration offices, even in the peak hours, it is found that transaction load is less 10 tps. So, we can infer that the proposed model, even in single-host deployment is efficient enough for practical implementation to tolerate the load. Multi-host deployment can handle much bigger implementations.

5. Conclusion

In the proposed framework, the client SDK is built on Node.js and the Chaincode (Smart Contract) is written Go language, which are Open Source. The framework integrates different heterogeneous platform of Registration Authority, Land Authority and Municipal Authority.

In our simulation it is observed that even the single-host deployment of the proposed model can handle transaction load of any registration office of India. However, in scaling up the implementation, multi-host deployment performs significantly better than single-host deployment for the Property Registration System. If transaction load is increased beyond the optimal point, throughput dropped but remains near optimal point for both deployments and it can handle the load. But it is also observed that average latency significantly increases for single-host deployment, when the transaction load crosses the maximum throughput and the latency gets bigger with the increase of transaction. But when the load is around 10 tps, the performance single-host system is still acceptable. For very high transaction load i.e. transactions greater than 20 tps, multi-host deployment is required be adopted. It justifies that the proposed model is good enough to handle varied amount of transaction load. The blockchain technology is providing decentralization of data and making it tolerant to failure of any single node. The immutability property of blockchain ledger provides solid trust between all stakeholders and participant. The smart contract is providing edge to deliver the services basing on consensus among the approving agencies. This blockchain platform is also solving the double spending problem and thus preventing sell of a same property to many buyers. As the whole blockchain technology is built on strong cryptographic technology, the system ensures security, privacy and integrity to the EPR ecosystem. This platform is also very beneficial to the participants like banks, insurance agencies and mortgage agencies. A lot of revenue and time can be saved by these agencies in connection with verification of documents. This can prevent fraudulent issuing of loan, insurance or mortgage. It is also creating immense scope for these agencies to use the output of this system to integrate their own system. There is still having scopes to make the Membership Service more flexible. Identity management part can also be made more convenient like using Aadhaar based authentication system for larger scalability. As a whole, it is practical to implement the framework, as it does not require any major changes in the existing e-Governance ecosystem. In future, many other government and other agencies can be made onboard to the framework to mutually benefit each other.

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, J. Gen. Philosophy Sci., vol. 39, no. 1, pp. 53-67, 2008.

[2] R. C. Merkle, "Protocols for public key cryptosystems", in Proc. DBLP, Oakland, CA, USA, Apr. 1980, pp. 122-134.

[3] M. Szydlo, "Merkle tree traversal in log space and time," in Advances in Cryptology EUROCRYPT. Berlin, Germany: Springer, 2004, pp.541-554.

[4] W. Yang et al., “Survey on Blockchain-Based Internet Service Architecture: Requirements, Challenges, Trends, and Future”, IEEE, Access, vol.7,2019, pp.75845-74872.

[5] E. Chukwu, L. Garg, “Systematic Review of Blockchain in Healthcare: Frameworks, Prototypes, and Implementations”, IEEE, Access, vol.8,2020, pp.21196-21214.

[6] Y. Yuan and F.-Y. Wang, “Blockchain and cryptocurrencies: Model, techniques, and applications,” IEEE Trans. Syst., Man, Cybern, Syst., vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[7] X. Xu et al., “The blockchain as a software connector,” in Proc. 13th Working IEEE/IFIP Conf. Softw. Archit. (WICSA), 2016, pp. 182–191.

[8] N. Szabo. (1996), “Smart Contracts: Building Blocks for Digital Markets”. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[9] J. Stark, “Making Sense of Blockchain Smart Contracts”, 2016. [Online]. Available: https://www.coindesk.com/making-sense-smart-contracts/

[10] Dale A. Whitman, “Are We There Yet? The Case for A Uniform Electronic Recording Act”, 2002, 24 W. New Eng. L. Rev. 245, 246.

[11] Dean Arthur R Gaudio, “Electronic Real Estate Records: A Model for Action”,2002, 24 W. New Eng. L. Rev. 271.

[12] Maksymilian Ewendt, “Leveraging Blockchain Technology in Property Records: Establishing Trust in a Risk-Filled Market” 2018, 19 N.C. J.L. & Tech. 99

[13] Balaji S ,”BlockChain based Secure Smart Property Registration Management System and Smart Property Cards”, IJRASET, Volume 7 Issue VI, June 2019

[14] Joshi S.M., Rajeswari K. (2020) “Efficient and Accurate Property Title Retrieval Using Ethereum Blockchain” In: Karrupusamy P., Chen J., Shi Y. (eds) Sustainable Communication Networks and Application. ICSCN 2019. Lecture Notes on Data Engineering and Communications Technologies, vol 39. Springer, Cham.

[15] A. Shahnaz, U.Qamar, A.Khadid, "Using Blockchain for EHRs", IEEE, Access, vol.7,2019, pp.147782-147794.

[16] Jie Xu, Kaiping Xue, Shaohua Li, Hangyu Tian, Jianan Hong, Peilin Hong, Nenghai Yu, “Healthchain: A Blockchain-Based Privacy Preserving Scheme for Large-Scale Health Data”, IEEE Internet of Things Journal, Vol. 6, No. 5, October 2019, pp.8770-8781.

[17] Senny Hapiffah, Ardiles Sinaga, “Analysis of Blockchain Technology Recommendations to be Applied to Medical Record Data Storage Applications in Indonesia”, International Journal of Information Engineering and Electronic Business, Vol.12, No.6, pp.13-27, 2020.

[18] Sidra Anwar, Sadia Anayat, Sheeza Butt, Saher Butt, Muhammad Saad, “Generation Analysis of Blockchain Technology: Bitcoin and Ethereum”, International Journal of Information Engineering and Electronic Business, Vol.12, No.4, pp. 30-39, 2020.

Authors’ Profiles

Siddhartha Sen is working as Scientist in National Informatics Centre, Ministry of Electronics and Information Technology, Government of India. He is working in the field of Information Security, e-Governance and Cryptography for Ph D degree under the supervision of Prof. Sripati Mukhopadhyay, former Professor and Head of the Department of Computer Science, University of Burdwan, West Bengal, India.

Sripati Mukhopadhyay, M Tech., Ph D, is is a Senior Professor of the Department of Computer Science & Engineering, Academy of Technology, Adi Saptagram, Hooghly, West Bengal, India. He has served Burdwan University, Indian School of Mines, Visva-Bharati University, North Bengal University, Rabindra Bharati University, etc. He has 33 years of teaching and research experience. His research interests include Information Security, Artificial Intelligence & Data Mining.

Sunil Karforma, M.E., Ph D, is holding the post of the head of the Department of Computer Science, University of Burdwan, West Bengal, India. Network Security, e-Commerce, e-Learning, e-Governance and cryptography are the fields of interest in research area.

How to cite this paper: Siddhartha Sen, Sripati Mukhopadhyay, Sunil Karforma, " A Blockchain based Framework for Property Registration System in E-Governance", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.13, No.4, pp. 30-46, 2021. DOI: 10.5815/ijieeb.2021.04.03