Blockchain-Based System for Transaction and Ownership of Land Assets

Abstract—**India is one of the world's largest and most lucrative land markets. Registration of land assets through official government channels is a prerequisite for transacting in real estate markets. Traditional land registry systems face multiple issues, such as reliance on physical documentation, data mutability, bureaucratic inefficiencies, and widespread mishandling. Traditional approaches such as Databases, are not capable of providing verifiability along with immutability and non-repudiation. Since existing proposals fail to address these complexities, Blockchain-based approaches have emerged as a potential solution. Some blockchain approaches have tried solving these issues unsuccessfully. Addressing these limitations, we propose a blockchain-based system for the transaction and ownership of land assets. It leverages an Ethereum-based permissioned blockchain for the registration of land assets. Our model enables users to create ownership based on physical documents via registration, which is transferred among users based on a set of pre-defined rules. These transactions are secured using asymmetric key cryptography. All transactions use digital signatures to ensure transactions are generated by the authorized entity. As proof of ownership, Non-Fungible Tokens (NFTs) are generated, and the owner's public key can be used to verify that it is assigned to them. The proposed system is lightweight, efficient and effectively utilizes on-chain memory by employing distributed cloud storage for storing NFTs. It is immutable, transparent, and secure, which can be integrated into traditional land registration systems.**

Index Terms—**Blockchain, ERC721 Token, IPFS, JSON, Land Asset Files(LAFs), Transaction Request Files(TRF).**

I. Introduction

India is the seventh largest country in terms of land area, which makes its land and real estate markets one of the biggest and most lucrative in the world. Indian economy is developing rapidly, ensuring that the market continues to grow. The real estate sector in India is projected to experience significant growth, with the market size expected to reach US$ 1 trillion by 2030, a substantial increase from US$ 200 billion in 2021. This represents a compound annual growth rate (CAGR) of 19.5% between 2017 and 2028. By 2025, it is estimated that the real estate market will contribute approximately US$ 650 billion, accounting for 13% of India's GDP [1]. However, the current Indian land record system is incredibly complicated and involves many bureaucratic hurdles. Mishandling is widespread at various levels of the system, and the registration process and record-keeping are plagued with issues.

For every land parcel in India, a **Record of Rights** document is maintained, which is used by the farmers to obtain benefits from the government in the form of various subsidies and is also used to secure loans, sell the land, etc. The structure of this document varies from state to state.

To register a piece of land in India, a person needs to:

- Visit the **sub-registrar's office** to check for the authenticity and measurement accuracy of the land.
- A validation of the documents is also required with the help of a lawyer, which is costly and time-consuming.
- A **final sale deed** is done after validation, where all the valid documents of the transaction are presented at the sub-registrar's office. Here, considerable money and fees are paid for scrutinising the deed.
- Finally, a visit to the local municipality office is required, where the **mutation of the land title** takes place. It roughly takes a month and some funds for the municipal office to do a land survey and transfer the property to the new buyer's name.

The traditional land registration system is extremely inefficient and is plagued with problems. Some of the issues are listed below:

- Fake Property Documents
- Mutability of Data
- Land Encroachment
- Physical Documentation
- Delay in processing loans and subsidies
- Bureaucracy

For the modernisation of the land records system in the country, a modified programme, the National Land Records Modernization Programme (NLRMP), now renamed as Digital India Land Records Modernization Programme (DILRMP), was formulated by the Government of India in 2008 [2]. The idea is to mark all the existing parcels of land in our country and create digital records. This enables easy facilitation of land entities in India, and hence the transfer of ownership can be tracked appropriately.

However, it has been a slow progress, and none of the problems have been solved. The system is still heavily reliant on physical documents, and the digital registration process has added a layer of bureaucracy to the system.

II. Background Study

A. What is Blockchain?

Blockchain is a decentralized distributed ledger [4] that maintains a continuously growing list of records known as blocks. It is operated and monitored by a network of devices (also called nodes). A blockchain can have full nodes that hold a complete copy of the transaction ledger, as well as light nodes which are used for verification only. The nodes can perform various roles such as transaction verification, block generation, block validation, broadcasting state of blockchain, etc.

The first blockchain was created by Satoshi Nakamoto in 2008 to serve as the distributed ledger for Bitcoin, a peer-to-peer decentralised currency [5], based on previous work by Haber et al. [6] [7] [8]. Since then, the technology has evolved and found a multitude of applications apart from cryptocurrency.

A block in a blockchain is a data structure that contains a set of transactions, along with a header that contains metadata about the block. The header includes a block number, a timestamp, a hash of the previous block's header, a nonce, and a hash of the current block's transactions. The blocks are chained to each other using the hash of the previous block, making it tamper-proof. The transactions are a record of data exchanges or other actions that have occurred on the network and are to be added to the blockchain.

A consensus mechanism is a process by which nodes in a decentralized network agree to the state of the network. It ensures that all nodes have the same information and that the information is correct. It also ensures that transactions are verified and added to the blockchain in a secure and decentralized manner.

The features of Blockchain that make it desirable for applications all around the world are:

1) **Authentication and Verifiablility:** Every node participating in the blockchain has to be registered first. Each transaction requires the entity to authenticate itself. All entities and transactions can be easily verified.
2) **Immutability:** Each block contains the hash value of the previous block, ensuring a single change in any block is reflected as erroneous in all consequent blocks.
3) **Decentralisation:** Every full node has access to the entire blockchain and can initiate trades. The trades are then verified by neighbours.
4) **Transparency and Traceability:** It is possible to trace the entire history of every transaction, right from the genesis block. Hence, any modification can be easily traced, just by looking at the hashes.

B. Ethereum Ecosystem and Smart Contracts

The Ethereum white paper published by Vitalik Buterin [9] identified several flaws in the Bitcoin blockchain system:

- Lack of flexibility in Bitcoin's scripting language, which limits the complexity and diversity of applications that can be developed in the Bitcoin ecosystem.
- Potential for centralization due to the use of specialized mining hardware.
- High block generation times in Bitcoin (10 minutes on average) can lead to longer confirmation times for transactions and limit the network's throughput
- Lack of formal specification for Bitcoin's protocol, making it difficult to analyze the security of the system.

Ethereum has been developed to solve these issues. Its design is conducive for users to develop Decentralized Applications (DApps), by providing a Turing-complete virtual machine called Ethereum Virtual Machine (EVM), enabling developers to create and execute smart contracts on the Ethereum blockchain.

Smart Contracts(SC) are self-executing computer programs that run on the Ethereum blockchain. They are coded using Solidity, a contract-oriented programming language, and stored on the blockchain in bytecode format. They are designed to enforce the rules and regulations of an agreement between two or more parties without the need for intermediaries or third parties. They can be used for various purposes such as the execution of financial transactions, voting systems, decentralized applications, and more.

The execution of the SC is deterministic, meaning that it always produces the same result given the same inputs. This ensures that all parties involved in the contract can trust its outcome and that it always executes as intended.

C. Distributed Storage

A distributed storage system is a data storage architecture where data is stored across multiple servers or nodes in a network. Data is broken down into smaller chunks and replicated across multiple nodes, each responsible for storing a portion of the data. This redundancy ensures there is no single point of failure.

The traditional client-server model of file sharing has its limitations, such as centralized control, limited bandwidth, and the possibility of data loss. Hence in 2014, the IPFS protocol was designed to address these issues by providing a decentralized, distributed system for storing and sharing files [10].

IPFS uses a peer-to-peer network architecture to store and retrieve files, making it more robust and resilient than traditional file storage systems. This architecture enables files to be stored on multiple nodes, ensuring that the data is available even if some of the nodes go offline. Additionally, IPFS uses content-addressed storage, which means that files are identified and accessed based on their unique cryptographic hash rather than their location on the network.

III. Review of Prior Works

Land and asset registration mechanisms are incredibly different in every country. However, the existing research offers a good insight into the techniques and procedures we can employ to implement land and asset registration using blockchain everywhere.

A. Property Chain

Property Chain is a concept proposed by the National Informatics Centre under the Government of India's IndiaChain initiative [11]. The chain should contain digitally signed details of the property documents issued by departments involved in the Property Management System. These documents should be digitally signed and sent to the Property Chain. The property Chain is supposed to link these documents to the parent document, build the chain, and store it securely. Documents of sale, inheritance, mutations, survey updates, acquisitions, etc.

can be stored in the property chain. The following are some features of the property chain:

- Traceable, immutable, verifiable, secure, trusted
- Transparent, Paperless
- No dependence on the third party
- No notaries required
- Independent verification of ownership

The property chain also allows the retrieval of information stored. The retrieval can be enabled through the portal for any verifying agency, government departments, and citizens. The verifying authority could view the property details on the portal without depending on a third party for verification.

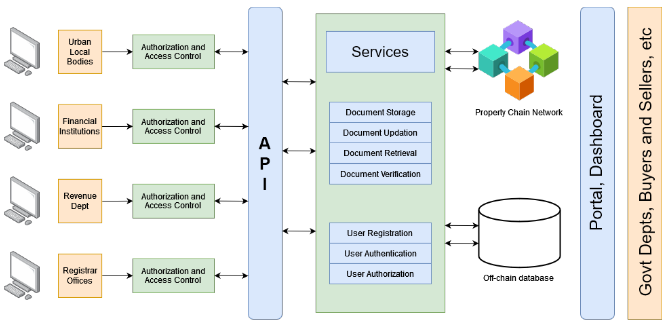

Fig. 1. Property Chain architecture proposed by [11]

B. Structure for Land Registration Data

Krishnapriya et al. proposed using Main Registration Offices (MRO) linked to Sub Registration Offices (SRO) [12]. MRO holds data on the original quantity of land present before the sale and SRO has data on the amount of land that has undergone transaction. SRO also has the data on remaining land available after a deal. The linkage forms a tree and is to be done using hashing. So, in essence, it creates a Merkle-tree-like structure [13] [14].

Users with multiple lands in multiple states are also tracked and linked with their respective lands. It forms a chain of users with basic transaction-related details, e.g. the previous and present owner of the property, the actual price, and the selling price of the property along with property size. This enables us to check if a specific land parcel is registered throughout the offices, determining the validity of the asset.

C. Transfer / Split of Asset

Transferring the whole asset is straightforward; Change the signature of the asset block to that of the new owner. However, it becomes complicated when you cut a parcel out of a pre-existing land and change ownership of that part. Krishnapriya et al. [12] propose two methods of land transaction. The idea of transfer of ownership for the entire land is to change the signing key of the land parcel to the new owner. However, the idea of splitting a land parcel is slightly different:

- A land parcel gets converted into two different land parcels
- The new owner's parcel gets signed by their private key

The previous owner's hash is also stored in the new land block. Anyone can verify the history of land parcels by backtracking using the hashes. This is possible because a Merkle tree is created by all land transfers. This tree makes it possible to backtrack and check the origin of each land block. If there is a discrepancy, i.e. hash of the previous block as present in the new block does not match the hash of the actual block, the chain can reject the new transaction.

D. Flow of Transaction

The flow of transacting the land is discussed and proposed by various authors. Krishnapriya et al. incorporate the property of splitting the land asset in their proposed flow of transaction [12]. Using relevant inputs like land ID, purchase amount, receiver's public key, etc. the algorithm retrieves the relevant information from the blockchain, such as previous transactions associated with the land. It then proceeds to create a new transaction, encapsulating the details of the land transfer. The transaction is signed using the elliptic curve algorithm, ensuring cryptographic proof of the sender's identity and maintaining the integrity of the transfer.

Upon successful transaction verification, the algorithm checks whether the land is undergoing split or whole transfer. In the case of split transfer, it splits the original land into new parcels based on the desired distribution, creating separate land assets for each party involved. Each new parcel is assigned a unique identifier by computing its hash value, ensuring a distinct and identifiable record. These updated land parcels are then added to the blockchain using a proof-of-work mechanism, securely recording the updated ownership and transfer details.

For whole transfers of the land, the algorithm follows a similar procedure, while skipping the splitting steps. Instead, it focuses on transferring the entire land asset from the sender to the receiver. The process involves signing and verifying the transaction, ensuring the authenticity of the transfer, and adding the updated ownership information to the blockchain.

Soner et al. propose a flow for transacting a land asset, as shown in Fig. 2. However, it involves a bank loan procedure and does not incorporate land splitting [15].

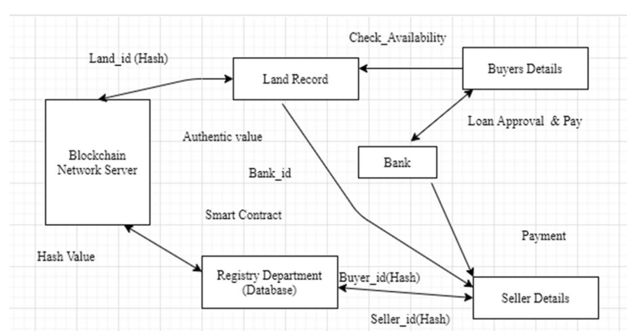

Fig. 2. Flow of transaction

E. Relevant Blockchain Type

Zakhary et al. proposed Fig. 3 as the arrangement for using multiple permissioned and one permissionless blockchain [16]:

- Permissioned for government to register assets
- Permissionless for doing transactions against cryptocurrency or trading assets against each other

Governmental offices run their trusted asset registration systems. These systems could be as simple as database management systems. They run permissioned blockchains with a set of trusted governmental officials called validators.

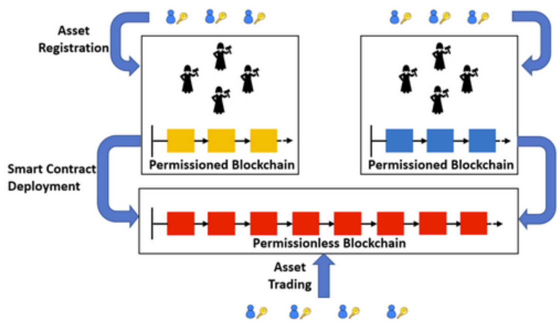

Fig. 3. Organization of multi-asset and transaction blockchains

There can be multiple types of assets, each registered in its appropriate blockchain. All of this is achieved in Ethereum and Bitcoin chains, which are permissionless blockchains where transactions are enabled between users.

IV. Proposed Solution

A. Motivation and Problem Statement

Land registry systems have relied on physical documentation to store records. It is an outdated system with a multitude of problems mentioned in *section I*. Recently, blockchain-based land registry approaches have gained popularity. A few approaches include ideas ranging from using public blockchains as open markets for land transactions and private blockchains for registration to using cryptocurrency to buy and sell assets on the blockchain itself directly. However, they lack specificity in their proposed algorithms with vague methods of registration, often failing to address the complexities of implementing such systems in the real world. Assets are mostly treated as single units, which works for assets like cars or houses. However, it fell short when land was concerned. The reliance on cryptocurrency is impractical as it falls into a legal grey area while also being highly volatile.

To address these drawbacks, we propose a blockchain-based system for the transaction and ownership of land assets. In light of this, our contributions are as follows:

1) We design a permissioned blockchain for the registry and transaction of assets between users.
2) We provide a mechanism for assigning ownership to land assets which can be openly verified.
3) We use Role-Based Access Control to manage permissions.
4) We provide a lightweight system which uses on-chain memory efficiently and employs distributed cloud storage for hosting land asset files.

B. System Model Design

The proposed system allows users to transact land assets using a permissioned blockchain. Asymmetric key cryptography is used to encrypt transaction files to be sent to other users and the land inspector for eventual approval. The land asset files are stored in distributed cloud storage, and the ownership of these assets is certified via the generation of Non-Fungible Tokens (NFTs).

- **System Environment:** For setting up the development environment, a Python-based framework called Brownie is used. Ethereum-based SCs are written in Solidity programming language and are then deployed to the blockchain via a Python program. Entities interact with the system through a text-based interface, also developed using Python.
- **Blockchain:** We use Ganache for generating an Ethereum-based local blockchain. We use it for hosting the SCs required to transact land assets and generate NFTs. Ganache itself generates the constituent nodes for the blockchain.
- **NFT:** The ERC721 token standard, also known as Non-Fungible Tokens, are Ethereum-based tokens which are non-replicable ie. the hash of the token is unique. They are produced via the use of a separate SC deployed for this specific purpose. Land asset ownership is assigned in our system by generating NFTs of the file containing data of that land asset while specifying the current owner's wallet address.
- **Key Generation and Distribution:** The Ganache blockchain is responsible for key generation and distribution. It uses Elliptic Curve Cryptography to generate public-private key pairs, which are assigned to each user joining the system. The user uses the private key to sign a transaction file, which then gets sent to another user or land inspector. The public keys are used as the wallet address of a user and to verify the signature on any incoming transaction file.
- **Entities:** There are two types of entities interacting with the system: Users and land inspectors. The users are people involved in the transaction of land assets. They cannot directly register transactions in the blockchain and only interact via an interface. They generate transaction requests, which require the approval of the land inspector. Land inspector, post transaction approval, and call functions on the SCs, which are deployed on the blockchain.
- **Distributed Cloud Storage:** We use the Inter-Planetary File System (IPFS) as our distributed storage. It hosts the files which are used as NFTs for land assets. We use Pinata, a Python-based API client, for uploading files to IPFS and generating a link for the same.

C. Storage Management

Any approach to storing user data for our purpose should adhere to the following goals:

- The storage should be secure from potential attacks
- Ideally, it should be decentralized to prevent a single point of failure
- The retrieval and modification of data should be quick and efficient
- It should provide proof of ownership
- It should be financially feasible

In the Ethereum blockchain, no direct support exists for external databases, both relational and non-relational. This is because the SCs are executed on the Ethereum Virtual Machine (EVM), which is a separate and isolated environment from traditional databases. However, it is possible to integrate SCs with external databases using specialized middleware or oracle services. These services act as intermediaries between the contract and the external database, providing a way for the SC to read and write data through specific API calls. They however charge service fees, perform repeated queries on the blockchain, add delay and provide no concrete proof of ownership, since the user does not digitally sign the data.

However, since Ethereum comes with PoS consensus, we can use oracle services to implement consensus algorithms such as PoID.

Another approach is to store all the data on the blockchain. Hence we have to personally create a CRUD (Create, Read, Update, Delete) database in the SC itself, through which we can manage the land record data. It is faster than oracle services while also being transparent (available on the blockchain) and immutable. However, the storage of large amounts of data on the blockchain results in a significant processing load. As more individuals and records are added to the system, high gas fees and financial infeasibility become major issues, hindering the adoption of the approach.

Thus, the idea is to efficiently use distributed cloud storage (e.g. IPFS) for storing land data files, while also utilizing on-chain storage for validation and access purposes. Since land data is extensive and elaborate, it can be stored on IPFS; lightweight data such as a list of usernames and access data can be stored on the blockchain itself. This allows optimized utilization of the system, keeping access times low and reducing the use of resources. Since we are using IPFS, open verification of data is possible due to easy accessibility. IPFS also allows tracking the modification history, further propagating the immutable and transparent nature of this storage mechanism.

D. Flow of Work

1) Phase I: User and Asset Registration: This phase involves the unilateral registration of users and their assets. The user enters its details, post which the user is registered on the blockchain. The blockchain generates a public and private key pair. The public key acts as the wallet address for the user, which is used to reflect the ownership of the land assets. The private key is used for digital signature. The public key is mapped to the username, and the private key is hidden from the network.

The user enters the corresponding land ID. If it does not exist in the system, the user is prompted to fill in more details, post which a Transaction Request File (TRF) is generated. This file is in JSON format, which is signed by the user's private key, and then sent to the land inspector. The land inspector decrypts the corresponding file using the user's public key, thus verifying the signature. After the inspector approves the details of the TRF, the land data gets registered in the blockchain, and the land ID is mapped to the land record. An NFT is also generated, which contains data on the registered land and is assigned the wallet address of the user. Hence, it can serve as proof of ownership and can be verified easily by any other user in the blockchain.

2) Phase II(a): Asset Transaction: A user interested in purchasing land can generate a TRF by entering a pre-existing land ID and specifying the desired type of land transfer. Land transfers can be of two types:

- **Whole:** When a user wishes to purchase an entire plot of land from a seller, a land transfer is carried out, resulting in a change of ownership.
- **Split:** When the user wishes to buy a portion of the original land plot from the seller, the section to be transferred experiences a change in ownership, while the remaining land plot remains under the original owner.

The TRF is then signed by the buyer's private key and sent to the seller of the requested land. The seller decrypts the file using the buyer's public key. The seller reviews the transaction details, signs the transaction with their private key, and forwards it to the land inspector. The flow of operations and events is displayed in Fig. 4.

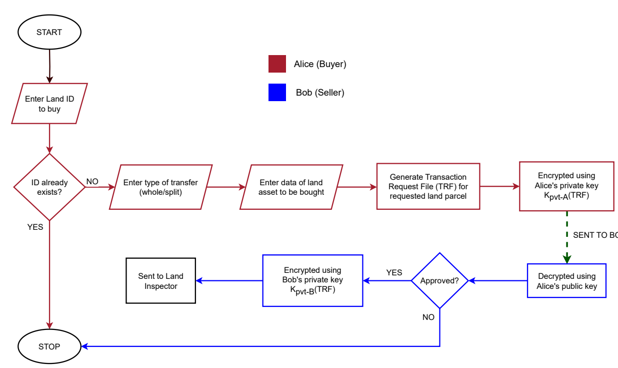

Fig. 4. Asset Transaction Flow

E. Phase II(b): Land Inspector Approval and NFT Generation

The land inspector can approve transactions, which are then added to the blockchain. A TRF is generated either during the first land registration or during the land transfer. The corresponding keys are used for decrypting the TRF. If the TRF satisfies the necessary conditions, the land inspector generates one or more Land Asset Files (LAFs) corresponding to the number of land parcels produced.

In the whole land transfer, there is no change in the land ID of the asset. But in split land transfer, new IDs are generated in the following manner:

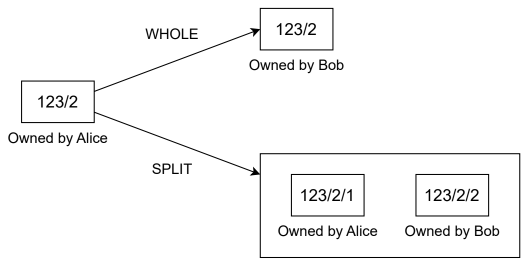

Fig. 5. Generation of Land IDs

The idea of generating LAFs is that the files can be used as proof of ownership for the new land parcels. To achieve this, the land inspector uploads these files to the IPFS cloud, via the local IPFS node. It uploads the block and gives it a content ID (CID). Based on the CID, its IPFS gateway URL is generated.

The land inspector creates an ERC721 NFT token. The metadata of the NFT is populated, where the Uniform Resource Indicator (URI) of the NFT is the IPFS gateway URL of the Land Asset File. The metadata is then associated with the SC explicitly created for this purpose through a unique hash value. Minting involves the creation of an instance of the NFT linked to the SC and metadata through the SC itself. The NFT is transferred to the new owner's public key. This transfer involves updating the ownership information in the SC and updating the metadata to reflect the new owner. After the NFT is minted, the NFT denoting the old land record is burned. The SC sends that specific token to a dummy address which is not accessible, rendering the NFT useless, inoperable, and non-transferable. This irreversible process of destroying an NFT is essential to maintain the integrity and authenticity of the system.

F. Interaction with Blockchain

We designed an interface using Python to interact with the SC. We initially deploy the SC on the blockchain using a Python program designed explicitly for this purpose. The deployed SC is assigned an address that can be used to access it.

All information related to users, their land, and their mappings is stored on the blockchain. When a SC is to be deployed, an instance is created in the interface, which can be stored in a variable. This variable can be used to perform the necessary functions intended in the SC. No user/land information is stored in the interface(except username).

Every user has an assigned set of public-private key pairs. The user's identity on the blockchain is their public key, which acts as their wallet address. When an NFT is minted, the ownership specified in the NFT refers to the same public key. This enables the user to view all their land assets in their wallet. The public key can be used to verify the user's ownership.

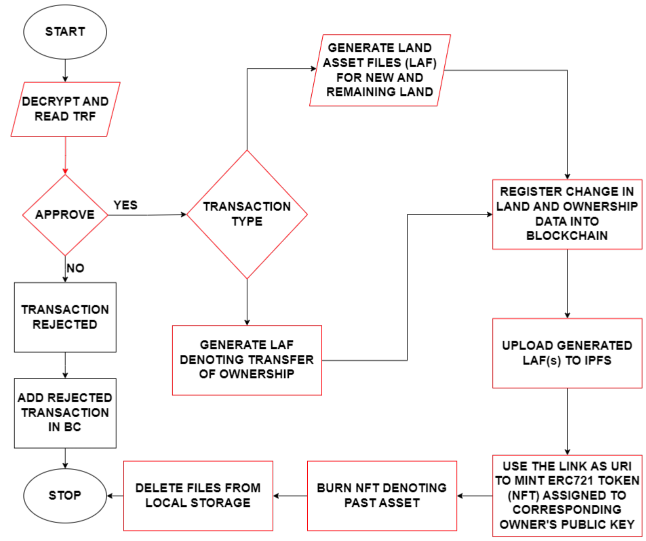

Fig. 6. Flow of Land Inspector Approval and NFT Generation

V. Performance Analysis and Results

There have been many proposals and implementations of Land Asset registration and transfer. We have compiled their comparisons into TABLE I and TABLE II.

- Scalability is determined by the average Transactions Per Second(TPS). Low scalability signifies a TPS of less than 20, Medium ranges from 21 to 50, High ranges from 51 to 100, and Very High starts from 101 onwards.
- ECDSA provides better resistance against cryptanalysis as compared to Diffie-Hellman and RSA-based signature.
- Since land assets are transacted is splits, it is of utmost importance to handle this.
- Access control is a data security process that enables organizations to manage who is authorized to access resources. Our reason for choosing RBAC and its appropriateness has been explained in subsection V-B.
- Storing the least amount of data in the blockchain is significant since every node replicates the same data. We have explained this in detail in subsection V-E. We have also compared IPFS with oracle services in subsection IV-C.
- The storage requirements are a direct result of storing user details and tokens created. This has been discussed in subsection IV-B.
- Blockchain can be implemented on various frameworks such as Ethereum Virtual Machine (EVM), Hyperledger Fabric, Hyperledger Sawtooth, Polka-dot Substrate, and so on. Each framework provides us with different tools. We chose EVM as it fit our needs.

- We use the ERC-721 NFT standard to create our token. Our reason for doing so has been discussed inIV-B.
- The consensus used plays a very impactful role in the blockchain system. Ethereum comes with PoS by default. We have proposed using PoID. However, it is achievable using JSON-RPC using Oracles.
- The proposed model requires low energy, as it employs PoID, and uses minimal storage.

A. Computation effort Analysis:

Most implementations have high computation costs compared to our proposed system. They either use high computation consensus algorithms [36] such as PoW, which adds computational overhead, or they store all information on-chain. All information on the blockchain is replicated on all its nodes, causing unnecessarily high redundancy. The gas fees reflect this fact. We can see in TABLE III that using IPFS costs less than 10% gas. The gas cost can further be reduced based on how the IPFS hash is stored in the blockchain. If stored as a string, the gas cost will be around 88000, if stored as a struct, it requires 55000 gas, and if stored in the event log, it requires even lower gas costs.

B. Consensus Algorithm:

The consensus algorithm used in the blockchain plays an impactful role in the computation, storage and delay costs associated with transactions. Choosing the appropriate consensus can make the difference between 7 transactions per second to 20,000 transactions per second. It can also mean the difference between a secure, tamper-proof blockchain system to an easily breakable system. We need to find the intricate balance of the security aspects such as access control, authorization, signature, privacy, etc along with computation costs, and delays incurred while maintaining decentralization. PoW has low TPS, but security scales well. PoA provides great TPS but depends on trust [37]. PoS requires the nodes to put stakes and using it makes no sense in land registration systems. PoS has security issues, poor scalability, tends to centralise, and doesn't make efficient use of resources. A consensus algorithm alone is not able to provide the necessary balance between security, reliability, and scalability. However, when used in conjunction with an access control policy, the combination works wonders. Access control enables us to decide what SCs can be accessed by entities, based on their identities, or roles. Since, our implementation has two entity types, a Role-Based Access Control(RBAC) fits best. RBAC allows us to have the least number of access policy rules stored in the blockchain, further reducing storage costs. We propose to use a Proof-of-Identity(PoID) consensus algorithm that has high TPS, with a minimal number of verifications required, while ensuring that the transactions are authenticated, signed, immutable, and secure. PoID is not available on EVM, however, it can be implemented using oracles. PoID has the following advantages:

- PoID security is based on the impossibility of an attacker to create identity forgeries, hence, Sybil attacks are impossible.
- Permanently censorship resilient: because of the high level of distribution and decentralization and the privacy of the minters, the PoID is marginally affected by censorship.

The drawbacks of PoID are mitigated with the use of human entity $"LandInspector"$ and IPFS. A hierarchy of signatures similar to DNSSEC can be used to further ensure no trades can be faked.

C. Signature Scheme analysis:

As visible in TABLE I, the majority of the systems either have not discussed their signature schemes or use RSA for their digital signature. We can see in TABLE **??** that ECCDSA provides much better resistance against cryptanalysis, as compared to RSA and Deffie-Hellman DSA with a much smaller key size. Therefore, not only does it reduce computational requirements, but also provides better security.

D. Impact of Land Splitting

Land assets are very rarely bought and sold as a complete asset. It is broken down, and combined, and each piece of land asset created due to splitting is unique. One portion of the split land can be drastically different from another portion of the same land asset. The differences may occur based on the direction the parcel of land faces, road connectivity, closeness to a water body, view, and so on. Therefore, it is of utmost importance to allow the selling and buying of portions of land assets. In our paper, we have used NFTs that are created when a land asset is first registered, or when the asset is split. In the case of a whole land transfer, the ownership of the NFT is transferred. However, when an asset is split, the equivalent number of NFTs is generated. Each NFT is the proof of ownership of that land parcel, and it can be easily verified with the public key of the owner. Each NFT is signed by the owner using ECDSA as proof of ownership.

E. Storage Analysis

Considering all the nodes of the blockchain network are full nodes, one major drawback of blockchain is that every node has a copy of the entire blockchain. Bitcoin [5] requires about 531.38GB of storage for each node as of $7thDecember2023$, and it's only increasing. There are ways around it to reduce the amount of storage space required [38]. One of them is to store data in the blockchains until they become obsolete and get rid of entire blocks that aren't holding necessary information, or by using selective deletion [39]. However, that is not feasible when it comes to land assets. The history of land assets has to be maintained for as long as possible to ensure no legal complications arise.

One of the ways we reduce storage requirements is by using IPFS, where we create NFTs and store the NFTs in IPFS, and only use the blockchain to store the signed hash of the NFT. Anyone can verify that the NFT belongs to the specified owner

TABLE I

COMPARISON OF PAPERS USING BLOCKCHAIN FOR LAND REGISTRY

Sl. No.	Paper Name	Scalability	Signature Scheme	Asset-splitting	Access Control	User Details Stored In
1	[21]	Low	ECDSA	No	None	Blockchain
2	[22]	Low	Not discussed	No	None	Blockchain
3	[23]	Medium	ECDSA	No	None	Blockchain
4	[24]	Low	RSA	No	Not discussed	Blockchain
5	[25]	Low	RSA	No	None	Blockchain
6	[26]	Low	RSA	No	None	Blockchain
7	[27]	Low	Not discussed	No	None	Blockchain
8	[28]	Low	ECDSA	No	None	Blockchain
9	[29]	Low	RSA	No	Not discussed	Blockchain
10	[30]	Low	RSA	No	Not discussed	Blockchain
11	[31]	No	ECDSA	No	None	Blockchain
12	[32]	No	ECDSA	No	None	Blockchain
13	[33]	Medium	RSA	No	IBAC	Blockchain
14	[34]	High	ECDSA	No	None	IPFS
15	[35]	Low	RSA	No	ACL	Blockchain
16	Proposed Scheme	Very high	ECDSA	Yes	RBAC	IPFS

TABLE II

COMPARISON OF PAPERS USING BLOCKCHAIN FOR LAND REGISTRY

Sl. No.	Paper Name	Storage Needs	Framework Used	Land Token used	Consensus used	Energy Needs
1	[21]	Very High	Ethereum	None	PoS	Low
2	[22]	Very High	Not discussed	None	Not discussed	Not discussed
3	[23]	Very High	Ethereum	None	PoS	Low
4	[24]	Very High	Hyperledger Fabric	None	pBFT	Medium
5	[25]	Very High	Not discussed	None	Proof of Authority	Low
6	[26]	Very High	Hyperledger Fabric	None	pBFT	Medium
7	[27]	Very High	Not discussed	None	PoW	High
8	[28]	Very High	Ethereum	None	PoW	High
9	[29]	Very High	Hyperledger Fabric	None	pBFT	Medium
10	[30]	Very High	Hyperledger Fabric	None	pBFT	Medium
11	[31]	Very High	Ethereum	None	PoW	High
12	[32]	Very High	Ethereum	None	PoW	High
13	[33]	Very High	Not discussed	None	Proof of Identity	Low
14	[34]	Low	Ethereum	ERC 721	PoS	Low
15	[35]	Very High	Hyperledger	None	pBFT	Medium
16	Proposed Scheme	Low	Ethereum	ERC 721	Proof of Identity	Low

TABLE III

GAS COST WITH IPFS VS WITHOUT IPFS

Sl. no.	Transaction Fees (Gas used)	
	With IPFS	Without IPFS
1	55600	765352
2	55458	758452
3	54985	542563
4	55124	546895
5	54346	742165
6	54856	715425

easily while also keeping the minimum amount of data in the blockchain reducing storage needs drastically. We also store minimal user information in the blockchain. Only the user ID and its corresponding public key are stored on the blockchain.

VI. CONCLUSION

In this paper, we have proposed a blockchain-based land transaction and ownership system. We used Ganache to create an Ethereum-based permissioned blockchain, while Brownie has been used to create the framework for the project. Since only SCs can be used to interact with the blockchain, we designed them using Solidity. We also designed a Python-based interface that enables users to register and securely transact land assets. We minted Non-Fungible Tokens (NFTs) to serve as proof of ownership of land assets. To ensure our system is lightweight and efficient, we used distributed cloud storage (IPFS) for storing NFTs. Our solution ensures immutability, transparency, and verifiability while being flexible enough that integration into traditional land registration systems is a cinch.

REFERENCES

[1] IBEF. Real estate industry in india, 2023.

[2] Amrutaunshu Nerurkar and Indrajit Das. Agile project management in large scale digital transformation projects in government and public sector: A case study of dilrmp project. In *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance*, pages 580–581, 2017.

[3] Prachee Mishra and Roopal Suhag. Land records and titles in india. *Ideas for India*, 20, 2017.

[4] Badr Bellaj, Aafaf Ouaddah, Emmanuel Bertin, Noel Crespi, and Abdellatif Mezrioui. Sok: A comprehensive survey on distributed ledger technologies. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–16, 2022.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[6] Stuart Haber and W Scott Stornetta. *How to time-stamp a digital document*. Springer, 1991.

[7] Dave Bayer, Stuart Haber, and W Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. In *Sequences II: Methods in Communication, Security, and Computer Science*, pages 329–334. Springer, 1993.

[8] Stuart Haber and W Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28–35, 1997.

[9] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[10] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[11] Govt. of India NIC. Concept paper on property chain. Concept Paper, 2019.

[12] S Krishnapriya and Greeshma Sarath. Securing land registration using blockchain. *Procedia Computer Science*, 171:1708–1715, 2020.

[13] Ralph C Merkle. Method of providing digital signatures, January 5 1982. US Patent 4,309,569.

[14] Ralph C Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology—CRYPTO'87: Proceedings 7*, pages 369–378. Springer, 1988.

[15] Swapnil Soner, Ratnesh Litoriya, and Prateek Pandey. Exploring blockchain and smart contract technology for reliable and secure land registration and record management. *Wireless Personal Communications*, 121(4):2495–2509, 2021.

[16] Victor Zakhary, Mohammad Javad Amiri, Sujaya Maiyya, Divyakant Agrawal, and Amr El Abbadi. Towards global asset management in blockchain systems. *arXiv preprint arXiv:1905.09359*, 2019.

[17] Maria Kaczorowska et al. Blockchain-based land registration: Possibilities and challenges. *Masaryk University Journal of Law and Technology*, 13(2):339–360, 2019.

[18] Jan Veuger. Dutch blockchain, real estate and land registration. *Journal of Property, Planning and Environmental Law*, 12(2):93–108, 2020.

[19] Qin Wang, Rujia Li, Qi Wang, and Shiping Chen. Non-fungible token (nft): Overview, evaluation, opportunities and challenges. *arXiv preprint arXiv:2105.07447*, 2021.

[20] Christian Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, pages 1–4. Chicago, IL, 2016.

[21] Shreyas Dubey, Divyansh Vinod, Akash Gupta, and Rajendra Kumar Dwivedi. Secure land registration management via ethereum blockchain. In *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pages 185–191. IEEE, 2023.

[22] Salman Humdullah, Siti Hajar Othman, Muhammad Najib Razali, and Hazinah Kutty Mammi. Secured data storage framework for land registration using blockchain technology. In *2021 3rd international cyber resilience conference (crc)*, pages 1–6. IEEE, 2021.

[23] G Kusuma, Ch Rupa, Sk Reshma, and G Rochana. Secure storage of land records and implementation of land registration using ethereum blockchain. In *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pages 404–409. IEEE, 2023.

[24] Aidil Rezjki Suljztan Syawaludin and Rinaldi Munir. Registration of land and building certificate ownership using blockchain technology. In *2021 International Conference on ICT for Smart Society (ICISS)*, pages 1–7. IEEE, 2021.

[25] Nidhi Gupta, Manik Lal Das, and Sukumar Nandi. Landledger: blockchain-powered land property administration system. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2019.

[26] Nolwazi Ncube, Belinda Mutunhu, and Khulekani Sibanda. Land registry using a distributed ledger. In *2022 IST-Africa Conference (IST-Africa)*, pages 1–7. IEEE, 2022.

[27] Milon Biswas, Jabed Al Faysal, and Kazi Asif Ahmed. Landchain: A blockchain based secured land registration system. In *2021 International Conference on Science & Contemporary Technologies (ICSCT)*, pages 1–6. IEEE, 2021.

[28] Ameya Thosar, Mayur Hame, Ashutosh Sarode, and Parminder Kaur. Land registry management using blockachain. In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, pages 335–340. IEEE, 2020.

[29] Sai Apurva Gollapalli, Gayatri Krishnamoorthy, Neha Shivaji Jagtap, and Rizwana Shaikh. Land registration system using block-chain. In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, pages 242–247. IEEE, 2020.

[30] Limark Dcunha, Larren Dsouza, Gokul Ramanan, and Sheetal Chaudhari. A blockchain-based approach to streamlining land registration using hyperledger fabric. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–6. IEEE, 2023.

[31] Md Sakibul Islam, Fahmid Shahriar Iqbal, and Muhaimenul Islam. A novel framework for implementation of land registration and ownership management via blockchain in bangladesh. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 859–862. IEEE, 2020.

[32] RC Suganthe, N Shanthi, RS Latha, K Gowtham, S Deepakkumar, and R Elango. Blockchain enabled digitization of land registration. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5. IEEE, 2021.

[33] RM Yasas, MHMND Bandara, T Praveena, K Abeywardena, and D Kasthurirathna. Decentralized property registration and management platform. In *2022 4th International Conference on Advancements in Computing (ICAC)*, pages 328–333. IEEE, 2022.

[34] Cyril Kunjumon, Aditya Mutharia, Salemula Shareef, and MS Suchithra. Implementing a land registration system using non-fungible tokens to represent land in the system and side-chain for data storage. In *2023 International Conference on Control, Communication and Computing (ICCC)*, pages 1–5. IEEE, 2023.

[35] Yogita Borse, Purnima Ahirao, Anushka Chawathe, and Deepti Patole. Embracing permissioned blockchain for a secure and reliable land registration system. In *2022 5th International Conference on Advances in Science and Technology (ICAST)*, pages 525–530. IEEE, 2022.

[36] Luca Ambrosini, Matija Piškorec, and Claudio J. Tessone. Visualization of blockchain consensus degradation. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–2, 2022.

[37] Xinrui Zhang, Qin Wang, Rujia Li, and Qi Wang. Frontrunning block attack in poa clique: A case study. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2022.

[38] Jun Wook Heo, Ali Dorri, and Raja Jurdak. Multi-level distributed caching on the blockchain for storage optimisation. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5, 2022.

[39] Peter Hillmann, Marcus Knüpfer, Erik Heiland, and Andreas Karcher. Selective deletion in a blockchain. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1249–1256. IEEE, 2020.

[40] Chaehyeon Lee, Changhoon Kang, Heeju Ko, Jongsoo Woo, and James Won-Ki Hong. A comprehensive and quantitative evaluation method for blockchain protocols. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5, 2023.

[41] Mikel Cortes-Goicoechea, Tarun Mohandas-Daryanani, Jose Luis Muñoz-Tapia, and Leonardo Bautista-Gomez. Autopsy of ethereum's post-merge reward system. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9, 2023.

[42] Marco Benedetti, Francesco De Sclavis, Marco Favorito, Giuseppe Galano, Sara Giammusso, Antonio Muci, and Matteo Nardelli. Powless bitcoin with confidential byzantine poa. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2023.