Cluster Computing
https://doi.org/10.1007/s10586-022-03535-0

Sidechain: storage land registry data using blockchain improve performance of search records

Amrendra Singh Yadav[1] · Nikita Singh[2] · Dharmender Singh Kushwaha[1]

Received: 19 June 2021 / Revised: 17 November 2021 / Accepted: 5 January 2022

Abstract
This article proposes a novel sidechain structure for a land/property registry management framework to overcome the limitations of an existing land registry process. This research proposes two different types of blockchains namely mainchain and sidechain. Metadata is stored in the mainchain that is publicly accessible, and non-transactional data, such as pictures, contracts, PDF, other allied information, is stored in the sidechain. It mainly focuses on using blockchain in the land registry data storage, improving information management, reducing authentication time, decreasing storage consumption that leads to optimized searching of land records in the blockchain. The proposed framework uses a summary file and InterPlanetary File System based on a Peer-to-Peer swarm-based network. The summary file concept is used for searching documents in the sidechain, which can be queried using attributes of the registry and can be accessed by only authorized parties or persons involved in that transaction. The comparative assessment establishes that the proposed approach is more efficient than the existing leading approaches. The land record search time of the proposed algorithm is on an average lesser by 60% as compared with the linear search approach and by 25% when compared with the hash table-based search algorithm. Also the proposed approach LRSA has reduces block approval time during consensus process by 55.63%, 28.24%, 25.30%, and 3.62% while comparing it with the existing leading algorithms PoW, PoS, DPoS, TrVCA, and LRSA respectively.

Keywords Sidechain · Mainchain · Peer-to-peer · Blockchain · Property transaction system · Two-way peg protocol

1 Introduction

Technology has boomed in almost every field in the modern era, such as e-commerce, industry, medicine, administration, government, etc. Therefore, technology is also required for the transactions performed in the real estate industries [1]. If a transaction for land or property is being performed between a buyer and a seller, that process is known as land/property registration. However, the current land record management scheme is riddled with numerous weaknesses and defects. Different agencies manage multiple land records, and there is no uniform structure in place to maintain these land records. These vulnerabilities also lead to land ownership disputes. In India, frauds relating to property documents and evasion of stamp duties are prevalent in financial transactions, corporate, and other government machinery. An increase in the frequency of property transactions comes with the risk of fraud, thus forcing society to demand transparency in transactions.

Governments worldwide are encouraging digital documentation and digital transactions to bring transparency into the system. Technologies do improve the efficiency and performance of the system, but there exist various challenges. Digital documents are vulnerable to alteration and various network attacks. Digital documents can be

✉ Amrendra Singh Yadav
yadavamrendrasingh@gmail.com

Nikita Singh
nikitasinghk@ifheindia.org

Dharmender Singh Kushwaha
dsk@mnnit.ac.in

1 Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India

2 Department of Computer Science and Engineering, ICFAI University, Hyderabad, India

Published online: 21 January 2022

tampered with, and ownership of the documents can be changed. Other challenge includes storage, availability, and scalability of these documents. Organizations need to provide secure storage for digital documentation, and access to authorized users must be monitored. The centralized storage involves a single point of failure as well as network traffic overhead.

The land registry system is reliant on government departments for processing and verification of all documents. Due to the lack of connectivity between all land register databases and the land registry system, the system's reliability and equality suffer. Land registration is a lengthy process that requires extensive paper verification, physical visits, financial exchanges, resulting in significant delays and a waste of time and money. Another significant issue is the lack of proper records. The land registry process requires periodic changes in ownership, which takes time and is incapable of providing real-time information for any verification requested. The land title is a very contentious issue since most cases in the court relate to land title dispute due to different reasons such as fake identity of people involved, false documents, and inability to recognize the real owners correctly.

Distributed ledger technology (DLT) is a decentralized and stable solution to addressing most of these issues. It has the potential to bring about much-needed radical improvements in several welfare operations that the government runs to help our country's underprivileged citizens [2]. Digital governance as a service is quickly becoming the de facto industry norm. The country's rapid economic growth has resulted in an increase in the frequency of property transactions, which comes with the risk of fraudulent activities, prompting society to demand secure and open transactions and improved efficiency and transparency. DLT allows transactions and information to be recorded on ledgers shared and synchronized across a distributed network. It finds its roots in Blockchain based technologies.

The nature of blockchain is immutable and transparent, making it reliable for invasive and deceitful property registration activities. Blockchain technology in the land registry solves many issues due to maintaining a centralized record of titles. There is a decentralized control and storage of records in blockchain, increasing confidence and giving rise to a collaborative framework [3]. A blockchain is a distributed network of peer-to-peer computers. Each peer has an identical copy of a transaction that the network's peers have agreed upon. The decentralized standard system for land registry records would eliminate the need for intermediaries, shorten the process's duration and cost, and improve the process's efficiency and trustworthiness among transacting parties.

In a blockchain, the size of a block is kept limited and includes details of few numbers of transactions. After both buyer and seller agree to any property transaction, one has to perform all the laborious but necessary tasks for making this a valid transaction. The buyer needs to get the authorized vendor's signature, stamp paper, payment has to be made, property documents need to be submitted, which can be an image, a PDF, etc. [4]. Identical copy of the document is getting stored at every step and this escalates with every minor changes. The problem regarding storage arises because there is no need to store the same non-transactional data at every step.

The amount of data stored in the blockchain should be limited to reduce the search and access time. Therefore, it is important to decide which data should be stored in the blockchain [5]. The storage problem is one of the important issues in land registry management systems using blockchain since it is a large document file (PDF, images, contract).

It is noteworthy that e-registry also carries multiple pages of document relating to sale deed/registry and it also needs to be stored securely. If all these documents are also stored in blockchain, then each block may only accommodate a very few transactions and size of blockchain shall increase drastically. This shall pose severe challenge on any search and storage. So we need a separate chain to store the non-transactional data and reduce the load on a blockchain. It is proposed that we store the metadata on blockchain and non-transactional data in the sidechain for increasing its efficiency. In order to perform a lookup for any transaction record, we will need an efficient retrieval/searching method to access these. Thus, this work mainly focuses on storing the property e-registry data storage, improving information management efficiency, and optimizing the searching of property records in the blockchain.

The optimized two-way peg protocol ensures that information between the mainchain and sidechain is shared securely.

1.1 Sidechain mechanism

Sidechain was initially used to allow bitcoin and other cryptocurrencies to move funds between multiple blockchains. The sidechain structure consists of a mainchain and several sidechains. Both mainchain and sidechain share the fundamental structure of blockchain technology, including the block structure. The primary functions of the sidechain are to enable the transfer of critical information between chains and alleviate pressure on the mainchain, enabling the system to achieve both mobility and the ability to use multiple networks (Fig. 1).

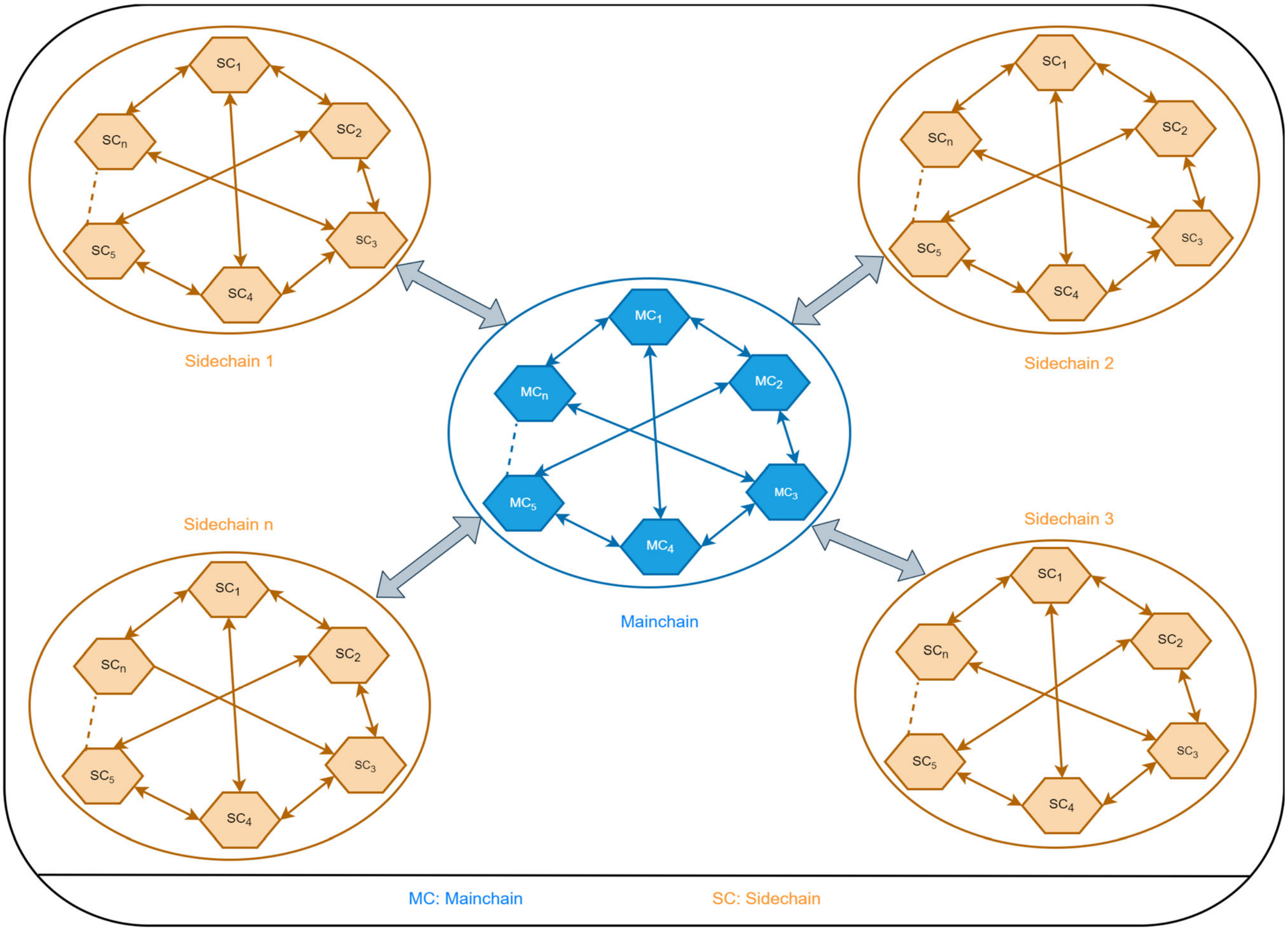

Fig. 1 Structure of sidechain mechanism

Sidechains are secondary blockchains that are connected to the mainchain. The original blockchain is usually referred to as the mainchain and all additional blockchains are referred to as sidechain. A sidechain is a separate blockchain, attached to a mainchain through the use of a two-way peg, which allows the data to be interchangeably moved across the mainchain and sidechain. It acts as a means to ensure integrity and security between the mainchain to the sidechain.

1.1.1 Simplified payment verification and two-way peg protocol

A two-way peg is a procedure that enables the two-way transfer of data between the sidechain and the mainchain. This customizability gives the freedom to the user to ingress a variety of features in the sidechain. Since the sidechain is separate from the mainchain, if any damage occurs in the main chain due to a cryptographic break, data at side chain still remains secure. Figure 2 illustrates the well-known two-way peg design solution for the sidechain technologies emphasizes their disadvantages and advantages.

1.2 Motivation

It's a long history of conflicts over property and the loss caused by these is quite evident. According to the most recent study, investments totalling trillions of dollars are entangled in ongoing land conflicts across globe. It is a concern for any country to enforce a property registry system to confront discrepancies arising with the present system. A secure and decentralized platform for reducing conflicts over the property based on the transaction history of records is required. Blockchain based technologies provide a solution for offsetting most of the problems discussed with the traditional framework of property record transaction and maintenance. With the proposed system, a public user can also query the system to check out his/her ownership history without a need for an intermediary institution. An access control mechanism can also be

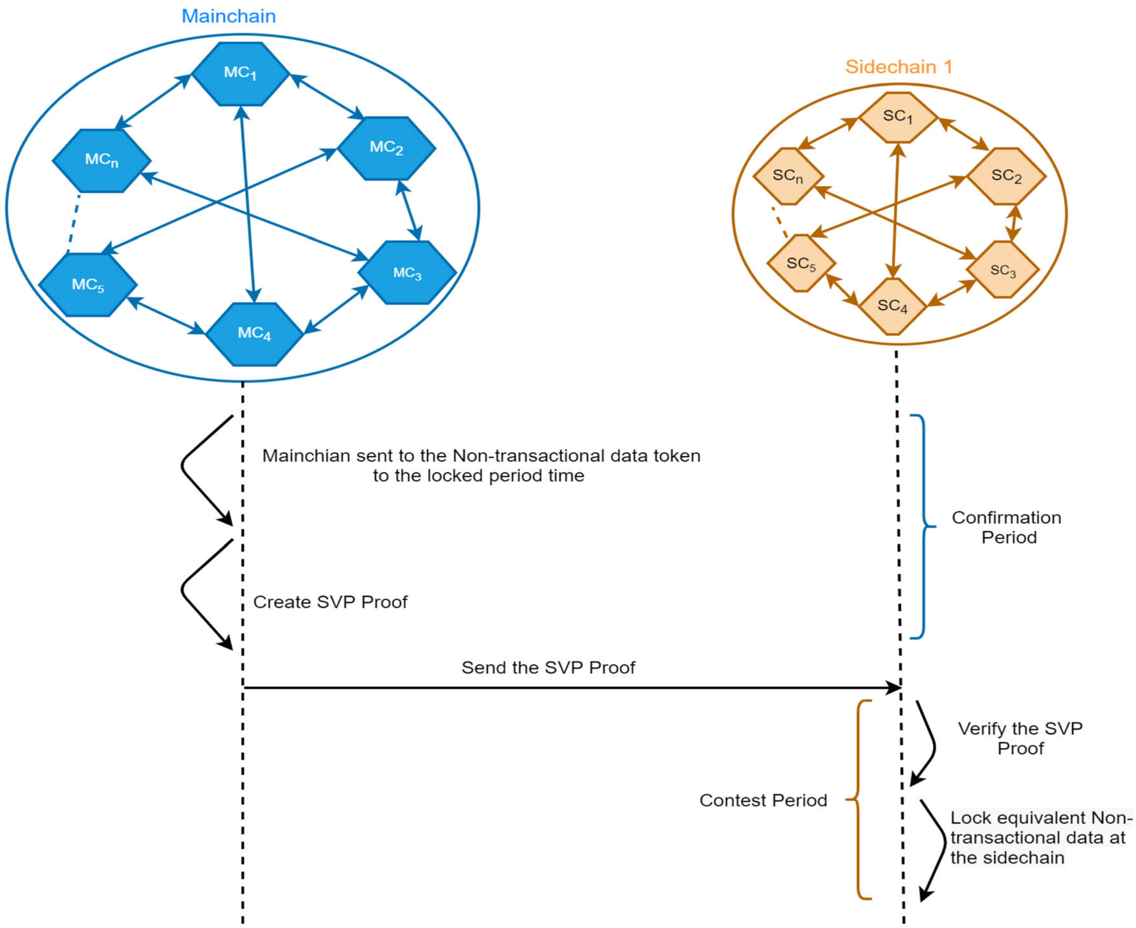

Fig. 2 Process of the two-way peg protocol in sidechain mechanism

enforced, not allowing unauthorized users to manipulate the information.

This article proposes a blockchain-based solution framework that addresses the mainchain's storage issue that consumes less storage, and improves search efficiency for land record data. The proposed solution focuses on providing a decentralized and secure P2P framework for managing land registration and an interface for verifying document authenticity. The proposed solution is built on a hybrid blockchain framework.

This article is organized in six sections. Section 2 briefs the leading research relevant to the suggested approach. Section 3 presents the proposed Efficient Storage Method with Sidechain Mechanism. Framework and is divided into three subsections. Section 3.1 proposes attributes of a transaction and block construction. The proposed peer-to-peer query system architecture is explained in Sect. 3.2. Section 3.3 proposes a search algorithm. Section 4 proposes a web interface to search the land record data through the sidechain. Section 5 details the results and its analysis. Finally, Sect. 6 concludes this article.

2 Related work

In 2008 Satoshi Nakamoto invented bitcoin, and to make it useful in the real world, blockchain technology was introduced [6]. Blockchain technology is based on a Peer-to-Peer distributed file system. Zibin Zheng [7] discussed consensus, architecture, and future trends in blockchain as an immutable ledger that allows various transactions to occur in a decentralized way. A comprehensive overview of blockchain technology is provided, and a comparison of various consensus algorithms such as PoW, DPoS, PoS, and PBFT. Anasuya and Vishwas [8], in their research, explain that immutability, transparency, and integrity are provided by public blockchain to the transactions processed in the chain of blocks. Bitcoin and Litecoin are examples of the transfer of value from one person to another. Singh and Vardhan [9] have suggested an efficient consensus algorithm and proposed framework management for land registries based on blockchain. Their main focus was to provide a decentralized and reliable framework to manage the e-stamp and property registration process.

Bowen Ding and Yuedong Xu [10] proposed a decentralized query engine based on IPFS for flexible and collaborative query processing. They named their approach Minerva and demonstrated its capabilities, and made it available online. Singh and Vardhan [11] proposed an approach in banking sector transactions for the e-cheque clearance system based on DLT. This approach was based on permissioned blockchain, and it expands the range of clearing e-cheque from local banking to global banking. In this proposed approach, all the participating banks must join the framework to provide rapid cheque clearance to their customers. It was designed so that any participating bank could see the cheque issued by its customer. Singh and Vardhan [12] have proposed DLT based real estate transaction mechanism that emphasizes keeping the block size smaller for faster operations. Further authors in [13, 14] propose that if blockchain-based e-governance applications have to cater to people living in remote areas with little Internet connectivity, one has to compute optimal block size to increase the efficacy of the real estate transaction mechanism.

Vujičić et al. [15] has also worked in blockchain and mainly focus on bitcoins. Kumari and Kushwaha [16] have proposed CTES model for distributed systems for Kerberos-style authentication with some added functionalities. They also discussed client authentication and authorization. Vardhan and Singh [17] have also proposed a DLT based e-stamp procurement framework in 2018 that discusses its efficiency in the authentication of the stamp paper needed as fee in real estate transactions. Singh and Vardhan [18] have also worked in banking transactions and proposed a DLT-based solution for the electronic cheque clearing system. The strategy is built on top of a permissioned blockchain with emphasis on trust management and consensus.

For block validation, Kumar et al. [19] propose a consensus algorithm. A random block is generated and added to the existing blockchain with the condition that it is agreed upon by more than half of the nodes in that blockchain network. Their research established a blockchain architecture based on a one-way hash function to allow vehicles to access and store traffic-related data on the blockchain network. Mallick and Kushwaha [20] proposed an enhanced version of the Push protocol, dubbed Ex-push, for Pub-Sub Hubub (PUSH) and web pub-sub. Pippal et al. [21] propose an ascendible authentication approach based on CTES. Scalability can be further enhanced by implementing load balancing within the distributed system.

Cocco et al. [20] suggested that sustainable development and the blockchain's potential as a banking technology take into account the Bitcoin system. Daming et al. [21] discuss the blockchain's vulnerability to various interventions in a weak point analysis. According to the study, blockchain technology is more secure and resilient to existing intrusion attacks. The authors conduct an in-depth analysis of the technologies that comprise the blockchain. The blockchain evolved due to the convergence of several technologies and various types of blockchain architecture. Santander [22] proposes application of blockchain technology in conducting real-time trades. Schwartz et al. [23] propose several Byzantine generals consensus algorithms suitable for decentralized and distributed payment systems. It lays emphasis on Bitcoin's underlying technology, the bitcoin backbone protocol (BBP), and demonstrates how blockchain can be used as a database for the Internet of Things (IoT) applications. Yadav et al. [24] have proposed a scalable trust based property transaction mechanism that is based on DLT and emphasize the need of trusted miners in order to develop a highly reliable e-governance mechanism.

Back et al. [25] discussed that a sidechain is an additional chain associated via a two-way peg mechanism to the main blockchain. A Sidechain may have its consensus protocol that may be entirely dissimilar from the protocol of the mainchain. In theory, a sidechain may add some new functionality too so as to improve the privacy and security of the traditional blockchain.

The literature review demonstrates the breadth of blockchain technology across various domains and applications. After reviewing these studies, it is clear that peer-to-peer network architecture is required for a tamper-resistant, transparent, reliable, authentic, efficient, and fast land/property transaction system. This paper introduces the idea of "sidechains" in the blockchain to address some of the research gap in the literature review. Existing approaches only present proposals that are theoretical in nature, and implementation is not discussed. This article proposes a sidechain mechanism and blockchain-based land registry storage that aims at improving the performance of land record search.

From the studies discussed above, it can be seen that the load on the blockchain will steadily increase due to an increase in transactions. Further non transactional data also needs to be stored. Storing this huge data on the blockchain and performing efficient search operations requires efficient search algorithms. To achieve this, efficient searching/retrieval and storing of data is needed. Hence, there is a need for an efficient searching/retrieval and storage mechanism.

The blockchain provides generic and uniform blockchain architecture. The load on the blockchain will steadily increase due to increase in transactions. Storing this huge data on the blockchain and performing efficient search operations requires efficient search algorithms, which is scarcely researched. There is a need to reduce the blockchain overhead by storing all related data associated with transactions in the blocks in a separate memory area. There is also a need for high-end servers to cater to this load and support efficient search, thus increasing the scalability.

3 Proposed efficient storage method with sidechain mechanism framework

The Proposed method for the land registry system for storage of registry related documents is based on sidechain technology. A sidechain is a blockchain that runs parallel to the mainchain. It extends the functionality of an interoperable blockchain network. The term "sidechain" refers to a mechanism that enables tokens from one blockchain to be securely used in a completely different blockchain while still being able to be returned to the original chain if necessary. When certain data related to any transaction in the mainchain has to be stored inside the chain, a token is generated. It has a lifetime of a certain period only. It acts as a means to ensure integrity and security. By convention, the original chain is generally referred to as the "Mainchain (MC)." In contrast, any additional blockchains that allow users to transact with the mainchain's tokens are referred to as "sidechains." The amount of data stored in the blockchain should be limited to reduce the search and access time illustrated in Fig. 3.

- Header: Block headers identify individual blocks in a blockchain.
- MP: Merkle path of the sending TX to the Merkle root contained in the referred MC Header.
- Sending Transaction (TX): Non-transactional data is stored in a sidechain. Metadata stored in the mainchain (MC), MC stored only the block hash, is used as a reference.

Therefore, it is important to decide which data should be stored either in the mainchain or in the sidechain. In this paper, metadata is stored in the mainchain to be publicly accessible. Non-transactional data, such as pictures, contracts, PDF, and other allied information, should be stored in the sidechain. A hash or signature for the sidechain content is generated, and that is stored in the mainchain. Sidechain helps us overcome scale blockages by allowing miners to form a block more than the size proposed by the mainchain protocol. It also increases transaction speed and operational efficiency. The verification of the sidechain computation result is cheaper than the mainchain. To search a block stored in it, we also propose a search engine and a newer concept of a summary file is also proposed, which is connected to IPFS based blockchain using an application programming interface (API) where anyone within the network can perform searching operation on the mainchain. Time is recorded using the console, and comparison of search time is made with a linear search and

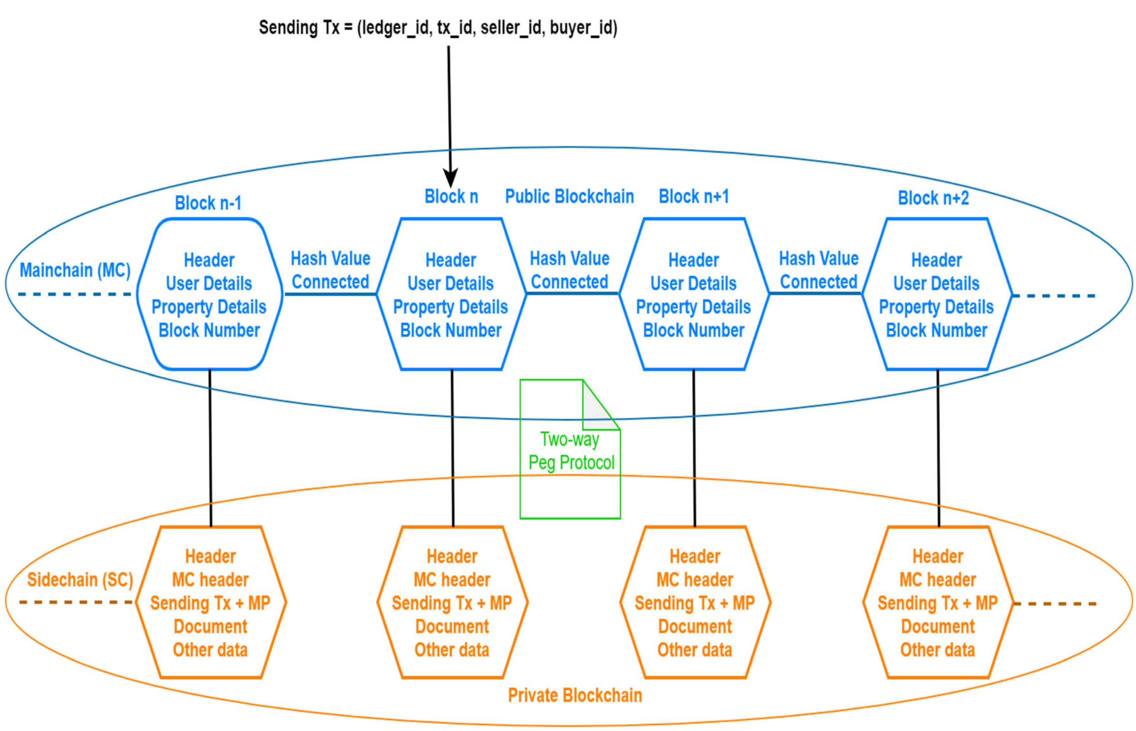

Fig. 3 Network architecture of the proposed sidechain mechanism

hash table-based algorithm and other land registry search algorithms (LRSA) are used.

A sidechain is a separate blockchain that is connected to its parent blockchain using a two-way peg. This is an emerging application that allows tokens and some other digital assets to be used securely from one blockchain to another and vice-versa if needed. Figure 3 represents how data is stored in IPFS using sidechain technology. Sidechain data as used in this proposed application is used to store non-transactional data, such as an image, pdf, contract papers, and hyperlinks. Sidechain data takes more space, and the mainchain cannot contain it because doing so will exceed its block size limit. These data are stored in a separate blockchain, which results in producing a hash value. The generated hash will be used as a referenced address, which is stored in the mainchain. Each referenced data has its content identifier (CID) that is the hash of the block storing that content. If it gets compromised or hacked, the loss will be accommodated within the chain itself and produce zero harm to the mainchain.

In this article, we propose an asset-transfer system for the land/property registry system that is based on sidechain technology. The mainchain can be queried by anyone, whereas the sidechains accept attribute-based queries from the parties involved in a land transaction or can be through authorization.

3.1 Attributes of land registry in a block

For developing a application specific blockchain, the block structure must be defined. This e-registry of the blockchain is used to maintain the details of the registered properties in the mainchain. All of the transaction's essential entries should be placed within the block. The different attributes of land registry transactions that are stored in the mainchain is illustrated in Fig. 4. A block contains information about the transaction, such as the buyer and seller's identities, property details, previous block's hash, etc. The transaction should be added to the pool of transactions. The leader miner creates a block consisting of few of these transactions when a sufficient amount of transactions are available in the transaction pool. The new block created is broadcast by a leader miner to all nodes. Then the consensus is achieved on the transaction block. Finally, the agreed block is added to the blockchain. The attributes of a particular transaction are as follows:

- Timestamp: Time of creation of the block.
- Previous Hash: sha256 hash of the block.

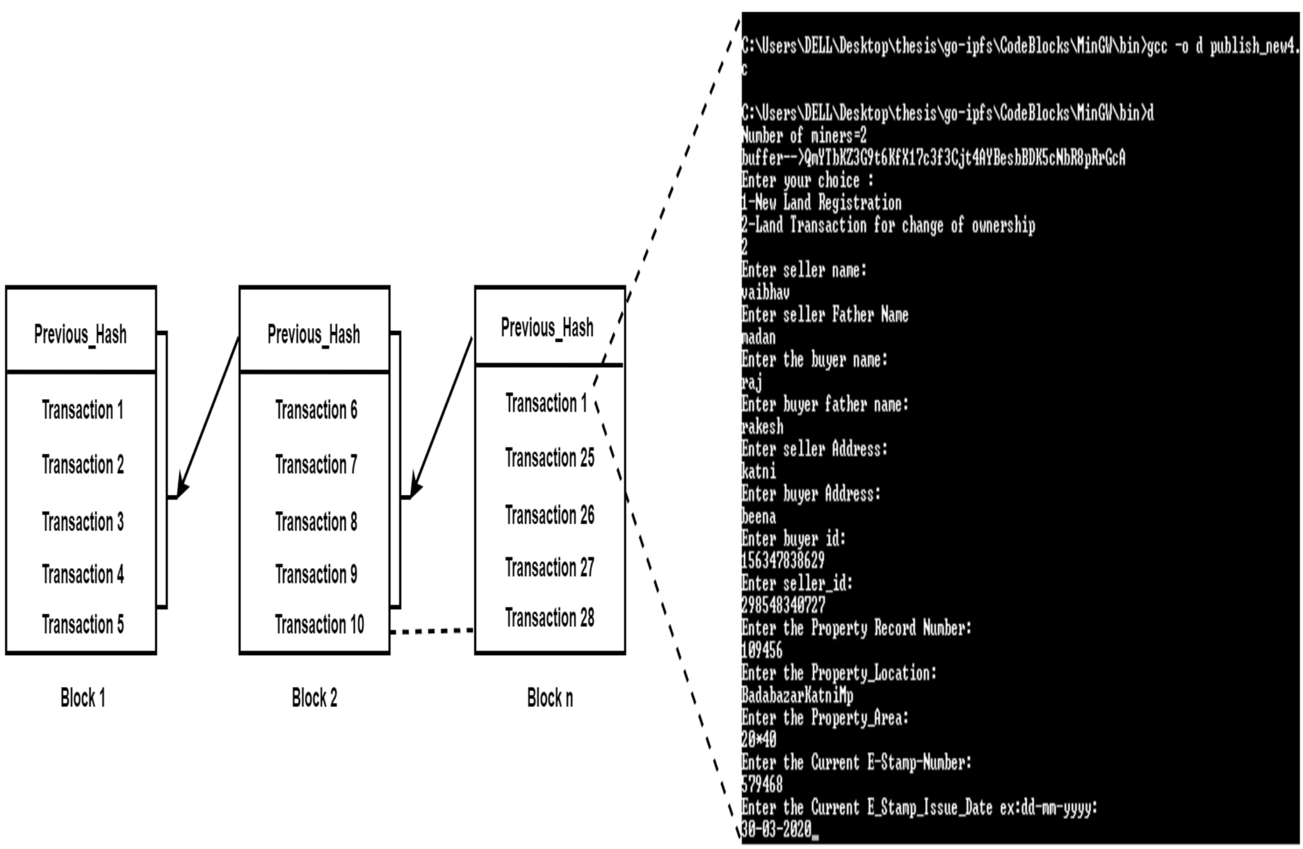

Fig. 4 Structure of property block record transaction

Table 1 Sample of property details are store in mainchain

Property number	Property location	Property area	E-stamp number	E-stamp issue date	Owner of property	Postal code of owner	Address of owner	Owner city	Owner state
101	Allahabad	800*800	200,796	20/07/96	Niharika	201,003	Sector L	Allahabad	U.P.
102	Ghaziabad	500*500	100,890	10/08/90	Ayushi	201,003	Patel	Ghaziabad	U.P.
103	Malihabad	400*400	200,720	20/07/20	Lavi	226,102	Ashiyana	Lucknow	U.P.
104	Noida	300*300	123,658	20/07/96	Amit	110,001	Sector 8	Noida	U.P.
105	Pakkabad	800*600	030918	03/09/18	Anvay	201,003	Dubgaga	Lucknow	U.P.
106	Lucknow	500*400	180,815	18/08/15	Anvi	211,004	Pakkabad	Lucknow	U.P.
107	Malihabd	900*900	231,568	08/09/75	Suman	226,012	Sector G	Lucknow	U.P.
108	Kanpur	200*400	120,392	12/03/92	Khushboo	208,011	Yashoda	Kanpur	U.P.
109	Delhi	600*600	2606/09	26/06/09	Adity	226,012	Malihabd	Lucknow	U.P.
110	Lucknow	700*700	150,895	15/08/95	Somya	206,122	Awas	Auraya	U.P.

- User Name: This gives the property seller/buyer name.
- User's Address: That specifies the user's location in the address.
- Property_Record_No: This is a unique number for the registered land.
- Area of Land: This field indicates the property space.
- Property Location: This is the location of Land/ Property. Contains the details of the land being sold.
- E-stamp Number: This is the permanent unique number of the property record.
- E-stamp Issue Date: Represent the change of ownership of property record.
- Digital_Signature_of_Authority: Signature of Issuing Authority property.
- Transaction ID: Unique ID of each transaction.
- Seller ID: Contains the pen details and Aadhar card number of the seller.
- Buyer ID: Contains the pan details and Aadhar card number of the buyer.

The block is a long JSON text file. We are using InterPlanetary File System (IPFS) for storing our blocks. IPFS is a peer-to-peer distributed file system that seeks to connect all computing devices with the same file system. Whenever we store a block in IPFS, it returns a hash, which is dependent on the content of the block, and its hash can address the block. This makes IPFS very ideal as a file system for storing our blockchain. A sample block with a single transaction looks like this:

{"index":73, "timestamp": "Sun May 9 21:47:08 IST 2021", "transactions":/{"sellerId": {"panDetails":"28,740,432", "aadharNo": "236985698745","name": "Anvi"}, "buyerId": {"panDetails": "95,656,858", "aadharNo": "129,665,239,658", "name": "Lavi"}, "landDescription": {"length":28, "width":29, "address": "Ashiyana, Lucknow"}, "price":1507", "transactionId":3000, "pdfLink": "tr.pdf"}/, "prevHash": "12206DC8A3B67234B97CB07CF63504E15A9604C2FDF66DF38531F0A2D3772BD86"}

Hash generated after storing the non-transactional data is used as a reference that serves as the path of that data stored in the local environment or the connected peer's storage environment and is stored in the mainchain. Table 1 shown the sample of the property registry and ownership record of the data stored.

3.2 Proposed peer-to-peer query system architecture

This proposed framework is primarily intended for making a fast parallel query in a blockchain storing details of land registries. Although we are storing land registries, this is a generic approach and can be used with any blockchain. The web service, which provides a user-friendly interface to query the blockchain, shows how this faster query system can be used in a real-world application.

The proposed DLT based secure land registry transaction framework is swarm-based on the peer-to-peer network implemented on IPFS. When IPFS is initialized, it generates a key pair and creates a local folder that contained the IPFS configuration and a repository object from a node. The identity of each node is determined using a hash of the node's public/private key pair.

The work proposes sidechain-based swarm architecture shown in Fig. 5. The proposed P2P query search engine is built upon the collaboration of four main integrates that are:

1. ***Summary File***: It is a data management tool where users will perform query operations. For fast queries in IPFS, we maintain a excel file in each peer that references CIDs and helps decrease the query time.
2. ***Search Engine***: Stores and organizes content during crawling and returns the result. A search engine is a

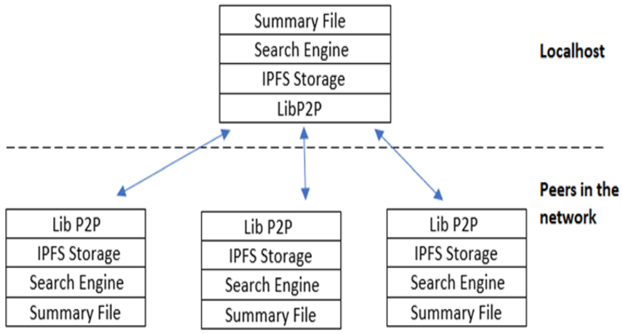

Fig. 5 Peer-to-Peer query system architecture

framework that permits a client to look for data on a system. While progressing into a P2P, decentralized arrangement, these procedures should be acclimated to suit the arrangement's adjustment.

3. ***IPFS storage***: Data is stored in the mainchain where IPFS uses a content addressing mechanism to uniquely identify each file in a global namespace connecting all computing devices. Instead of using a central server, IPFS uses a decentralized system of peers who has possession of some data, thus creating a robust file storage system and sharing. Any peer connected to the network can access a file using its content address.
4. ***LibP2P***: Libp2p is a modular system of protocols, specifications, and libraries that enable P2P network applications.

3.3 Proposed summary file based search algorithm

The Search time of the proposed summary file based search algorithm is elaborated in Sect. 4. It is being compared with traditional existing approaches, such as linear search [26] and hash table based search [17] under experimental setup for ascertaining the efficiency of the proposed algorithm. To access query using sidechain, the algorithm works as follows:

3.3.1 Linear search algorithm

Linear search [26] sequentially checks each row for the queried attribute one by one until a match has been found or every element has been checked in the summary file as per Algorithm 1.

Algorithm 1: Linear Search Algorithm

Input: val, where val is a random value entered to perform searching from a given sample.
Output: Hash of queried block.
Procedure: Linear Search Algorithm
    if user authentication is successful then \\ authenticate using client_secret.json
        load sheet
        for each property_record **in** row
            if property_record **matches with** val **then**
                return hash of property_record
            end if
        end for
    end if
end procedure

3.3.2 Searching ledger using hash table based

A hash table is maintained for each attribute in this approach. When we query for an attribute, the hash is searched for the matching key. If the hash table contains the search key, the corresponding block details are displayed. If the key for the search does not exist in the hash table, it indicates that the value was not found as per Algorithm 2.

Algorithm 2: Ledger Based Search Hash Table

```
Input: survey_id (land number allotted after survey)
Output: resultJSON
Procedure Hash Table Based Search
if user_id is authentic then \\ authenticate using client_server.json
      load hashTable from transactionDetialSheet of each block
      result={}
      if survey_id exist in hashTable then
            if any block is not visited then
                        result1=send_search_request(survey_id,user_id)
                                  \\ send request to other nodes and wait for reply
                        result=result union result1
      else
            if any block is not visited then
                  result1=send_search_request(survey_id,user_id)
                                  \\ send request to other nodes and wait for reply
                  result=result union result1
 return result
End Procedure
```

	A	B	C	D	E
1	**Seller_ID**	**Buyer_ID**	**Property_Record_No**	**E-stamp No.**	**Hash**
2	S1	B2	100	111111	QmNbLoV6e9Q2KBHWEbXg5nPLvwkhHL8DjAbBNDiMwwAoAi
3	S2	B3	101	111112	QmVYY92W9DWEk6DQKtPXxAzQU6RCmULfdpYkP9yjzWKcWG
4	S1	B1	102	111113	QmZZhAZurWdQrfFXf6GY7y4Zwg2aWu9knJ4PBGePhz3zxX
5	S3	B4	104	111114	QmcoDhx8gFzjVbBcsqBVVBmPqMzWe4ST4L3ZujoU9o96dr
6	S4	B5	105	111115	QmT2R2zsMUt6HX6nMWQXqVGFCntDod5qTVSfGhtpJ74guN
7	S6	B7	106	111116	QmR9MZtN6CbgrmYQa3PmgvN5o8K3MnpA71QbsW9bCLJNnb
8	S7	B8	107	111116	QmRMLvK6m5cERtAmYKKWet8LdjSp2GfLSeV9mR1fCr6MRh
9	S7	B1	108	111117	QmfG57xfp7E4DZNK2tRDGLabxjfXeCJGPNBzotZVkGDLgV
10	S8	B6	109	111118	QmNa1nSs9ec5dEEjvQZN6yHULJ8HD1RajRcbSgWk6zeek3
11	S9	B8	110	111119	QmZsEcUQsdSRHYpjejLh4awrvsdazTeDhGJD7F9VRgDYjX
12	S10	B10	111	111119	QmQbVE9MD73MvSf7keT3pzBwpZq1hiJJinWFsPKMnvYeyh
13	S11	B9	112	111121	QmQpV1Cn8ZJmf9VysrH8Mr8qDwMRLCdC1WH8ckwe84KuTm
14	S11	B9	113	111121	QmdK7CYqctumrzV7mRfFG3EZPYH4xFhvTdYfsoW2RbPiTY
15	S12	B12	114	111121	Qmbbtknq5xMFXbiHmfuLgP5hHR1ZgkG8ATHpTvEMsknpXU
16	S13	B13	115	111122	QmQpg95qPxpVC631dGgUVZEXxEhzfj4WKbTsiQUE3RKeLc
17	S14	B14	116	111123	QmQbVr1tsei8nkUDm6JYUmaAUreK1DucWRoiBtP5Xe9qQL
18	S15	B15	117	111124	QmWbScDs1nt1a86jisJQcvXhDhF7VnwJ1kKqzGRpBTfSLx
19	S18	B18	118	111118	QmRfCWnXJAMwPX1PLnufWuad9UYzWuDUCMvVDcfZFEwxUm
20	S19	B19	119	111119	QmNirxYAVorEzDt8foZzBxHKrSUhX23RaiRDWsyFKzJjrh
21	S20	B20	120	111120	QmfNM5j45darmiwURR3GMGdryKgtMabGGJBWpasSJHVdQL
22	S21	B21	121	111121	QmTAWUrnGe3AqqBU4HEgXrt9JsexADPinH6jrqKKuvMmmF
23	S22	B22	122	111122	QmQ1BuD3T5CG9SZ7JkHNeNcNaQEHBurWNTam4xNxNkhMGj
24	S23	B23	123	111123	Qmem2AxmukumvHZyWZHNEQ9BZt7erLzEFVwWz923FrCx6U

Fig. 6 Summary file is containing the meta-data of the land record

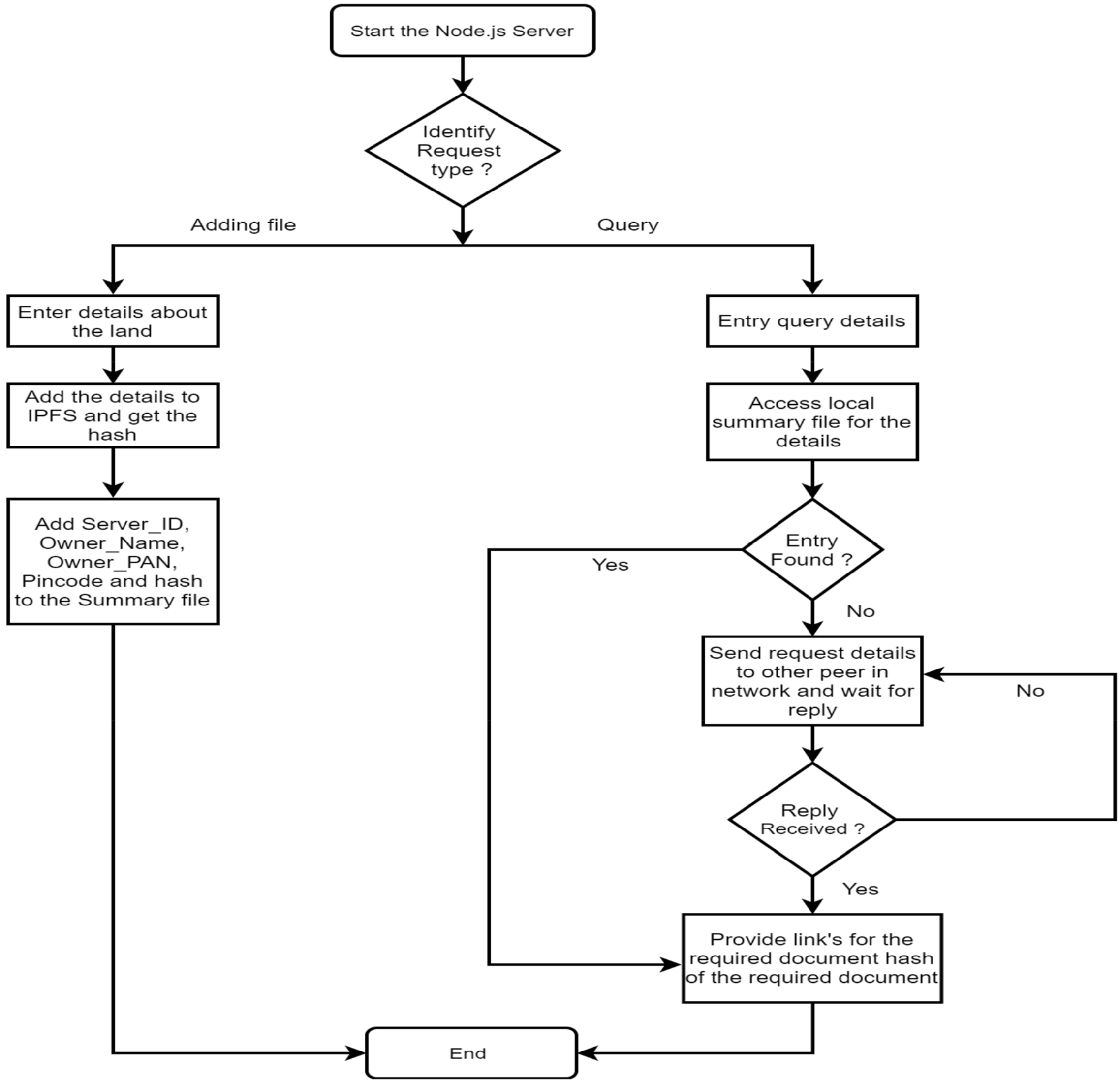

Fig. 7 Flow chart of query and adding the file

4 Implementing property record search using summary file

A decentralized framework is implemented by adopting various hardware and software tools. An attempt has been made to increase the efficiency and reduce the searching of land records in the blockchain in the proposed summary file based search algorithm that uses mainchain as well as side chain. IPFS platform is used to store non-transactional data in the sidechain. The hash obtained by adding non-transactional data is used as a reference and is stored in the mainchain. There are the following different modules of the proposed framework:

1. Summary file.
2. Querying and adding the records of land.
3. The web interface.
4. Proposed Web Server based LRSA.

4.1 Summary file

In this module, we have used a summary file for faster execution of queries in IPFS. We maintain a file in each peer. The summary file is used to fetch property record data faster from the database. The summary file contains only necessary attributes that help search. Figure 6 shows the representation of the summary file. The summary file is local to each node, and each entry has the attributes namely Seller_ID, Buyer_ID, Property_Record_Number, E-Stamp_Number, and hash. This helps us to resolve the query is a fast and efficient method. This file is connected to the node.js server using the peer id of the summary file. To link a summary file with the IPFS, an API of blockchain is created using a node.js server.

4.2 Proposed algorithm for querying and adding the records of land

This article proposes a query algorithm for searching and adding the land record in the blockchain. There can be many queries at any time, so retrieving a query should be fast, and for this specific purpose, we introduce the concept of a summary file. The summary file contains an entry required to get the corresponding hash for the particular entry shown in Fig. 6. To access the summary file, client secret.json file is required, and it must also be present, and it must be ensured that any unauthorized persons don't get access to the file. The query system is open for all, and as such, we do not need any authorization to use the system. When the query is fired, the response must be sent to the user within a specific acceptable time and ensure this ensures that the user does not wait forever for a reply. The user must get the response or the error message with sufficient detail to understand the issue and correct it. While adding an entry of land, a proper check must be made to let the user know if he is making any mistake and save from an invalid entry being stored. It is also required to make sure that the corresponding entry in the summary file is also being made as soon as the entry is made into the IPFS cluster. It is required that the query is first searched in the local summary file, and if entry is not found, it must be broadcasted on the network so that the query result can be found. If the query is using the owners PAN or owner name and pin code, it is required that the query is searched in a local summary file and broadcast on the network as the same person can own multiple lands.

We insert Transaction_Time, Transaction_ID, Property_Record_Number, Property_Location, Property_Postal_code, Property_Area, Seller_Name, Seller_Father_Name, Seller_Address, Seller_City, Seller_State, Seller_Postal-code, Seller_Pan, Buyer_Name, Buyer_Father_Name, Buyer_Address, Buyer_City, Buyer_State, Buyer_Postal_Code, and Buyer_Pan. After submitting this data, the hash is returned on the interface from the Node.js server. By clicking this hash, it will retrieve the land record. We can do queries by Owner Property_Record_ Number, Owner_Postal_Code, Owner_ID, and Postal_Code in querying land interface records. The land record is retrieved from IPFS through the Node.js server (Shown in Fig. 7).

4.2.1 Algorithm of query in summary file or adding the property record

The proposed approach has been implemented in the IPFS that runs on a Windows 10 operating environment. A decentralized framework is implemented by adopting various hardware and software tools. This proposed algorithm is elaborated in Algorithm 3.

Algorithm 3: Working of Query and Searching Property Record

1\. **Input:** Server, Node.js, addFile, query
2\. **Output:** LinkOfRequiredDoc or HashOfRequiredDoc
3\. Server.Start(Node.js)
4\. **If** request **is** addFile
5\.   enter LandDetails in Details
6\.   Add Details to IPFS
7\.   Generate Hash Details and set to Hash
8\.   Add Server_ID, Owner_Name, Owner_PAN, Pincode, Hash to SummaryFile
9\. **Else if** request **is** query
10\.   **If** Enter Query to search file in SummaryFile **then**
12\.   **return** LinkOfRequiredDoc or HashOfRequiredDoc
13\.   **Else**
14\.   set Reply to Peer.Request(Enter Query)
15   **If** (Reply == true)
16\.   return LinkOfRequiredDoc or HashOfRequiredDoc
17\.   **Else**
18   Goto Step 12
19   **End if**
20\. **End if**

Figure 8 describes the working of the query system. To perform searching, a user starts node.js server, and a message gets displayed on the console about the port running the server. One can perform searching using the said attributes. When searching is enabled, it initially searches in its local environment, and if not found, then the required detail is sent to the connected peers, and the reply is received with a hash of the mainchain on the same port. This search engine listens to the query and replies to the same port and console result. With the help of peers in the network, one can easily fetch information using query in the mainchain, but only authenticated and authorized users will access data from the sidechain. This proposed algorithm increases the search engine efficiency achieved by using a summary file. This helps us to resolve this query in a faster and efficient way. In this method, multiple users can query in the sidechain, which initially fetches local data and, if unsuccessful, then send the required details to other peers in the connected network.

4.3 The web interface

To easily query a transaction ID and the PDF link of a land deal corresponding to a particular ID number, we have made a web interface illustrated in Fig. 8. We use JSP so that the call to the backend Java routines can be made quickly. We have provided a search bar which takes seller/buyer ID as its input and outputs the transaction ID and the PDF link related to that ID. In the backend, we have used our algorithm to make this query faster.

4.3.1 Proposed web server based land registry search algorithm (LRSA)

In order to establish efficacy of sidechain and mainchain we make use of Google developer console to create a project that also uses Google API services for fetching records from the Google sheets that store the summary file as illustrated in Fig. 6. A web interface is created where in PB provides the property record number and based on this block hash is fetched and this is used by the ROM to fetch

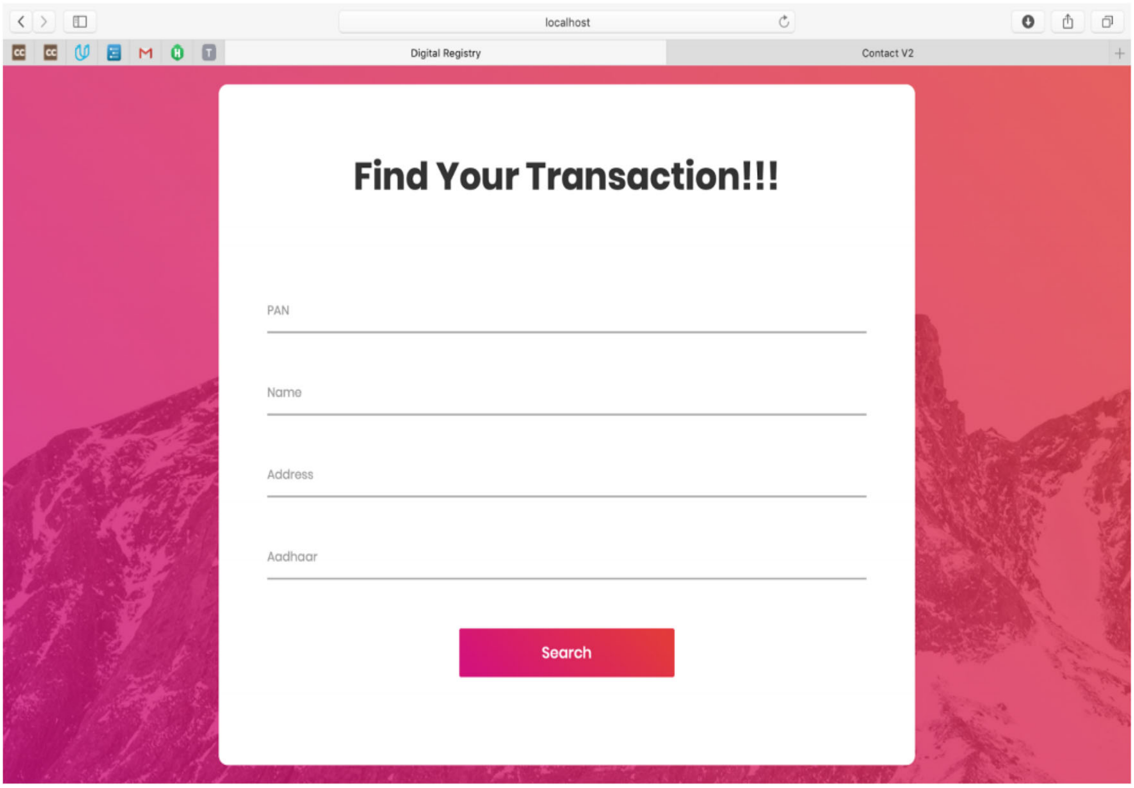

Fig. 8 Interface to search land registry database

the required block that contain the property documents as illustrated in Fig. 8.

The proposed LRSA is similar to the hash table-based method. We map the blocks and the corresponding transaction with the help of the Merkle tree. To show the output, we have used the AJAX call on the index page. The user can enter any supported ID of the buyer/seller as per the algorithm. It can be his PAN ID, his address, his Aadhaar card number, or his name. On requesting, Ajax sends the request containing IDs to the JSP file, and for each non-null ID, it subsequently calls the backend java program. This backend Java program queries the blockchain. The blockchain may or may not be stored on this server. If the webserver is a peer of the blockchain, it queries it internally and sends back the result. If not, it asks its neighbouring machines in the IPFS cluster for the blocks and queries. The intersection of results corresponding to each ID is taken. The results are sent back in JSON format. Now from this JSON, we dynamically represent the result in table format on the index page. The machine on which the webserver is deployed needs not be a peer, but to avoid network latency, it is highly recommended that a peer only be made a web server. The proposed land registry algorithm discussed below:

Table 2 Hardware and software specification

Server specification	
Processor	Intel Xeon X5355 @ 2.66 GHz (8 Cores)
Memory	8 GB RAM
Storage	4 * 146 GB
Network interface	1GbE Data Rate Per Port
Operating system	Windows
Tool	IPFS 0.5.0
Simulation parameters	
Block hash size	256 -bit
Blocks	10, 20, 30, 50, 100, 150, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900, 1000 (Number of Nodes)

Algorithm 4: Land Registry Search Algorithm (LRSA) Through Users Details

1. **Input:** searchParameters, leadingHash, noOfThreads
2. **Output:** resultlnJson
3. **Procedure**: Search(searchParameters, leadingHash, noOfThreads)
4. set null to queryResult
5. set resultPan from TransactionDetails(searchParamters.panID, leadingHash, noOfthreads) using multiThreadedQuery
6. set resultAadhaar from TransactionDetails(searchParamenters.aadhaarID, leadingHash, noOfthreads) using multiThreadedQuery
7. set resultAddress from TransactionDetails (searchParamenters.addressID, leadingHash, noOfthreads) using multiThreadedQuery
8. set resultName from TransactionDetails (searchParamenters.nameID, leadingHash, noOfthreads) using multiThreadedQuery
9. set queryResult as resultPan;
10. queryResult intersection resultAadhaar
11. queryResult intersection resultAddress
12. queryResult intersection resultName
13. **return** queryResult.convertToJson();
14. **End Procedure**

5 Performance and results analysis

The performance of the proposed LRSA is computed in terms of validity and robustness. The proposed land registry management system has been implemented in the IPFS installed on a Windows 10 machine. We implement a decentralized framework by adopting various hardware and software tools, as summarized in Table 2. For our test purpose, we randomly initialized multiple blockchains and query these. We use Java APIs provided by IPFS to interact with the IPFS server and JSON API by google to handle JSONs efficiently.

This article illustrates a real-world scenario of the query system in IPFS technology. In this proposed work, we have used the IPFS platform for storing data in the mainchain. We have also used node.js concepts to connect a summary file using the API of blockchain to perform a query in the sidechain. Google sheets is used to handle JSONs efficiently. The proposed work performs queries on attributes having data types as integer and string. We test and record the experimental results from by increasing the server's load by putting more details in the summary file. The readings recorded are shown in the table when searching is performed using Property_Record_No.

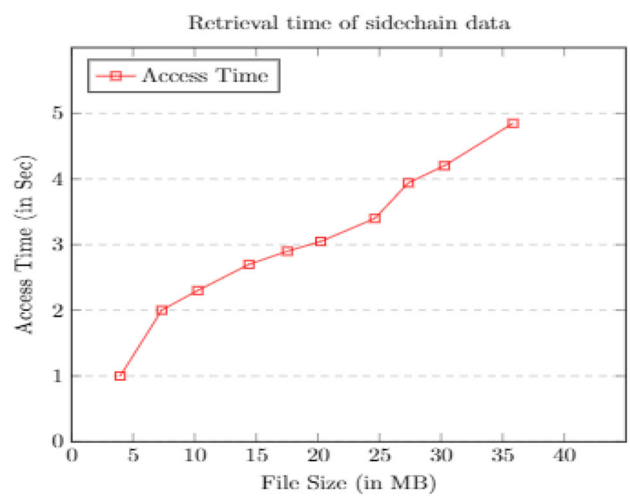

Fig. 9 Access time vs. file size of sidechain

5.1 Retrieval of sidechain data

This section records the search results when summary file is not maintained and used for the search operation. In Fig. 9, one can note that the time taken to access the sidechain data is linear and increases steadily with an increase in the file size. The searching approaches need to traverse the sidechain to retrieve the data information about the land and users.

Table 3 Time comparison between linear search, hash table search, and proposed approach (in ms)

Sr. No.	Number of nodes	linear search	Hash table search	Proposed approach (LRSA)	Diff. linear search and hash table	Diff. linear search and proposed approach	Proposed approach improvement (%) wrt linear search	Proposed approach improvement (%) wrt hash table search
1	10	12.70	6.90	3.94	8.76	2.96	68.97	42.89
2	20	13.40	7.10	4.45	8.95	2.65	66.79	37.32
3	30	14.69	7.80	5.35	9.34	2.45	63.58	31.41
4	50	15.90	8.05	5.97	9.93	2.08	62.45	25.83
5	100	17.23	8.50	6.49	10.74	2.01	62.33	23.64
6	150	18.31	9.15	7.02	11.29	2.13	61.66	23.27
7	200	19.49	10.15	7.63	11.86	2.52	60.85	24.82
8	250	20.79	10.80	8.23	12.56	2.57	60.41	23.79
9	300	23.17	11.96	8.71	14.46	3.25	62.40	27.17
10	350	25.23	12.79	9.33	15.9	3.46	63.02	27.05
11	400	27.23	13.70	10.45	16.78	3.25	61.62	23.72
12	500	29.98	14.55	11.05	18.93	3.5	63.14	24.05
13	600	33.10	15.99	11.86	21.24	4.13	64.16	25.82
14	700	35.89	17.25	12.91	22.98	4.34	64.02	25.15
15	800	38.23	18.73	13.89	24.34	4.84	63.66	25.84
16	900	40.99	19.83	15.05	25.94	4.78	63.28	24.10
17	1000	43.19	21.35	16.25	26.94	5.1	62.37	23.88

5.2 Comparison with traditional approaches

The proposed land record search approaches searching time is compared with traditional approaches, such as linear search and hash table-based search. The comparative results have been shown in Table 3. Table 3 compares the traditional approaches—linear search and hash table-based search with LRSA based on searching the record of the experiments. The number of nodes is incremented from 10 to 1000, and the searching time is noted. The tabulation is for the results obtained from the different simulations for computing execution time.

The graphical representation in Fig. 10 shows the number of nodes varying from 10 to 1000. Compared to the linear search and hash table-based search approaches, the proposed search algorithm—LRSA, performs better. The results are able to establish that the proposed approach

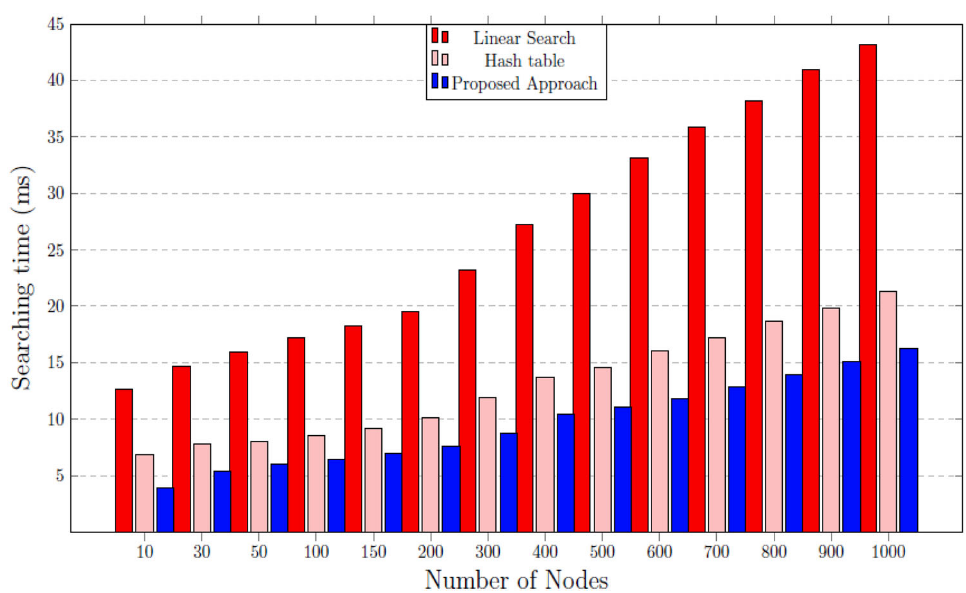

Fig. 10 Performance analysis of time required to search the property records (in ms)

Table 4 Block approval time during consensus process of PoW, PoS, DPoS, TrVCA, and proposed LRSA (in ms)

S. No.	No. of node (N)	PoW in (in ms)	PoS in (in ms)	DPoS in (in ms)	TrVCA in (in ms) [27]	Proposed LRSA (in ms)	Diff. PoW and LRSA (in ms)	Diff. PoS and LRSA (in ms)	Diff. PoS and LRSA (in ms)	Diff. TrVCA and LRSA (in ms)	Proposed LRSA (%) wrt PoW	Proposed LRSA (%) wrt PoS	Proposed LRSA (%) wrt DPoS	Proposed LRSA (%) wrt TrVCA
1	10	129.8	106.2	88.3	57.1	45.8	84	60.4	42.5	11.3	79.09	56.87	48.13	19.78
2	20	186.2	149.6	125.6	80.4	70.2	116	79.4	55.4	10.2	77.54	53.07	44.10	12.68
3	30	207.6	183.4	153.2	93.4	84.4	123.2	99	68.8	9	67.17	53.98	44.90	9.63
4	50	296.4	225.3	198.7	114.5	103.3	193.1	122	95.4	11.2	85.70	54.15	48.01	9.78
5	100	358.2	283.7	252.4	149.6	136.4	221.8	147.3	116	13.2	78.18	51.92	45.95	8.82
6	150	416.6	319.6	297.6	184.3	172.1	244.5	147.5	125.5	12.2	76.50	46.15	42.17	6.61
7	200	476.9	376.3	341.3	223.6	211.5	265.4	164.8	129.8	12.1	70.52	43.79	38.03	5.41
8	250	537.3	408.4	383.6	265.1	251.3	286	157.1	132.3	13.8	70.02	38.46	34.48	5.20
9	300	601.6	458.7	433.2	303.7	289.2	312.4	169.5	144	14.5	68.10	36.95	33.24	4.77
10	350	663.7	505.3	477.4	348.2	332.7	331	172.6	144.7	15.5	65.50	34.15	30.31	4.45
11	400	725.1	553.5	525.5	388.6	369.2	355.9	184.3	156.3	19.4	64.29	33.29	29.74	4.99
12	500	784.9	604.8	576.9	431.4	413.5	371.4	191.3	163.4	17.9	61.40	31.63	28.32	4.14
13	600	848.6	652.7	624.8	470.6	457.9	390.7	194.8	166.9	12.7	59.85	29.84	26.71	2.69
14	700	906.5	703.5	674	512.5	487.3	419.2	216.2	186.7	25.2	59.58	30.73	27.70	4.91
15	800	970.7	757.7	724.8	555.1	536.5	434.2	221.2	188.3	18.6	57.30	29.19	25.97	3.35
16	900	1033.2	809.3	773.6	596.8	577.2	456	232.1	196.4	19.6	56.34	28.67	25.38	3.28
17	1000	1093.8	858.6	824.8	639.3	616.1	477.7	242.5	208.7	23.2	55.63	28.24	25.30	3.62

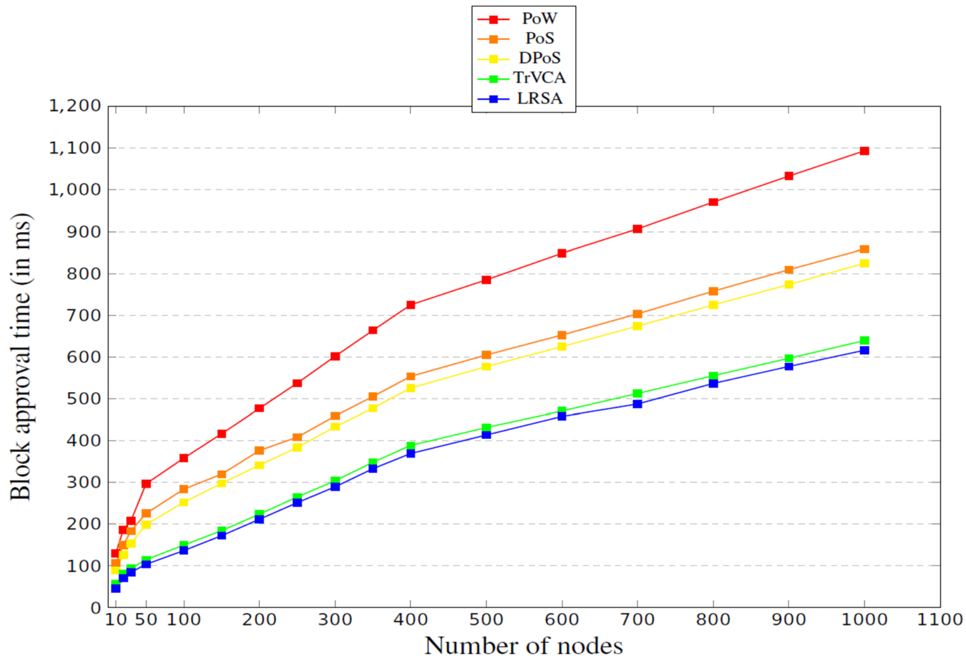

Fig. 11 Block approval time vs. number of nodes (in ms)

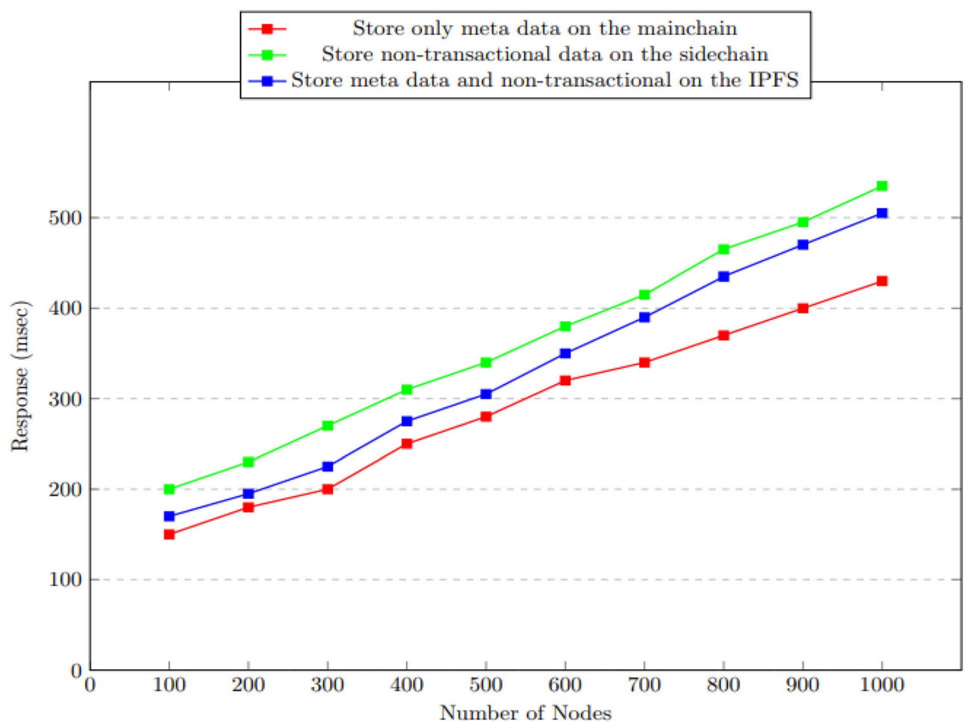

Fig. 12 Performance between the storage of data response time in blockchain and IPFS

namely LRSA reduces the searching time on an average by 63.21% and 27.04% as compared to the linear search and hash table-based search approaches. Thus, as compared to the traditional approach, the proposed LRSA performs better.

5.3 Comparison with traditional consensus approaches

Table 4 compares the traditional methods for block approval time during consensus process for PoW, PoS, DPoS, TrVCA and proposed LRSA approach under experimental setup. The block approval time during

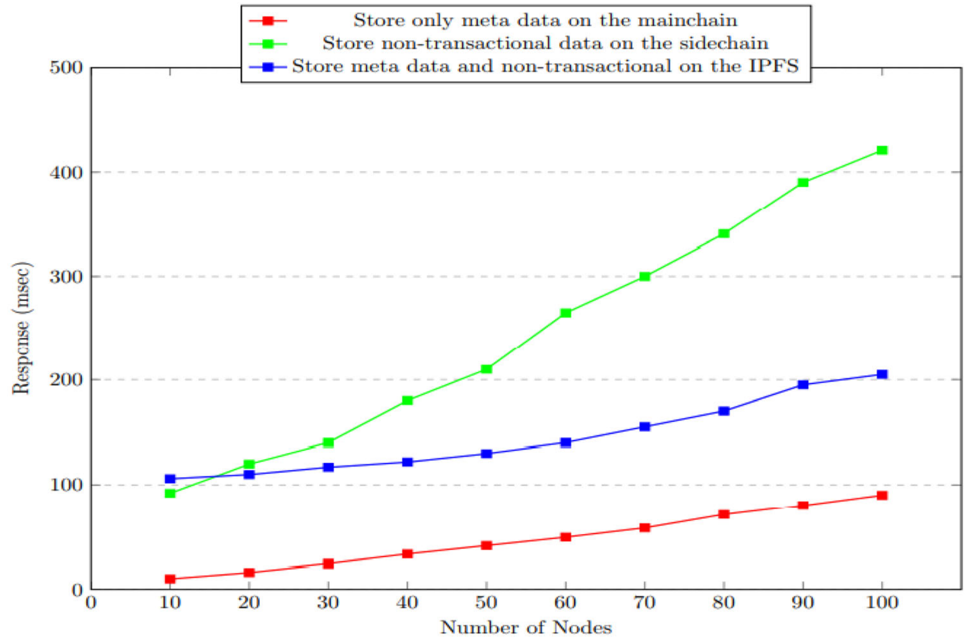

Fig. 13 Performance between the query response time searching the record blockchain and IPFS

consensus process of the PoW, PoS, DPoS, TrVCA, and LRSA approach are shown in Table 4. The node count is increased from 10 to 1000, and the block approval time during consensus process is recorded. The results of various simulations used to determine the block approval time during consensus process are recorded.

Figure 11 depicts a graphical representation of the number of nodes ranging from 10 to 1000. The proposed consensus algorithm namely LRSA performs better than the PoW, PoS, DPoS and TrVCA consensus algorithms. The results establish that the proposed approach namely TrVCA reduces block approval time during consensus process by 55.63%, 28.24%, 25.30%, and 3.62% respectively, compared to PoW, PoS, DPoS, TrVA and LRSA approaches. Hence the proposed consensus algorithm LRSA performs better than the existing approach and can be used for other similar applications.

5.4 Response time for data storage and searching record

The execution time and discovery time proposed in this section can be compared with the mainchain, sidechain, and IPFS approaches. Figure 12 shows the number of nodes from 100 to 1000, and Fig. 13 shows the number of nodes from 10 to 100.

Figure 12 illustrates that the response time for add land record data on the mainchain increases. This figure reduces the data storage mainchain, so we reduce the query response time because the proposed framework application develops two different blockchain types: mainchain and sidechain. Mainchain stores the metadata to be publicly accessible, and non-transactional data is stored in the sidechain. So, the data store in a sidechain does not require a consensus mechanism to validate data and synchronization. The data store mainchain to sidechain generates a token that can validate and verify the data storage in the sidechain and synchronization. As the non-transactional data is increased, then the non-transaction data store in a sidechain. The figure shows the red line is mainchain, blue IPFS, and green sidechain response time.

Figure 13 depicted how query response time varies as the record of data increases. As is the case with storage, we are reducing the amount of data to improve query performance. When the mainchain and sidechain frameworks are used, a query consists of two steps: querying the mainchain and querying the sidechain. There is an additional step in comparison to the traditional query method. Compared to the traditional method, the proposed framework's mainchain and sidechain model advantage eliminates the mainchain's storage problem.

6 Conclusions

This article proposes an efficient storage mechanism for property e-registry documents using sidechain and searching traditional property records in a blockchain based framework. It also proposes use of a summary file that speeds up the lookup operation when any PB wishes to enquire about the intended property. The registry office searches the record in the mainchain using summary file for getting block hash and the property record number. Next, the registry office searches the hash address of the block in the sidechain to access the property record. It accesses the related records based on the property record number, and provides the related record to the buyer. The proposed approach outperforms the traditional approaches such as linear search and hash table-based search in terms of retrieval time. The result establishes that the proposed approach namely LRSA reduces the search time on an average by about 60% and 25% respectively as compared to linear search and hash table based search approaches. With increase in number of miner nodes, the increase in search time of the proposed PSA is not as severe as the other two algorithms. This establishes its scalability. The proposed algorithm LRSA reduces block approval time during consensus process by 55.63%, 28.24%, 25.30%, and 3.62% while comparing it with the existing leading algorithms PoW, PoS, DPoS, TrVCA, and LRSA respectively.

There are also a few disadvantages to the blockchain, such as there is no generic and uniform blockchain architecture. The optimum block size can be first derived and later an application could be built to enhance the processing capability further. There is a need to reduce the blockchain overhead by storing all related data associated with transactions in the blocks in a separate memory area. There is also a need for high-end servers to cater to this load and support efficient search, thus increasing the scalability.

Author contributions ASY study design performed the experiments and wrote the manuscript. Moreover, other authors NS and DSK, guided preparing the manuscript and experiments.

Funding No funding was received.

Data availability My manuscript has no associated data.

Declarations

Ethical approval The manuscript in part or in full has not been submitted or published anywhere.

References

1. R, Indian states look to digitize land deals with blockchain, Reuters. (2017) https://www.reuters.com/article/us-india-land rights-tech/indian-states-look-todigitize-land-deals-with-block chain-idUSKBN1AQ1T3. Accessed May 20, 2021
2. Halpin, H., Piekarska, M.: Introduction to security and privacy on the blockchain. In: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, Paris (2017)
3. Yadav, A.S., Kushwaha, D.S.: Digitization of land record through blockchain-based consensus algorithm. IETE Tech. Rev. (2021). https://doi.org/10.1080/02564602.2021.1908859
4. Yadav, A.S., Agrawal, S., Kushwaha, D.S.: Distributed Ledger Technology-based land transaction system with trusted nodes consensus mechanism. J. King Saud Univ. Comput. Inform. Sci. (2021). https://doi.org/10.1016/j.jksuci.2021.02.002
5. Yadav, A.S., Kushwaha, D.S.: Query optimization in a blockchain-based land registry management system. Ingénierie des Systèmes d Inf **26**(1), 13–21 (2021)
6. Nakamoto, S., Bitcoin, A.: (2008). A peer-to-peer electronic cash system. Bitcoin–URL: https://bitcoin.org/bitcoin.pdf
7. Zheng, Z., et al.: An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData congress). IEEE, Honolulu (2017)
8. Patil, V.T., Acharya, A., Shyamasundar, R.K.: Landcoin: a land management system using litecoin blockchain protocol. In: Proceedings of the Symposium on Distributed Ledger Technology (2018)
9. Singh, N., Vardhan, M.: Distributed ledger technology based property transaction system with support for iot devices. Int. J. Cloud Appl. Comput. (IJCAC) **9**(2), 60–78 (2019)
10. Ding, B., Xu, Y: Minerva: Decentralized Collaborative Query Processing over InterPlanetary File System
11. Singh, N., Kumar, T., Vardhan, M.: Blockchain based e-cheque clearing framework. Scalable Comput. Pract. Exp. **20**(3), 511–526 (2019)
12. Singh, N., Vardhan, M.: Digital ledger technology-based real estate transaction mechanism and its block size assessment. Int. J. Blockchains Cryptocurrencies **1**(1), 67–84 (2019)
13. Singh, N., Vardhan, M.: Computing optimal block size for blockchain based applications with contradictory objectives. Proc. Comput. Sci. **171**, 1389–1398 (2020)
14. Singh, N., Vardhan, M.: Multi-objective optimization of block size based on CPU power and network bandwidth for blockchain applications. In: Proceedings of the Fourth International Conferenceon Microelectronics, Computing and Communication Systems. Springer, Singapore (2021)
15. Vujičić, D., Jagodić, D.J., Ranđić, S.: Blockchain technology, bitcoin, and Ethereum: a brief overview. In: 2018 17th International Symposium Infoteh-Jahorina (Infoteh). IEEE, East Sarajevo (2018)
16. Kumari, A., Kushwaha, D.S.: Kerberos style authentication and authorization through CTES model for distributed systems. In: International Conference on Information Processing. Springer, Berlin (2011)
17. Singh, N., Vardhan, M.: Blockchain based E-stamp procurement system with efficient consensus mechanism and fast parallel search. J. Mech. Contin. Math. Sci. **13**(4), 73–89 (2018)
18. Singh, N., Kumar, T., Vardhan, M.: Blockchain-based e-cheque clearing framework with trust based consensus mechanism. Cluster Comput. **24**(2), 851–865 (2021)
19. Kumar, A., Yadav, A.S., Kushwaha, D.S.: VChain: efficient blockchain based vehicular communication protocol. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, Noida (2020)
20. Cocco, L., Pinna, A., Marchesi, M.: Banking on blockchain: Costs savings thanks to the blockchain technology. Future internet **9**(3), 25 (2017)
21. Li, D., et al.: Information security model of block chain based on intrusion sensing in the IoT environment. Cluster Comput. **22**(1), 451–468 (2019)
22. Santander: Santander launches the first real-time trades in Spain using we.trade, a blockchain platform that helps companies go international. https://bit.ly/2Fw2pj7. Accessed 01 Oct 2020

23. Schwartz, D., Youngs, N., Britto, A.: The ripple protocol consensus algorithm. Ripple Labs Inc White Paper **5**(8), 151 (2014)
24. Yadav, A., Singh, N., Kushwaha, D.S.: A scalable trust based consensus mechanism for secure and tamper free property transaction mechanism using DLT. Int. J. Syst. Assur. Eng. Manag. (2021). https://doi.org/10.1007/s13198-021-01335-0
25. Back, A., et al.: Enabling blockchain innovations with pegged sidechains. http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains 72 (2014)
26. Parmar, V.P., Kumbharana, C.K.: Comparing linear search and binary search algorithms to search an element from a linear list implemented through static array, dynamic array and linked list. Int. J. Comput. Appl. 121(3) (2015)
27. Yadav, A.S., Kushwaha, D.S.: Blockchain-based digitization of land record through trust value-based consensus algorithm. Peer-to-Peer Netw. Appl. **14**, 3540–3558 (2021)

Amrendra Singh Yadav obtained his obtained B.Tech degree in Information Technology from Uttar Pradesh Technical University Lucknow, India in 2012 and M.Tech degree in Computer Science and Engineering from Madan Mohan Malaviyan University of Technology, Gorakhpur (U.P), India in 2014. Now he is currently a PhD student in the Department of CSE, Motilal Nehru National Institute of Technology Allahabad, India. His research interest includes Blockchain Technology and network optimization. He has published more than 15 research articles in various SCI and Scopus indexed journals and conferences.

Nikita Singh is working with ICFAI TECH Hyderabad in the Department of Computer Science & Engineering. She received PhD degree in Computer Science and Engineering at National Institute of Technology Raipur, Raipur, India. Her research interest includes Block Chain Technologies, DLT, ML & Image Processing. She has more than 10 Research Papers in highly cited SCI / Scopus indexed Journals.

Dharmender Singh Kushwaha received the B.E in computer engineering from University of Pune, India in 1990. He received the M.Tech and PhD in computer science and engineering from Motilal Nehru National Institute of Technology Allahabad, Allahabad, India in 2007. He was recipient of Gold medal for his masters. Since 2018 he is working as Professor with Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, India. His research interest includes: Distributed systems, service-oriented architecture, software engineering, data structure, image Processing and Blockchain Technology. Contribution in the current study, he stressed on the need for research that helps national security, outline of the proposed system along with its applicability, and mentor and supervision of the proposed work.