(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 October 2021 (07.10.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/198750 A1

(51) International Patent Classification:
H04L 9/06 (2006.01) *H04L 12/26* (2006.01)
H04L 12/24 (2006.01) *H04L 29/08* (2006.01)

(21) International Application Number:
PCT/IB2020/056892

(22) International Filing Date:
22 July 2020 (22.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202041013941 30 March 2020 (30.03.2020) IN

(72) Inventor; and

(71) Applicant: **SOGALA, Satchidananda Sivachidambarasarma** [IN/IN]; "Srichid", Third Floor, #563, Second cross, Second Main, RBI Layout, Seventh Phase, JP Nagar, Bangalore 560078 (IN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

(54) **Title:** SYSTEM AND METHOD TO MANAGE INFORMATION AND DOCUMENTS ON A NATIVE BLOCKCHAIN NETWORK SYSTEM INCLUDING PERMISSIONED BLOCKCHAIN, STORAGE, SHARING, ORGANISATION, PORTING AND VARIOUS APPLICATIONS

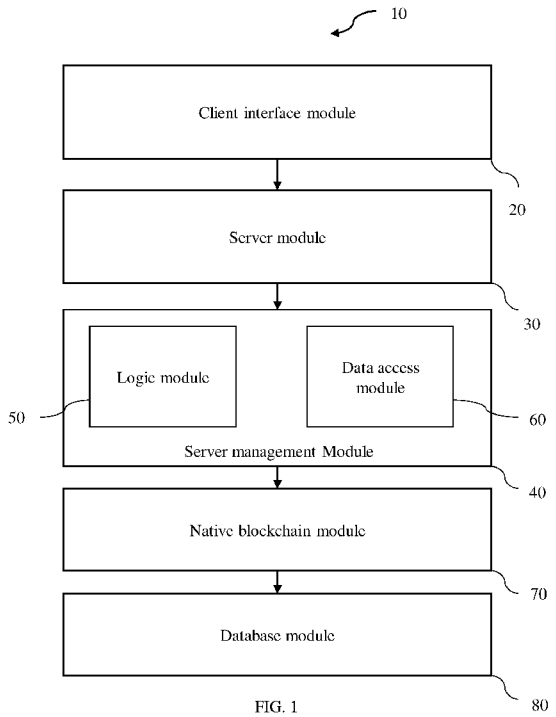

FIG. 1

(57) **Abstract:** Systems using blockchain technology for storage, access, distribution, exchange and execution of electronic information including images, files, media between entities (peer to peer, peer to vendor, peer to system) is more secure during transmission and storage on connected devices and servers than traditional systems. However, there does not exist a decentralized system to protect electronic information to view, edit, store and distribute electronic information securely using webmail plugins, drives or through permissioned blockchain network using smart contracts or through mobile applications. The system includes a client interface module, a server module, a server management module, a logic module, a data access module, a native block chain module and a database module.

WO 2021/198750 A1

[Continued on next page]

Published:

— *with international search report (Art. 21(3))*

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

SYSTEM AND METHOD TO MANAGE INFORMATION AND DOCUMENTS ON A NATIVE BLOCKCHAIN NETWORK SYSTEM INCLUDING PERMISSIONED BLOCKCHAIN, STORAGE, SHARING, ORGANISATION, PORTING AND VARIOUS APPLICATIONS

This International Application claims priority from a complete patent application filed in India having patent application number 202041013941, filed on March 30, 2020 and titled "SYSTEM AND METHOD TO MANAGE INFORMATION AND DOCUMENTS ON A NATIVE BLOCKCHAIN NETWORK SYSTEM INCLUDING PERMISSIONED BLOCKCHAIN, STORAGE, SHARING, ORGANISATION, PORTING AND VARIOUS APPLICATIONS".

FIELD OF INVENTION

Embodiments of a present disclosure relates to a field of electronic sharing, storing and messaging, and more particularly to a system and a method to manage information and documents on a native blockchain network.

BACKGROUND

Blockchain technology (sometimes simply referred to as blockchain) is a relatively new technology that has been used in digital currency implementations. The blockchain is a data structure that stores a list of transactions and can be thought of as a distributed electronic ledger that records transactions between source identifier(s) and destination identifier(s). The transactions are bundled into blocks and every block (except for the first block) refers back to or is linked to a prior block in the chain. Computer nodes maintain the blockchain and cryptographically validate each new block and thus the transactions contained in the corresponding block.

The integrity (e.g., confidence that a previously recorded transaction has not been modified) of the entire blockchain is maintained because each block refers to or includes a cryptographic hash value of the prior block. Accordingly, once a block refers to a prior block, it becomes difficult to modify or tamper with the data (e.g., the transactions) contained therein. This is because even a small modification to the data will affect the hash value of the entire block. Each additional block increases the

difficulty of tampering with the contents of an earlier block. Thus, even though the contents of a blockchain may be available for all to see, they become practically immutable.

Blocks in the blockchain are created using cryptography. Blockchain technology utilizes cryptography as a means of ensuring transactions are done safely, while securing all information and storages of value. Therefore, anyone using blockchain can have complete confidence that once something is recorded on a blockchain, it is done so legitimately and in a manner that preserves security.

An efficient approach would be to provide a decentralized system to protect electronic information to view, edit, store and distribute electronic information securely using webmail plugins, drives or through permissioned blockchain network using smart contracts or through mobile applications.

Hence, there is a need for an improved system to manage information and documents on a native blockchain network and a method to operate the same and therefore address the aforementioned issues.

BRIEF DESCRIPTION

In accordance with one embodiment of the disclosure, a system to manage information and documents on a native blockchain network is disclosed. The system includes one or more processors. The system also includes a client interface module operable by the one or more processors. The client interface module is configured to provide interfacing facility for a first blockchain node and a second blockchain node on the native blockchain network. The client interface module allows sharing of information and documents.

The system also includes a server module operable by the one or more processors. The server module is configured to host the client interface module on the native blockchain network. The system also includes a server management module operable by the one or more processors. The server management module includes a logic module operable by the one or more processors. The logic module is configured to determine a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node.

The logic module is also configured to determine a second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node. The server management module also includes a data access module operable by the one or more processors. The data access module is configured to enable the second blockchain node to access the data corresponding to the information and documents.

The system also includes a native blockchain module operable by the one or more processors. The native blockchain module is configured to provide a plurality of function for the data corresponding to the information and documents. Furthermore, the plurality of function refers to a creation of a block on the native blockchain network, provide transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network.

The system also includes a database module operable by the one or more processors. The database module is configured to store transaction blocks, folders, subfolders, documents, other transactions in Redis database.

In accordance with one embodiment of the disclosure, a method for managing information and documents on a native blockchain network is disclosed. The method includes facilitating interfacing between a first blockchain node and a second blockchain node on the native blockchain network. The method also includes allowing sharing of information and documents. The method also includes hosting the first blockchain node and the second blockchain node on the native blockchain network.

The method also includes determining a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node. The method also includes determining a second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node. The method also includes enabling the second blockchain node to access the data corresponding to the information and documents.

The method also includes providing a plurality of function for the data corresponding to the information and documents. The method also includes storing transaction blocks, folders, subfolders, documents, other transactions in Redis database.

To further clarify the advantages and features of the present disclosure, a more particular description of the disclosure will follow by reference to specific embodiments thereof, which are illustrated in the appended figures. It is to be appreciated that these figures depict only typical embodiments of the disclosure and are therefore not to be considered limiting in scope. The disclosure will be described and explained with additional specificity and detail with the appended figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosure will be described and explained with additional specificity and detail with the accompanying figures in which:

FIG. 1 is a block diagram representation of a system to manage information and documents on a native blockchain in accordance with an embodiment of the present disclosure;

FIG. 2 is an architecture representation of an embodiment representing the system to manage information and documents on a native blockchain of FIG. 1 in accordance of an embodiment of the present disclosure;

FIG. 3 shows the internal working of the native blockchain network corresponding to the overview of the entire process on the native blockchain;

FIG. 4 shows document storage flow diagram corresponding to the native blockchain network in accordance of an embodiment of the present disclosure;

FIG. 5 shows block creation process on the native blockchain in accordance of an embodiment of the present disclosure;

FIG. 6 shows public key infrastructure corresponding to cyber security implementation with the native blockchain network;

FIG. 7 shows another embodiment for public key infrastructure corresponding to cyber security implementation with the native blockchain network;

FIG. 8 shows file sharing flow diagram corresponding to the native blockchain network in accordance of an embodiment of the present disclosure;

FIG. 9 shows smart contract working between users and nodes;

FIG. 10 shows procedure to access the native blockchain network to access through mobile application;

FIG. 11 is a schematic representation of an embodiment representing the management of information and documents on the native blockchain network with the help of private key and the public key;

FIG. 12 is schematic representation of storing of the information and documents securely in a backup storage system in accordance of an embodiment of the present disclosure;

FIG. 13 is schematic representation of downloading information and documents securely from the backup storage system in accordance of an embodiment of the present disclosure;

FIG. 14 is a block diagram of a computer or a server in accordance with an embodiment of the present disclosure;

FIG. 15 is a flowchart representing the steps of a method for managing information and documents on a native blockchain network in accordance with an embodiment of the present disclosure;

FIG. 16 is a flowchart representing the steps of a method for syncing drive to the web application on a backup storage device in accordance with an embodiment of the present disclosure;

FIG. 17 is a flowchart representing the steps of a method for uploading and downloading of files and folders application on a backup storage device in accordance with an embodiment of the present disclosure;

FIG. 18 is a schematic representation of a peer to peer connection over the distributed network in accordance with an embodiment of the present disclosure; and

FIG. 19 is a schematic representation of custom file viewer in accordance with an embodiment of the present disclosure.

Further, those skilled in the art will appreciate that elements in the figures are illustrated for simplicity and may not have necessarily been drawn to scale. Furthermore, in terms of the construction of the device, one or more components of the device may have been represented in the figures by conventional symbols, and the figures may show only those specific details that are pertinent to understanding the embodiments of the present disclosure so as not to obscure the figures with details that will be readily apparent to those skilled in the art having the benefit of the description herein.

DETAILED DESCRIPTION

For the purpose of promoting an understanding of the principles of the disclosure, reference will now be made to the embodiment illustrated in the figures and specific language will be used to describe them. It will nevertheless be understood that no limitation of the scope of the disclosure is thereby intended. Such alterations and further modifications in the illustrated online platform, and such further applications of the principles of the disclosure as would normally occur to those skilled in the art are to be construed as being within the scope of the present disclosure.

The terms "comprises", "comprising", or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a process or method that comprises a list of steps does not include only those steps but may include other steps not expressly listed or inherent to such a process or method. Similarly, one or more devices or subsystems or elements or structures or components preceded by "comprises... a" does not, without more constraints, preclude the existence of other devices, subsystems, elements, structures, components, additional devices, additional subsystems, additional elements, additional structures or additional components. Appearances of the phrase "in an embodiment", "in another embodiment" and similar language throughout this specification may, but not necessarily do, all refer to the same embodiment.

Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by those skilled in the art to which this disclosure belongs. The system, methods, and examples provided herein are only illustrative and not intended to be limiting.

In the following specification and the claims, reference will be made to a number of terms, which shall be defined to have the following meanings. The singular forms "a", "an", and "the" include plural references unless the context clearly dictates otherwise.

Embodiments of the present disclosure relate to a system to manage information and documents on a native blockchain network. The system includes one or more processors. The system also includes a client interface module operable by the one or more processors. The client interface module is configured to provide interfacing facility for a first blockchain node and a second blockchain node on the native blockchain network. The client interface module allows sharing of information and documents.

The system also includes a server module operable by the one or more processors. The server module is configured to host the client interface module on the native blockchain network. The system also includes a server management module operable by the one or more processors. The server management module includes a logic module operable by the one or more processors. The logic module is configured to determine first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node.

The logic module is also configured to determine second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node. The server management module also includes a data access module operable by the one or more processors. The data access module is configured to enable the second blockchain node to access the data corresponding to the information and documents.

The system also includes a native blockchain module operable by the one or more processors. The native blockchain module is configured to provide a plurality of function for the data corresponding to the information and documents. Furthermore, the plurality of function refers to a creation of a block on the native blockchain network, provide transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network.

The system also includes a database module operable by the one or more processors. The database module is configured to store transaction blocks, folders, subfolders, documents, other transactions in Redis database.

FIG. 1 is a block diagram representation of a system (10) to manage information and documents on a native blockchain in accordance with an embodiment of the present disclosure. In one embodiment, the system also manages files, videos and the like. In such embodiment, management work indicates storage, access, distribution and exchange of electronic information and documents. Further, such functions lead to protection against cybercrimes.

The system (10) to manage information and documents on a native blockchain network is disclosed. The system (10) includes one or more processors. The system (10) also includes a client interface module (20) operable by the one or more processors. The client interface module (20) is configured to provide interfacing facility for a first blockchain node and a second blockchain node on the native blockchain network. The client interface module allows sharing of information and documents. It includes an enterprise application which may be accessed using a web browser.

In one embodiment, nodes are the communication entities of the blockchain system. As used herein, the term "node" may perform a logical function in the sense that multiple nodes of different types may run on the same physical server. Nodes are grouped in trust domains and are associated with logical entities that control them in various ways. Nodes may include different types, such as a client or submitting-client node which submits a transaction-invocation to an endorser (e.g., peer), and broadcasts transaction-proposals to an ordering service (e.g., ordering node). Another type of node is a peer node which may receive client submitted transactions, commit the transactions and maintain a state and a copy of the ledger of blockchain transactions.

An ordering-service-node or orderer is a node running the communication service for all nodes, and which implements a delivery guarantee, such as a broadcast to each of the peer nodes in the system when committing transactions and modifying a world state of the blockchain, which is another name for the initial blockchain transaction which normally includes control and setup information.

The system (10) also includes a server module (30) operable by the one or more processors. The server module (30) is configured to host the client interface module on the native blockchain network. In one embodiment, the server host may be Microsoft Azure, Aws, Apache web server and the like.

The system (10) also includes a server management module (40) operable by the one or more processors. The server management module (40) includes a logic module (50) operable by the one or more processors. The logic module (50) is configured to determine a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node. In one embodiment, the first set of approaches may be a real time method as formulated data transformation and controlling. In such embodiment, the data is provided in the native blockchain platform via the first blockchain node.

The logic module (50) is also configured to determine a second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node. In one embodiment, the routing mechanism is determined real time by the logic module (50) based on the access point of the second blockchain node.

The server management module (40) also includes a data access module (60) operable by the one or more processors. The data access module (60) is configured to enable the second blockchain node to access the data corresponding to the information and documents.

The system (10) also includes a native blockchain module (70) operable by the one or more processors. The native blockchain module (70) is configured to provide a plurality of functions for the data corresponding to the information and documents. Furthermore, the plurality of functions refers to a creation of a block on the native blockchain network, provide transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network.

The system (10) also includes a database module (80) operable by the one or more processors. The database module (80) is configured to store transactions blocks, folders, subfolders, documents, other transactions in Redis database.

The system (10) further comprises a registration module operable by the one or more processors. The registration module is configured to provide registration facilities to the first blockchain node and the second blockchain node on the native blockchain network. During the registration a private key and a public key is generated. Further, the private key and the public key are stored in a key vault.

In one exemplary situation, a customer fills the registration form and attaches his/her ID proof, system extracts the inputs of the customer from data given by the customer, data filled through the form is analysed & attached ID Proof is authenticated from the customer, after verification an alias Id is generated for the Id proof, and the customer is identified only using alias Id and the original document is not stored in the database.

The system (10) further comprises a backup storage module operable by the one or more processors. The backup storage module configured to store backup for the database module (80). The backup may be stored on a solid-state drive. SSD is made as compact disk by changing the permission into write once read many (WORM). Write once read many (WORM) describes a data storage device in which information, once written, cannot be modified. This write protection affords the assurance that the data cannot be tampered with once it is written to the device.

FIG. 2 is an architecture representation (10) of an embodiment representing the system to manage information and documents on a native blockchain of FIG. 1 in accordance of an embodiment of the present disclosure. First layer (20) is the user interface with customised application which is accessed using web browser. Customised application version may be for individual, organisation, company and the like. The first layer (20) has the ability of sharing document in multilevel workflow to the users.

Second layer (30) of the architecture (10) is the web server. The layer provides location where application is hosted. The server may be Microsoft Azure, Aws, Apache web server and the like.

Third layer (40) is the server management component in which both data logic layer and business logic layer is combined here. Both such layers are the most crucial part of this application. Business logic determines how data is transformed or calculated and how it is routed to user or software (workflow). Data Access Layer allows the client (or user) modules to be created with a higher level of abstraction. Furthermore,

the data access object and the business methods interact with blockchain layer (70) in order to create block, make transaction, fetch data from data base, execute methods to mine blocks and the like.

Creation of the block happens in this blockchain layer (70). In such embodiment, the blockchain layer has three more layers in it the first layer is the Consensus Layer. The consensus layer is the protocol that describes the format of rules between user, for example smart contract. The protocol also allows new blocks to be added to the chain in DB.

Second layer includes a mining layer. In such layer, blocks are mined in order to validate the transaction for example proof of work. Further, third layer is a Propagation layer. The propagation layer is a protocol that determines how the blocks are transmitted between nodes in the network.

The architecture (10) further includes a database layer (80). All the transactions Blocks, folders, subfolders, documents, other transactions are stored in Redis database with a key. Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyper loglogs, geospatial indexes with radius queries and streams.

Lastly, an additional layer of a backup layer may be present. In such embodiment, solid state drive (SSD) is used to store all the data which are stored in Redis database (DB). The SSD has the exact replica of the data stored Redis DB.

FIG. 3 shows the internal working of the native blockchain network (90) corresponding to the overview of the entire process on the native blockchain. In a particular embodiment, a document (100) is shown to be encrypted (110) and stored in a provided blockchain network. During the time of downloading, the stored encrypted document is decrypted to obtain back the document (130) originally stored. During storing, blocks are being created during storing and accessing (120).

FIG. 4 shows document storage flow diagram (140) corresponding to the native blockchain network in accordance of an embodiment of the present disclosure. In Redis DB (180), a copy of the block with a same key is stored in SSD as a backup,

whenever data is retrieved from the DB the data is verified in the SSD to prevent from dangling block in Redis DB.

Whenever a document is uploaded by the user a new (150) block is created, figure shows the retrieval of the genesis block from Redis DB (180) in order to get hash of genesis block (hash of the previous block) to create block 1 (170), the genesis block is fetched from Redis DB (100) for the retrieval process of the block from Redis DB.

The data from the "Current Key" is read to find the latest block uploaded. The data is fetched from the Redis DB (180) using GET operation, later decompressed (i.e unzip process) , the JSON string is converted back to STRUCT type definition (Unmarshalling) of the (original state of the block) block, then the hash of the genesis block becomes the previous block of the current block . Also new data1 is encrypted assigned to data field in the block, new hash value is generated, this same process continues for upcoming uploads.

For displaying the document to the user, all the blocks related to the user is retrieved from the Redis DB (180) and verified by SSD Layer, the block are just parsed, data in the block is collected and corresponding Folder & Sub Folder(i.e folder Id & subfolder Id) in which document was uploaded is also fetched from Redis DB (180). And then the documents are displayed to the user (150).

Validation of the block is the most important part of the blockchain data, one of the advantages of using blockchain is data security. Data security means that tampering with the old data and altering the method of securing new data is prevented by both the cryptographic method and the non-centralized storage of the data itself. However, blockchain is just a data structure in which data can be easily changed if any other user has access to it. First, each block has to check whether the data in block is changed or not. Second, Previous blocks hash has to check if the block is changed or not & recalculated. After all the Blocks are recalculated, the verification is passed.

Proof of Work is a scheme used in generating a new block for a blockchain. Proof of work requires a significant amount of work, also known as computing time, to generate a piece of information for a new block. The generated information must be simple and verifiable. Therefore, it can be easily verified by any nodes in the network. The

generated information is the proof that this block is valid and some work has been done to generate it.

The amount of work required for generating a new block is determined from the average computing time on generating a new block in the entire Blockchain network. The algorithm picked to implement Proof of Work scheme is a hash algorithm. The most widely used hash algorithm in Proof of Work is SHA-512. To generate a hash that has a specific format, like a number of leading zeros, is very computing intensive and time-consuming. verifying the source data that matches with the generated hash is trivial. Because a specific piece of data can only get a specific hash, the source data must be changed to generate a different hash. This is solved by introducing "nonce" in the data structure. The nonce is an integer. By increasing the nonce, the hash algorithm can generate a different hash. This process will be ended until the generated hash meets the requirement, it is called as difficulty.

Furthermore, FIG 4 (140) illustrates the mining process, the creation of block is same as mentioned above but only extra element that needed to be included is the difficulty, the difficulty is an integer that indicates the number of leading zeros required for a generated hash The Mine method tries to find a hash that matches with difficulty. If a generated hash doesn't meet the difficulty, then it increases nonce to generate a new one. The process will be ended when a qualified hash is found. The figure shows generation hash between genesis block and block1, this method is used for creation of other blocks.

FIG. 5 shows block creation process on the native blockchain in accordance of an embodiment of the present disclosure. In such embodiment, the creation of the block, all this metadata is encrypted and the STRUCT type block consists of Index: to identify the block, Timestamp: time at which the block is created, Previous hash: hash of the previous block, Hash: hash of the current block & Nonce fields.

In stated above as shown the first block will be a genesis block it will not have previous block value or data, a hash is generated with Index, Timestamp, Previous hash, Hash, Nonce using SHA-512, this is the hash value 0X6465462b3c3ec8841b54169588c97f7 of the Genesis Block (210) as shown in Figure 5 this hash value will be the previous

hash value in Block 1 (200), data will be empty since there is no data, or any data can be added statically, the blocks original form will be STRUCT type.

In Block 1 (200), the real metadata of the document & original data(document) uploaded by the user is encrypted as shown in figure 5 (HJTX3gAAAAMdib4lZvIi4I8jUnEWnkEwvGk0o) using AES algorithm this encrypted value is set as the value of the 'data' in block 1 (200), and as mentioned above hash of the Index, Timestamp, Previous hash, Hash, Nonce is generated using SHA-512 and set as the value of the "hash" in the block. And, such process keeps on repeating on each upload or in case of any other transaction.

FIG. 6 shows public key infrastructure corresponding to cyber security implementation with the native blockchain network. FIG. 7 shows another embodiment for public key infrastructure corresponding to cyber security implementation with the native blockchain network.

The messaging is very useful for this enterprise version (organisation), since messaging system is available in the application itself. this system uses PKI infrastructure for messaging it will very secured enough the protect the messages. Both the users user1 and user 2 are given private key and public key during the registration process; users can use same key for sending message. But for most practical applications, several parties or communication transactions may be involved, and it becomes necessary for encryption keys to be transmitted over networks whose security may be in doubt. Group messaging will also be featured.

In another embodiment, the public key and the private key are issued by the certification authority to the customer and the company. In such scenario, server sends mail to the customers by wrapping the public key within mail and the private key of the customer is used by the customer itself to decrypt the message.

FIG. 8 shows file sharing flow diagram corresponding to the native blockchain network in accordance of an embodiment of the present disclosure. Exchanging or sharing data in blockchain, figure 8 illustrates sharing document in blockchain network to the other user and other users in the other node, in order to share users share document to other user, two users have to be friend, account id of the user is

exchanged both the user become friend, during this process a smart contract is established between the users or nodes and it is uploaded on the blockchain network.

In provided situation, user selects a document from his portal and selects the user to whom he wants to send the document, the selected documents key is found and the block is fetched from the Redis DB and also cross verified in the SSD. data in the block is parsed, the document that user wants to share is identified, instead of giving access to the existing block a new block is created to the end user.

New key in Redis is generated and block is stored (same process for creating block as explained above), and separate folder and subfolders are created for the user2. So that all the received files are categorised for viewing the document, user can even send message with the document that is shared.

The workflow layer is present on the first layer. i.e in the interface layer, this part of the application is purely dedicated for the working organisation. In such process a user interacts with enterprise application, the user may be a part of the same application or a complete external user, for example a user fills the form and attaches document, documents and user data need to be verified by enterprise application officials. The uploaded document is received by the one level of users of the workflow through inbox, this uploaded document is Inwarded, allotted to next level user for approval of the document. If the received document is rejected then it is sent back to the user for rectification, similarly the user's data and the document has to go through four level of validation, each level user can also upload extra document with the received one all this document will move along with user's data to next level.

Deletion of Blocks may be done in two approaches. Deletion blocks is done to free DB memory and reduce computational of the system; deletion of block is also deleting document from user perspective. In first approach, once a document is deleted by the user, user has to agree upon the terms and condition while deleting the data in blockchain (based on consensus). Only the flag is set in the DB for the corresponding block to show as if the block is deleted, but in the DB the block is still available in order to maintain the integrity of the blockchain. Still the data is persevered in blockchain for next 30 days for emergency recovery. After 30 days these blocks become orphan or considered as dangling block. This block is deleted form the

blockchain and to preserve the integrity of the blockchain, the hash is recomputed from the beginning to the last block.

New nonce will get generated for every block as a proof of work. the transaction that used the deleted blocks hash for sharing document is still maintained for log purpose. Any other block which is deleted without any consensus will not be deleted and a proper feedback from the user is taken in order to secure user data Protect against suspicious account login. Similarly, any block which doesn't have nonce without proof of work will be considered as invalid and deleted. Problem with first approach is that high computational power is required to recompute all the blocks from the begin to the end, higher the computational complexity higher will be time complexity.

In another approach only the data in the block is deleted but the hash value of the block is preserved. during regular validation of the chain, hash of the block is considered as valid but the block is marked as a "dummy block". This method will not consume time or computational overhead and the chain will grow as usual. The advantage of this method is, it will free up DB memory without using high computational power of the systems. This dummy block will not cause any harm to chain or the user.

FIG. 9 shows smart contract (360) working between users and nodes. In given embodiment, the registered users before sharing of data accepts the contract proposal associated with the native blockchain platform. Further, the conditions (390) may be accepted after checking the laid down rules. After such acceptance only, the key exchange takes place.

Furthermore, a smart contract is a computer code running on top of a blockchain containing a set of rules under which the parties to that smart contract agree to interact with each other. If and when the pre-defined rules are met, the agreement is automatically enforced. The smart contract code facilitates, verifies, and enforces the negotiation or performance of an agreement or transaction. It is the simplest form of decentralized automation. In one embodiment, the smart contract for sharing the file, the first condition to share document between users is that the both of the users (380, 370) must have an account, and both have to be friend.

In one embodiment, user 1 selects a document from his portal and selects the user to whom he wants to send the document, the selected documents key is found and the

block is fetched from the Redis DB and also cross verified in the SSD. data in the block is parsed, the document that user wants to share is identified, instead of giving access to the existing block a new block is created to the end user (user2). New key in Redis is generated and block is stored (same process for creating block as explained above), and separate folder and subfolders are created for the user2. So that all the received files are categorised for viewing the document, user can even send message with the document that is shared.

FIG. 10 shows procedure to access the native blockchain network to access through mobile application (400). Mobile Application System designed to run enterprise blockchain file system on mobile device such as a phone/tablet or watch. A mobile application is made and will be available on play store, apple store and the like, for the user to access it based on supported platform. Figure 10 illustrates the working of individual application (410) in mobile, a separate mobile interface is built with all enterprise application features which was available on web application (420), mobile application also has user dashboard, users can upload files from their mobile phone request will be processed as it was in web application (420).

Figure 10 shows mobile application (410) calls a web api, web api process the request and creates folder, subfolder or uploads documents. If the request type is to "upload document" or "share document" then request is forwarded to the blockchain layer (430) for the creation of the block, else folder or subfolder is created. The remaining block creation, storing (440, 450), validation, proof of work process will be same as explained. Also, users in the organisation do get to login through mobile application (410) and perform their workflow operations (approve/reject, allot, inward etc) and some other process as mention above.

FIG. 11 is a schematic representation of an embodiment representing the management of information and documents on the native blockchain network (460) with the help of private key and the public key. In an exemplary scenario, a registered user may at first upload the message on the system (480). For transmitting, the message is encrypted via an encryption technique and public key (470). The system (480) enables storing of the message on the platform for further use.

Any recipient may download the message for viewing from the system (500). The downloaded message may be decrypted by application of decryption techniques and the private key (490). The message may be read or passed to another user according to need.

FIG. 12 is schematic representation (510) of storing of the information and documents securely in a backup storage system in accordance of an embodiment of the present disclosure. The system (530) enables common platform storage for any file storage. The file storage may be a document, video and the like. First, after uploading, the file is encrypted by an encryption technique and the public key. The encryption leads to the formation of encrypted file (540). The keys are stored in the Azure key vault.

Further, the information relating to encrypted file, such as metadata is encrypted further with blockchain technique or algorithm. Such further encryption leads to the formation of text extension file (550). Text extension file (550) is also stored on the system (530) along with the encrypted file (540). Both the file is downloaded whenever needed.

Furthermore, the file is copied to the external HDD or SDD and the file is encrypted using the destination file path, source file path, AES encryption key, AES encryption IV key and this file is stored in the destination path. In such embodiment, the file will have encrypted extension stating that the file is encrypted. In another embodiment, file signature is calculated using the encrypted file path, signature key to sign the file, this signature uses Hash Based Message Authentication Code (HMAC) HMACSHA256 for signing the file.

A blockchain is simply a chain of blocks that contains information. Each block has a cryptographic hash of the previous block, a timestamp, and transaction data. In a specific situation, OBJECT type block consists of Index: to identify the block, Timestamp: time at which the block is created, Previous hash: hash of the previous block, Hash: hash of the current block & Nonce fields, Data: file related Information. It is pertinent to note that using a blockchain technique or algorithm, the data tampering is reduced to considerable level.

FIG. 13 is schematic representation (560) of downloading information and documents securely from the backup storage system in accordance of an embodiment of the

present disclosure. The stored text extension file (550) as well as the stored encrypted file (540) is decrypted for downloading of the file. First, via a blockchain decryption algorithm, the stored text extension file (550) is decrypted to obtain a manifest file of the encrypted file (540). Lastly, the decrypted manifest file along with private key is being used to decrypt the encrypted file corresponding to the main stored file. The decrypted file is downloaded for viewing via a system (530).

The encrypted data in .txt is read using file read in command and this data represents the block, block is sent to the blockchain algorithm. Manifest data, AES encryption key, AES IV key, Destination path, is parsed from the block. the encrypted file is copied to the user system and then the file is decrypted using the manifest data, AES encryption key, AES IV key, private key. Then the original file is retrieved back in the user's system.

FIG. 14 is a block diagram of a computer or a server (570) in accordance with an embodiment of the present disclosure. The server (570) includes processor(s) (600), and memory (580) coupled to the processor(s) (600).

The processor(s) (600), as used herein, means any type of computational circuit, such as, but not limited to, a microprocessor, a microcontroller, a complex instruction set computing microprocessor, a reduced instruction set computing microprocessor, a very long instruction word microprocessor, an explicitly parallel instruction computing microprocessor, a digital signal processor, or any other type of processing circuit, or a combination thereof.

The memory (580) includes a plurality of modules stored in the form of executable program which instructs the processor (600) via a bus (590) to perform the method steps illustrated in Fig 1. The memory (580) has following modules: a client interface module (20), a server module (30), a server management module (40), a logic module (50), a data access module (60), a native block chain module (70) and a database module (80).

The client interface module (20) is configured to provide interfacing facility for a first blockchain node and a second blockchain node on the native blockchain network. The server module (30) is configured to host the client interface module on the native

blockchain network. The server management module (40) comprises the logic module (50) and the data access module (60).

The logic module (50) is configured to determine a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node. The logic module (50) is also configured to determine a second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node.

The data access module (60) is configured to enable the second blockchain node to access the data corresponding to the information and documents. The native blockchain module (70) is configured to provide a plurality of function for the data corresponding to the information and documents. The database module (80) is configured to store transactions blocks, folders, subfolders, documents, other transactions in Redis database.

Computer memory elements may include any suitable memory device(s) for storing data and executable program, such as read only memory, random access memory, erasable programmable read only memory, electrically erasable programmable read only memory, hard drive, removable media drive for handling memory cards and the like. Embodiments of the present subject matter may be implemented in conjunction with program modules, including functions, procedures, data structures, and application programs, for performing tasks, or defining abstract data types or low-level hardware contexts. Executable program stored on any of the above-mentioned storage media may be executable by the processor(s) (600).

FIG. 15 is a flowchart representing the steps of a method (610) for managing information and documents on a native blockchain network in accordance with an embodiment of the present disclosure. The method (610) includes facilitating interfacing between a first blockchain node and a second blockchain node on the native blockchain network in step 620. In one embodiment, facilitating the interfacing between the first blockchain node and the second blockchain node on the native blockchain network includes facilitating the interfacing between the first blockchain node and the second blockchain node on the native blockchain network by a client interface module.

The method (610) also includes allowing share of information and documents in step 630. In one embodiment, allowing share of information and documents incudes allowing share of information and documents by the client of interface module.

In another embodiment, allowing share of information and documents incudes allowing share of information and documents comprising information and documents from the field of finance, banking, e-commerce transaction and revenue management.

The method (610) also includes hosting the first blockchain node and the second blockchain node on the native blockchain network in step 640. In one embodiment, hosting the first blockchain node and the second blockchain node on the native blockchain network includes hosting the first blockchain node and the second blockchain node on the native blockchain network by the server module.

The method (610) also includes determining a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node in step 650. In one embodiment, determining the first set of approaches for transforming the data corresponding to the information and documents as provided by the first blockchain node includes determining the first set of approaches for transforming the data corresponding to the information and documents as provided by the first blockchain node via a logic module.

The method (610) also includes determining a second set of approaches for routing data corresponding to the provided information and documents to the second blockchain node in step 660. In one embodiment, determining the second set of approaches for routing the data corresponding to the provided information and documents to the second blockchain node includes determining the second set of approaches for routing the data corresponding to the provided information and documents to the second blockchain node by the logic module.

The method (610) also includes enabling the second blockchain node to access the data corresponding to the information and documents in step 670. In one embodiment, enabling the second blockchain node to access the data corresponding to the information and documents includes enabling the second blockchain node to access the data corresponding to the information and documents by a data access module.

The method (610) also includes providing a plurality of function for the data corresponding to the information and documents in step 680. In one embodiment, providing the plurality of function for the data corresponding to the information and documents includes providing the plurality of function for the data corresponding to the information and documents by the native blockchain module.

In another embodiment, providing the plurality of function for the data corresponding to the information and documents includes providing the plurality of function for the data corresponding to the information and documents includes performing the plurality of function via a set rules.

In yet another embodiment, providing the plurality of function for the data corresponding to the information and documents includes the plurality of function via the set rules comprising a consensus layer, a mining layer and a propagation layer.

In yet another embodiment, providing the plurality of function for the data corresponding to the information and documents includes providing the plurality of function comprising a creation of a block on the native blockchain network, transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network.

The method (610) also includes storing transactions blocks, folders, subfolders, documents, other transactions in Redis database in step 690. In one embodiment, storing the transactions blocks, the folders, the subfolders, the documents, other the transactions in the Redis database includes storing the transactions blocks, the folders, the subfolders, the documents, other the transactions in the Redis database by a database storage module.

The method (610) also includes enforcing contract terms between the first blockchain node and the second blockchain node on the native blockchain network. In one embodiment, enforcing the contract terms between the first blockchain node and the second blockchain node on the native blockchain network includes enforcing the contract terms between the first blockchain node and the second blockchain node on the native blockchain network by a smart contract module.

The method (610) also includes providing registration facilities to the first blockchain node and the second blockchain node on the native blockchain network. In one embodiment, providing the registration facilities to the first blockchain node and the second blockchain node on the native blockchain network includes providing the registration facilities to the first blockchain node and the second blockchain node on the native blockchain network by a registration module.

The method (610) also includes storing backup for the database module on a solid-state drive. In one embodiment, storing the backup for the database module on the solid-state drive includes storing the backup for the database module on the solid-state drive by a backup storage module.

FIG. 16 is a flowchart representing the steps of a method (700) for syncing drive to the web application on a backup storage device in accordance with an embodiment of the present disclosure. The method includes downloading extension file from the web application before installation in step 710.

The method also includes logging with web application credentials in step 720. The method also includes creating a default folder with a pre-defined name in step 730. The method also includes updating replicated files and folders into a shell folder (local drive) in step 740. In one embodiment, the stored files in the web application are in blockchain in step 750.

FIG. 17 is a flowchart representing the steps of a method (780) for uploading and downloading of files and folders application on a backup storage device in accordance with an embodiment of the present disclosure. The method includes connecting the portable device in step 790. The method also includes selecting a path for downloading the file in step 800. The method also includes downloading the file in original format in step 810. The method also includes logging with web application credentials in step 820.

The method also includes selecting external device or local drive for dragging and dropping file in step 830. The method also includes storing the encrypted data after generating two files encrypted file and text file in 840. Here, the text file is generated because the database is not been used. The uploaded file may be downloaded by selecting the uploaded file path. The file may be downloaded as required.

FIG. 18 is a schematic representation of a peer to peer connection over the distributed network (850) in accordance with an embodiment of the present disclosure. Peer-to-peer (P2P) computing or networking is a distributed application architecture. Moreover, each of a plurality of peers (890) acts as a plurality of servers and a plurality of clients respectively in a network (880). The Blockchain is applied for this network (880). Each node maintains its own database. All the databases are replica of the other database in a network (880). The Master node is maintained to store genesis block. Before adding data (860) (Block), the data (860) is validated with the previous hash of the remaining nodes in the network. The peer-to-peer computing also use key (870) after validation.

FIG. 19 is a schematic representation of custom file viewer (900) in accordance with an embodiment of the present disclosure. The custom viewer is created by Converting any document (910) to HTML (920) or any customized extension like (.sri) and perform the operations like Markup, E-sign, Redline or any operation which user needs. In one embodiment, the document (910) includes image, PDF, word, excel, PPT and the like. Furthermore, the HTML (920) is converted back to its original Type (930). The original type (930) includes image, PDF, word, excel, PPT and the like.

Present disclosure of a system to manage information and documents on a native blockchain network may be used in number of fields. Such provided native blockchain network is a permissioned blockchain on a decentralized platform. The system uses block chain based DMS on the Microsoft Platform. Further, the Block chain based DMS with workflow on the Open Source platform using Angular 6 for the front-end, Redis for the database, GO language for programming and Linux as the operating system. The proof of work is executed on the Microsoft platform.

Use of block chain, provides Customer Experience through faster and personalized responses by accessing the right information at the right time. Blockchain based file system with Hyperledger, ensures the high levels of privacy and security of the customers data.

Applications may be on Ethereum platform for lending and trading finance banking. Organic community network uses blockchain for the secure ecommerce purposes with PKI Security. Moreover, Banks Consortium Lending may use such described native

blockchain network. Few more application identified areas include, Property Identification and Management System and Khatha Transfer and Revenue Management Block chain-based Academic certification (Certificate / Marks Cards) system.

Key benefits of the above described system are in the domain of file archiving setup, file Search, document view, ease of access, including mobility, document download, file sharing and the like.

While specific language has been used to describe the disclosure, any limitations arising on account of the same are not intended. As would be apparent to a person skilled in the art, various working modifications may be made to the method in order to implement the inventive concept as taught herein.

The figures and the foregoing description give examples of embodiments. Those skilled in the art will appreciate that one or more of the described elements may well be combined into a single functional element. Alternatively, certain elements may be split into multiple functional elements. Elements from one embodiment may be added to another embodiment. For example, order of processes described herein may be changed and are not limited to the manner described herein. Moreover, the actions of any flow diagram need not be implemented in the order shown; nor do all of the acts need to be necessarily performed. Also, those acts that are not dependant on other acts may be performed in parallel with the other acts. The scope of embodiments is by no means limited by these specific examples.

WE CLAIM:

1. A system (10) to manage information and documents on a native blockchain network, comprising:

one or more processors;

a client interface module (20) operable by the one or more processors, wherein the client interface module (20) is configured to provide interfacing facility for a first blockchain node and a second blockchain node on the native blockchain network, wherein the client interface module (20) allows sharing of information and documents;

a server module (30) operable by the one or more processors, wherein the server module (30) is configured to host the client interface module (20) on the native blockchain network;

a server management module (40) operable by the one or more processors, wherein the server management module (40) comprising:

a logic module (50) operable by the one or more processors, wherein the logic module (50) is configured to

determine a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node; and

determine a second set of approaches for routing the data corresponding to the provided information and documents to the second blockchain node;

a data access module (60) operable by the one or more processors, wherein the data access module (60) is configured to enable the second blockchain node to access the data corresponding to the information and documents;

a native blockchain module (70) operable by the one or more processors, wherein the native blockchain module (70) is configured to provide a plurality

of function for the data corresponding to the information and documents, wherein the plurality of function refers to a creation of a block on the native blockchain network, provide transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network;

a database module (80) operable by the one or more processors, wherein the database module (80) is configured to store transactions blocks, folders, subfolders, documents, other transactions in Redis database.

2. The system (10) as claimed in claim **1**, comprises a smart contract module operable by the one or more processors, wherein the smart contract module is configured to enforce contract terms between the first blockchain node and the second blockchain node on the native blockchain network.

3. The system (10) as claimed in claim **1**, comprises a registration module operable by the one or more processors, wherein the registration module is configured to provide registration facilities to the first blockchain node and the second blockchain node on the native blockchain network, wherein during the registration a private key and a public key is generated.

4. The system (10) as claimed in claim **1**, wherein the native blockchain module comprises a set of layers for performing a plurality of function for the data corresponding to the information and documents, wherein the set of layers comprises a consensus layer, a mining layer and a propagation layer.

5. The system (10) as claimed in claim **1**, comprises a backup storage module operable by the one or more processors, wherein the backup storage module configured to store backup for the database module (80), wherein the backup may be stored on a solid-state drive.

6. The system (10) as claimed in claim **1**, wherein managing of information and documents relates to a plurality of fields comprising finance, banking, e-commerce transaction and revenue management.

7. The system as claimed in claim **1**, wherein the native blockchain module comprises a peer-to-peer (P2P) computing or networking in a distributed network, wherein each of a plurality of peers act as a plurality of servers and a plurality of clients respectively in a network, wherein the network in based on a blockchain.

8. The system as claimed in claim **1**, wherein the native block chain module comprises a filesystem for any extension file, wherein one extension file will perform the operations like Markup, E-sign and Redline.

9. A method (610) for managing information and documents on a native blockchain network, comprising:

facilitating, by a client interface module, interfacing between a first blockchain node and a second blockchain node on the native blockchain network (620);

allowing, by the client interface module, share of information and documents (630);

hosting, by a server module, the first blockchain node and the second blockchain node on the native blockchain network (640);

determining, by a logic module, a first set of approaches for transforming data corresponding to the information and documents as provided by the first blockchain node (650);

determining, by the logic module, a second set of approaches for routing the data corresponding to the provided information and documents to the second blockchain node (660);

enabling, by a data access module, the second blockchain node to access the data corresponding to the information and documents (670);

providing, by a native blockchain module, a plurality of function for the data corresponding to the information and documents (680); and

storing, by a database storage module, transactions blocks, folders, subfolders, documents, other transactions in Redis database (690).

10. The method (610) as claimed in claim **9**, comprising enforcing, by a smart contract module, contract terms between the first blockchain node and the second blockchain node on the native blockchain network.

11. The method (610) as claimed in claim **9**, comprising providing, by a registration module, registration facilities to the first blockchain node and the second blockchain node on the native blockchain network.

12. The method (610) as claimed in claim **9**, comprising storing, by a backup storage module, backup for the database module on a solid-state drive.

13. The method (610) as claimed in claim **9**, wherein performing, by the native blockchain module, the plurality of function for the data corresponding to the information and documents via a set of layers.

14. The method (610) as claimed in claim **9**, wherein performing, by the native blockchain module, the plurality of function via the set rules comprising a consensus layer, a mining layer and a propagation layer.

15. The method (610) as claimed in claim **9**, wherein allowing, by the client interface module, sharing of information and documents comprising information and documents from the field of finance, banking, e-commerce transaction and revenue management.

16. The method (610) as claimed in claim **9**, wherein providing, by the native blockchain module, the plurality of function comprising a creation of a block on the native blockchain network, transaction on the native blockchain network, fetch data based on the native blockchain network and execution methods to mine block based on the native blockchain network.

17. The method (610) as claimed in claim **12**, wherein storing in a backup module comprises downloading initially a .exe file through a user credentials so as to undergo a PKI architecture, wherein the PKI architecture files and folders are dragged and dropped to get uploaded in a shell folder which is created in a side bar of file explorer.

18. The method (610) as claimed in claim **12**, wherein storing in a backup module comprises selecting any portable drive such as internal or external for storing files and folders, wherein two files such as encrypted file and text file is stored.

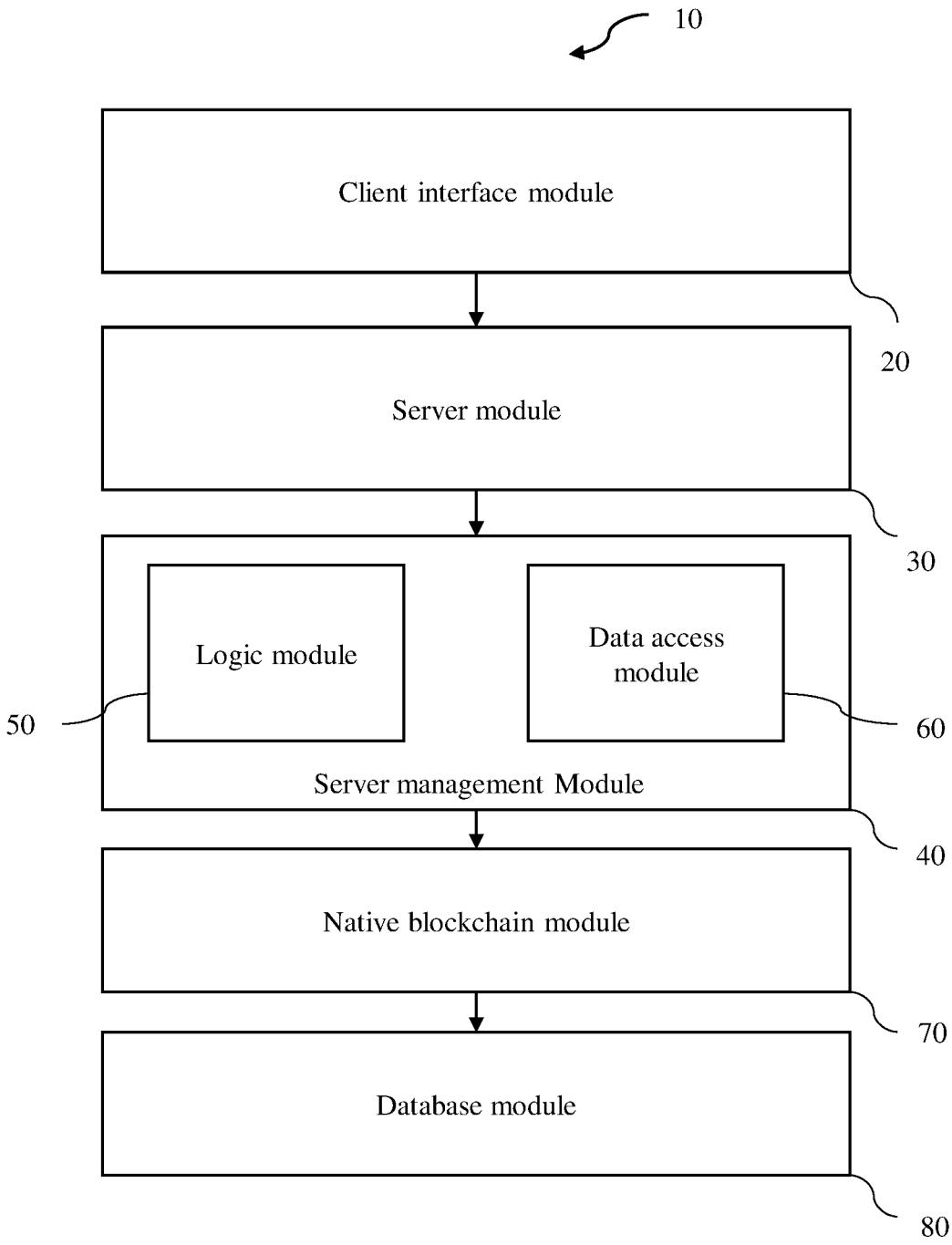
10
Client interface module
20
Server module
30
Logic module
Data access module
50
60
Server management Module
40
Native blockchain module
70
Database module
80

FIG. 1

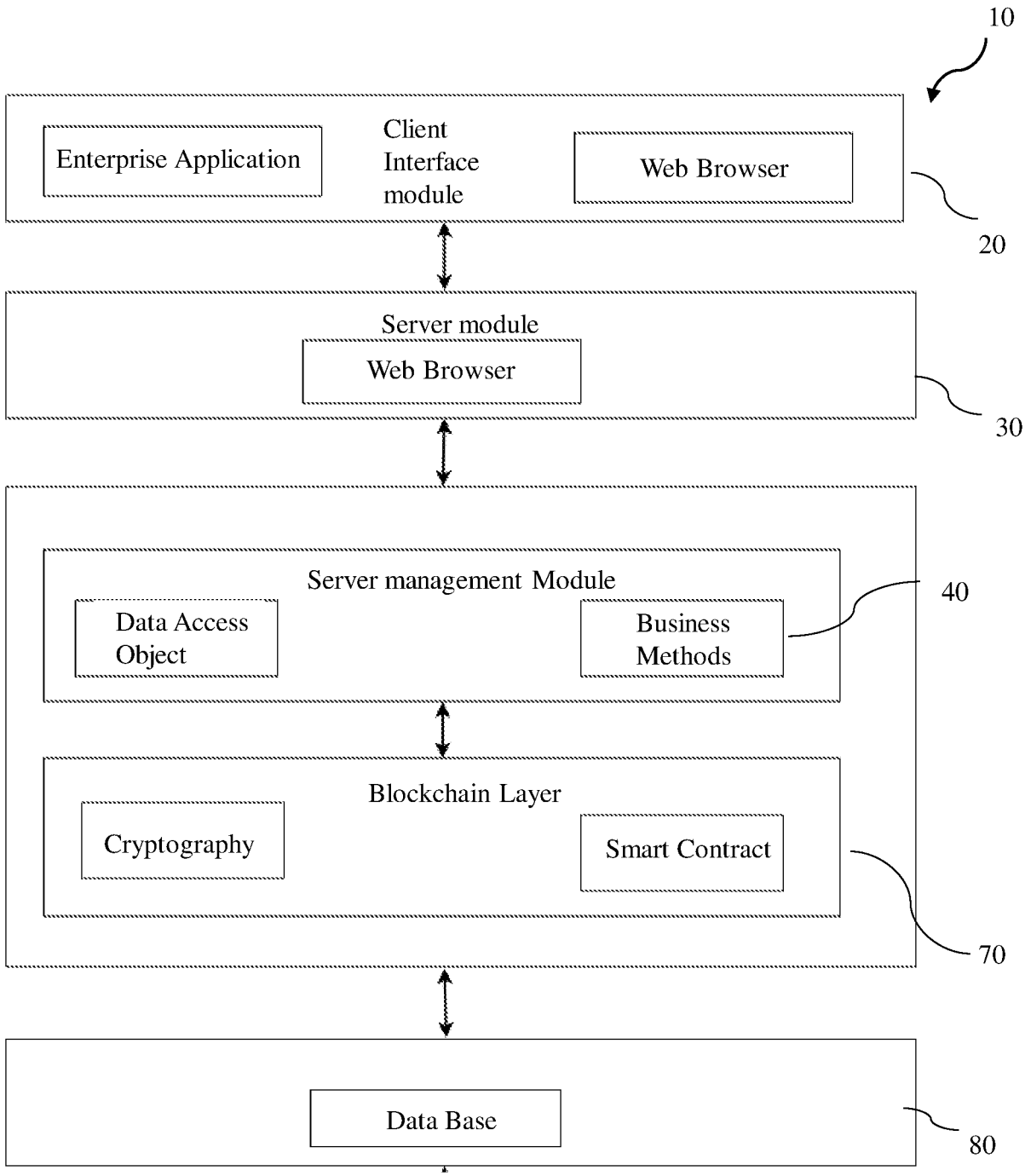
10
Enterprise Application
Client Interface module
Web Browser
20
Server module
Web Browser
30
Server management Module
Data Access Object
Business Methods
40
Blockchain Layer
Cryptography
Smart Contract
70
Data Base
80

FIG. 2

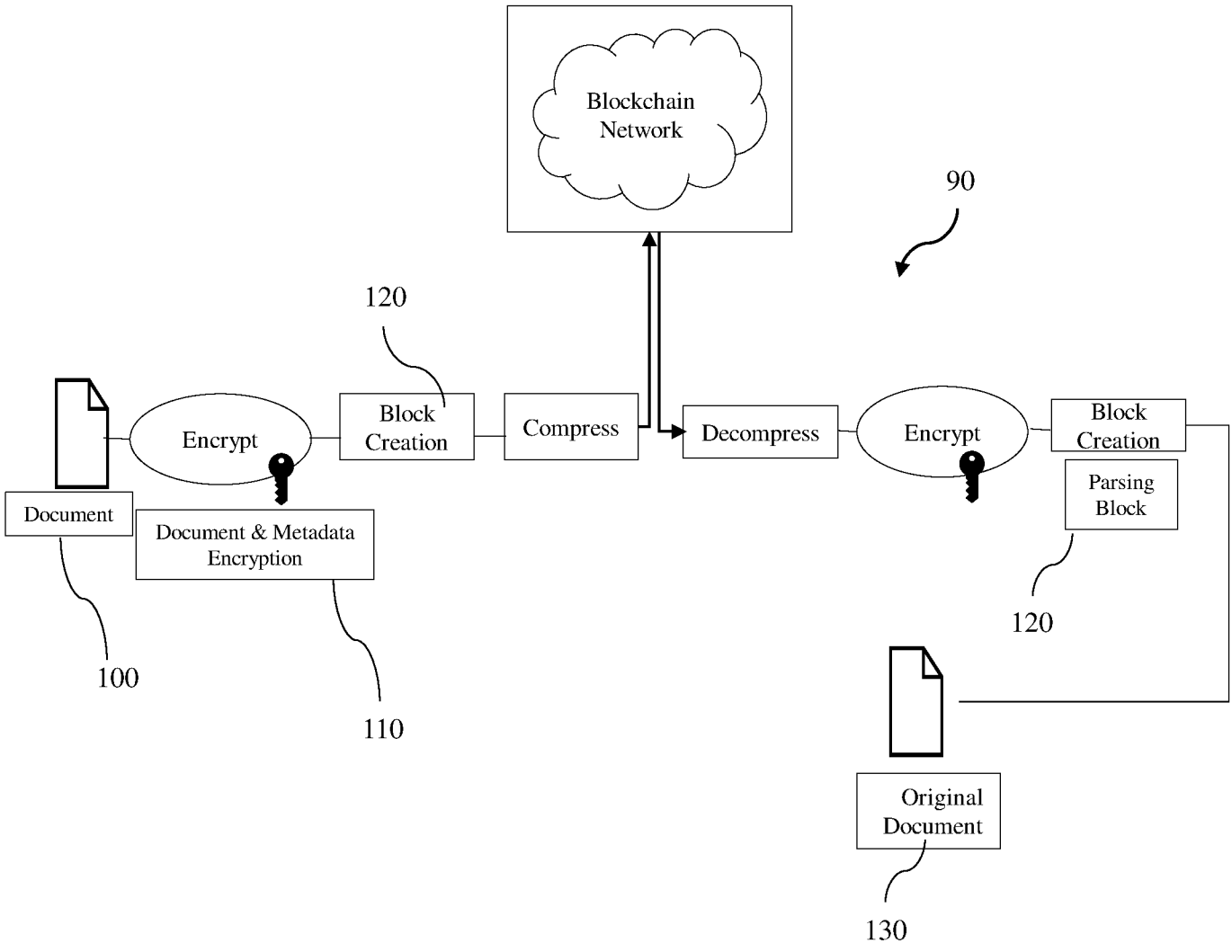
Blockchain Network
90
120
Encrypt
Block Creation
Compress
Decompress
Encrypt
Block Creation
Parsing Block
120
Document
Document & Metadata Encryption
100
110
Original Document
130

FIG. 3

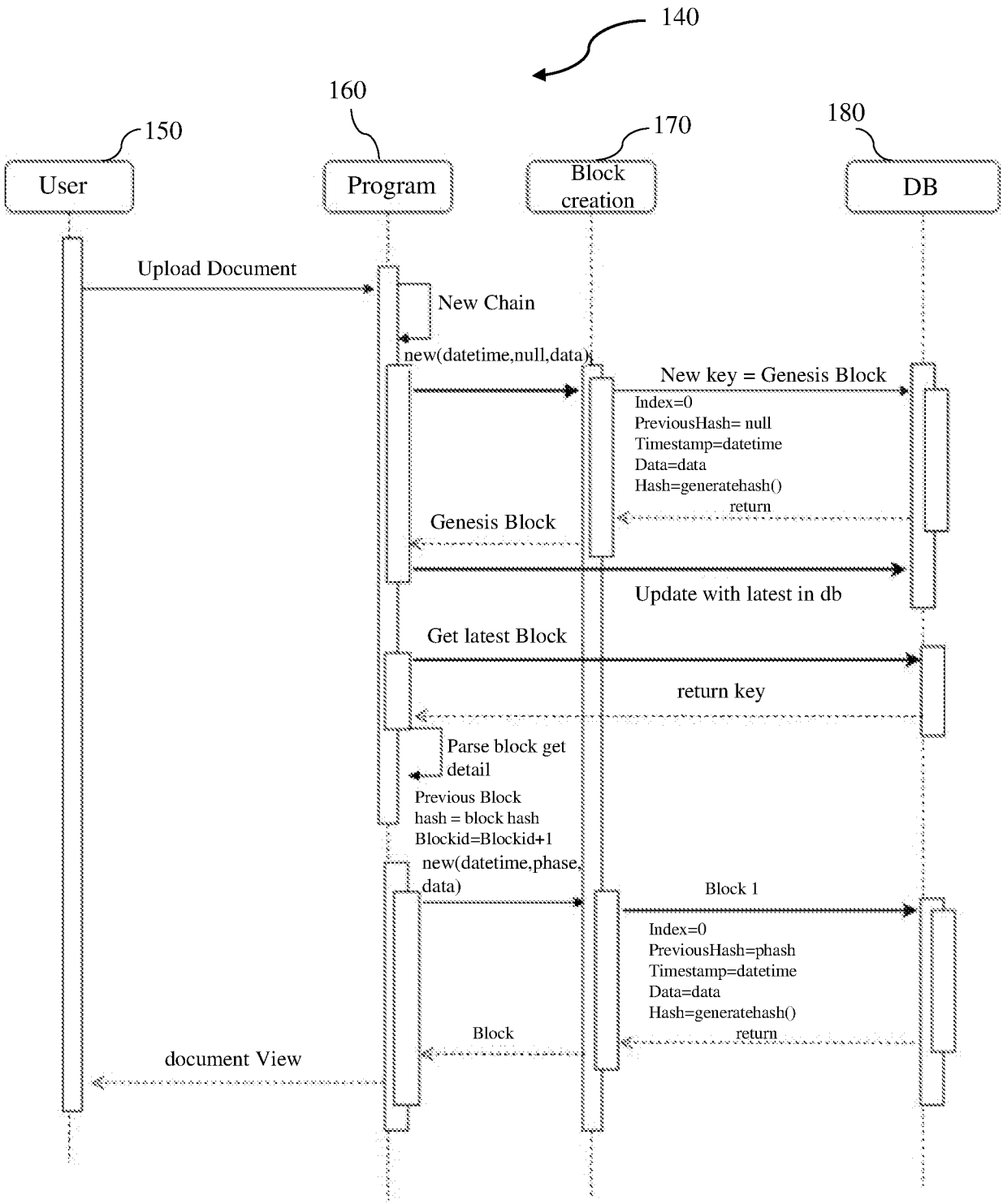
140
160
150
170
180
User
Program
Block creation
DB
Upload Document
New Chain
new(datetime,null,data)
New key = Genesis Block
Index=0
PreviousHash= null
Timestamp=datetime
Data=data
Hash=generatehash()
return
Genesis Block
Update with latest in db
Get latest Block
return key
Parse block get detail
Previous Block hash = block hash
Blockid=Blockid+1
new(datetime,phase, data)
Block 1
Index=0
PreviousHash=phash
Timestamp=datetime
Data=data
Hash=generatehash()
return
Block
document View

FIG. 4

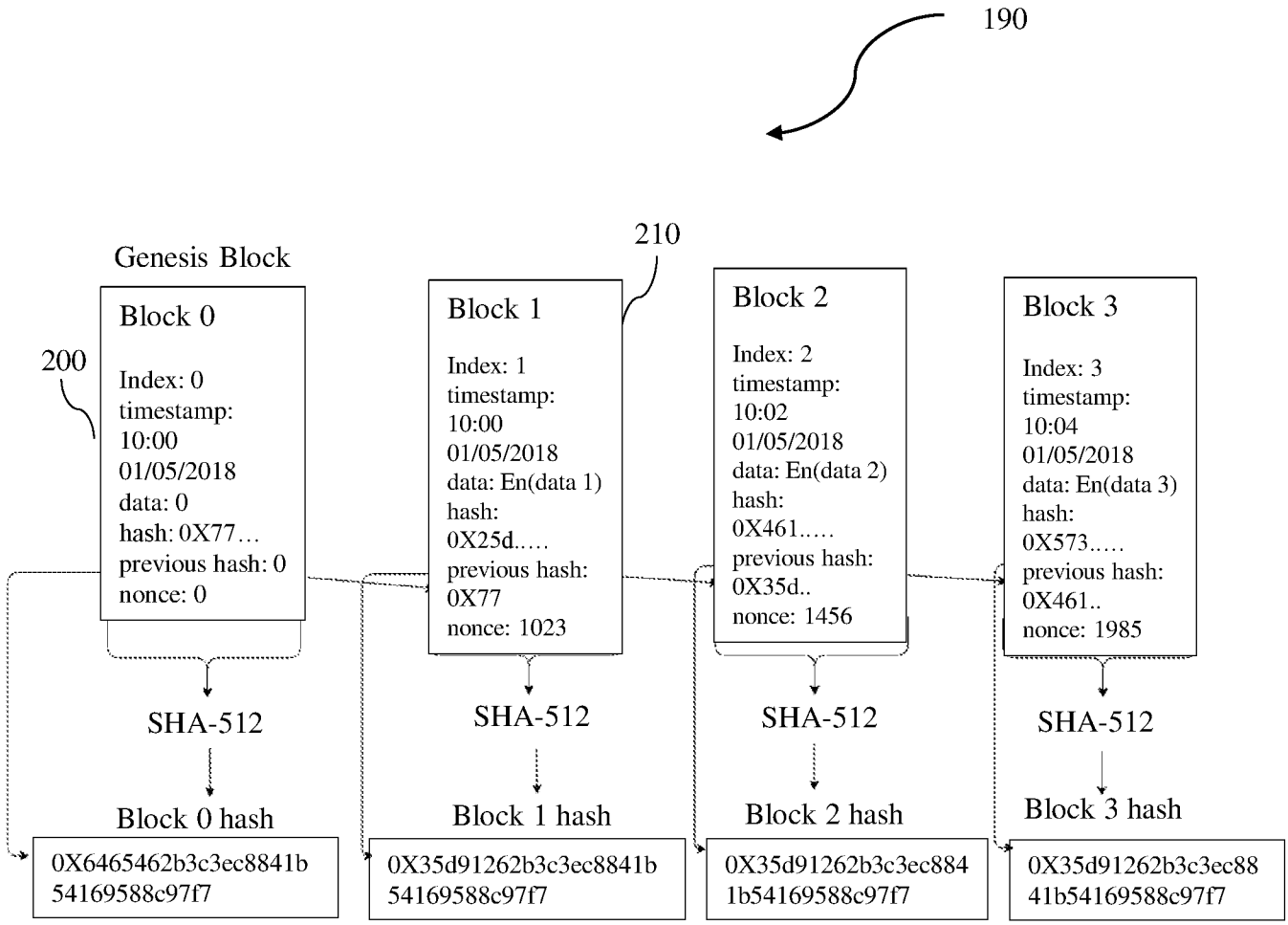
190
Genesis Block
Block 0
200
Index: 0
timestamp:
10:00
01/05/2018
data: 0
hash: 0X77…
previous hash: 0
nonce: 0
210
Block 1
Index: 1
timestamp:
10:00
01/05/2018
data: En(data 1)
hash:
0X25d.....
previous hash:
0X77
nonce: 1023
Block 2
Index: 2
timestamp:
10:02
01/05/2018
data: En(data 2)
hash:
0X461.....
previous hash:
0X35d..
nonce: 1456
Block 3
Index: 3
timestamp:
10:04
01/05/2018
data: En(data 3)
hash:
0X573.....
previous hash:
0X461..
nonce: 1985
SHA-512
SHA-512
SHA-512
SHA-512
Block 0 hash
0X6465462b3c3ec8841b
54169588c97f7
Block 1 hash
0X35d91262b3c3ec8841b
54169588c97f7
Block 2 hash
0X35d91262b3c3ec884
1b54169588c97f7
Block 3 hash
0X35d91262b3c3ec88
41b54169588c97f7

FIG. 5

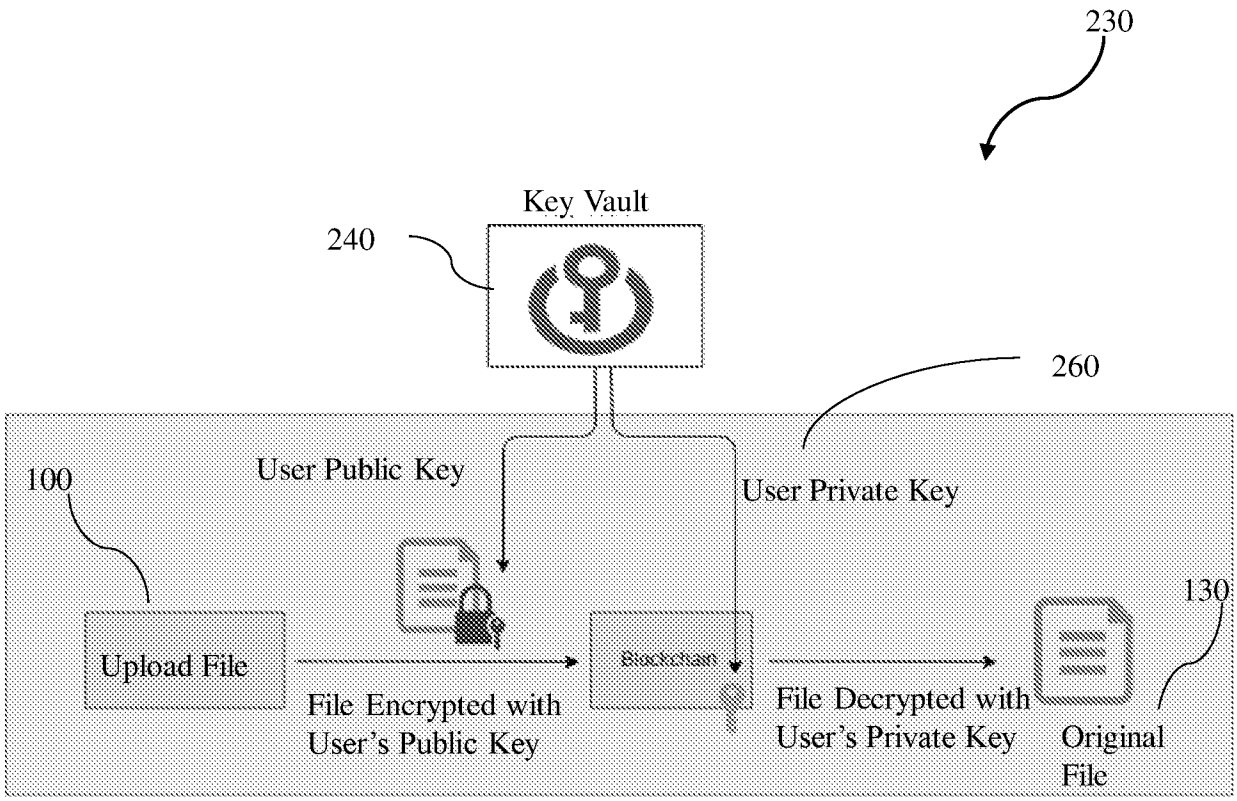
230
Key Vault
240
260
User Public Key
User Private Key
100
Upload File
Blockchain
130
File Encrypted with
User's Public Key
File Decrypted with
User's Private Key
Original
File

FIG. 6

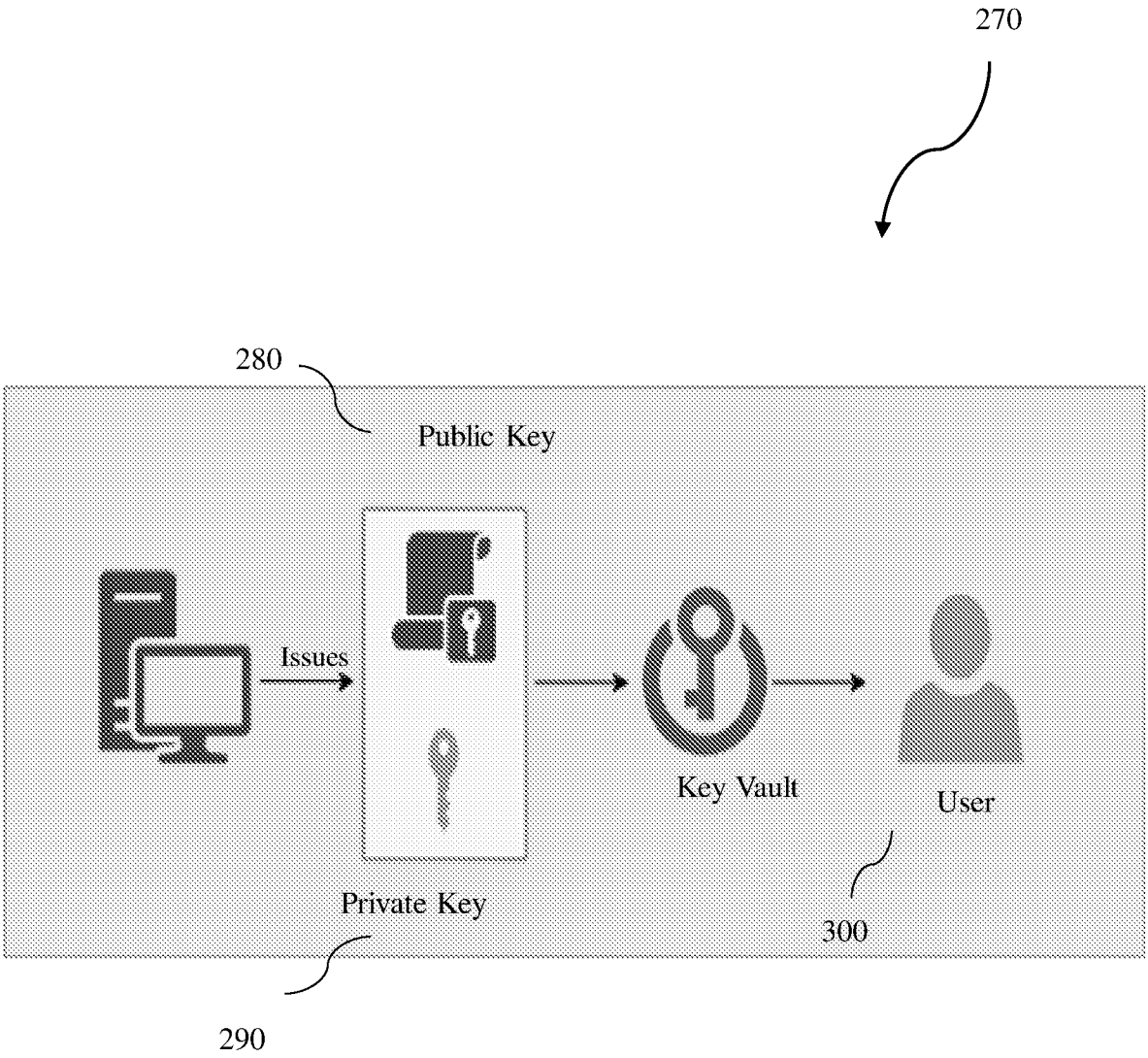

FIG. 7

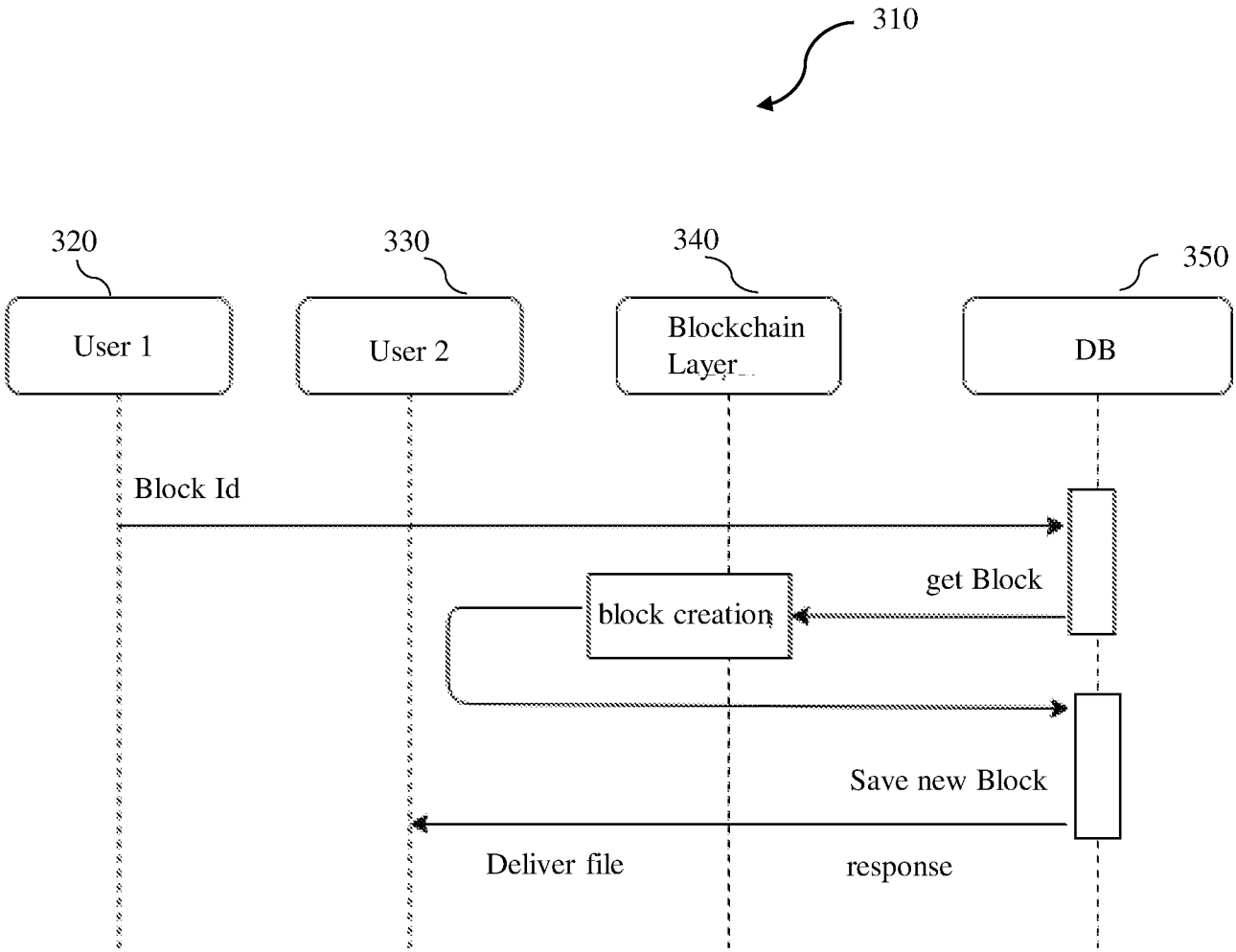
310
320
330
340
350
User 1
User 2
Blockchain Layer
DB
Block Id
get Block
block creation
Save new Block
Deliver file
response

FIG. 8

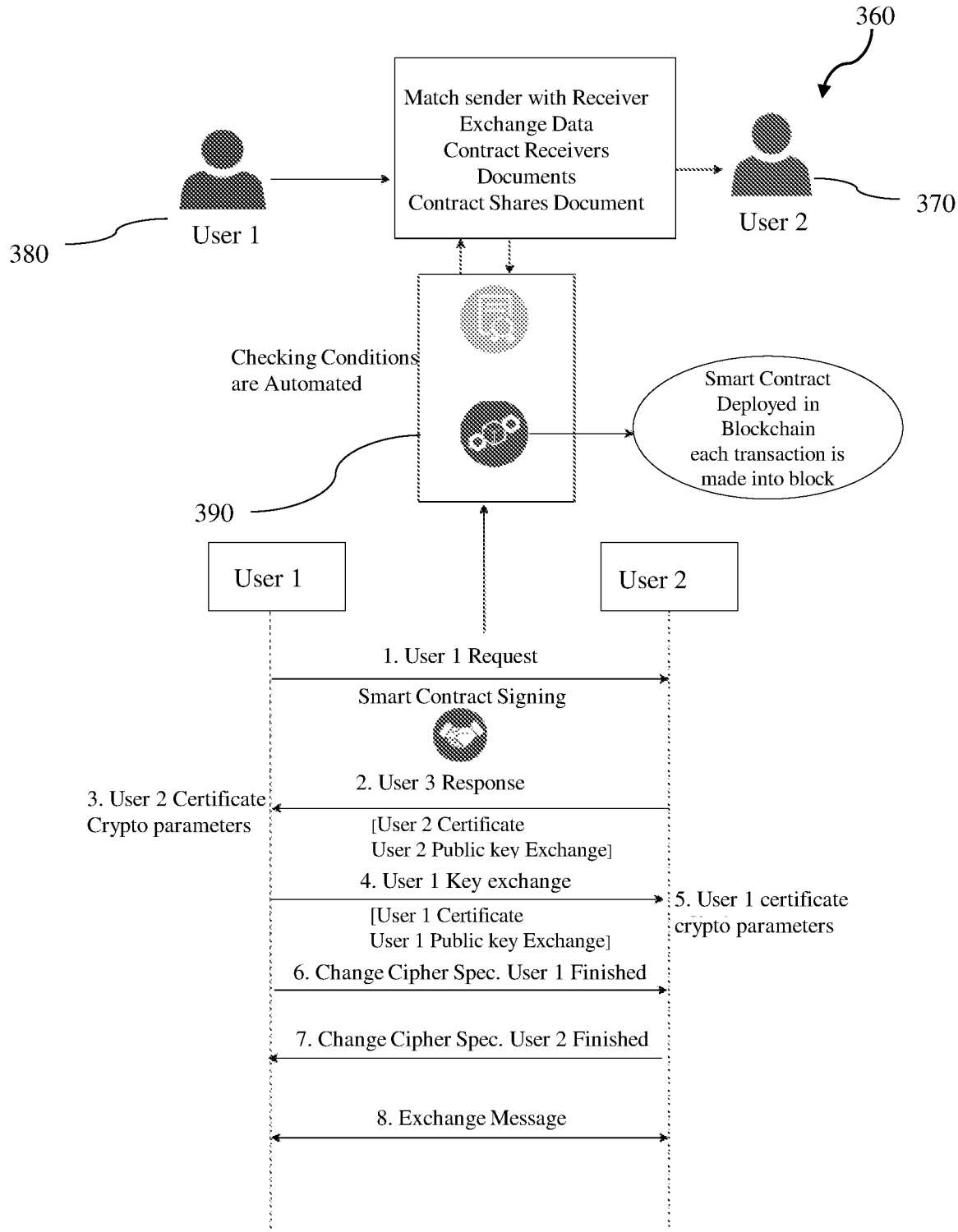
360
Match sender with Receiver
Exchange Data
Contract Receivers
Documents
Contract Shares Document
User 1
User 2
380
370
Checking Conditions
are Automated
Smart Contract
Deployed in
Blockchain
each transaction is
made into block
390
User 1
User 2
1. User 1 Request
Smart Contract Signing
2. User 3 Response
3. User 2 Certificate
Crypto parameters
[User 2 Certificate
User 2 Public key Exchange]
4. User 1 Key exchange
5. User 1 certificate
crypto parameters
[User 1 Certificate
User 1 Public key Exchange]
6. Change Cipher Spec. User 1 Finished
7. Change Cipher Spec. User 2 Finished
8. Exchange Message

FIG. 9

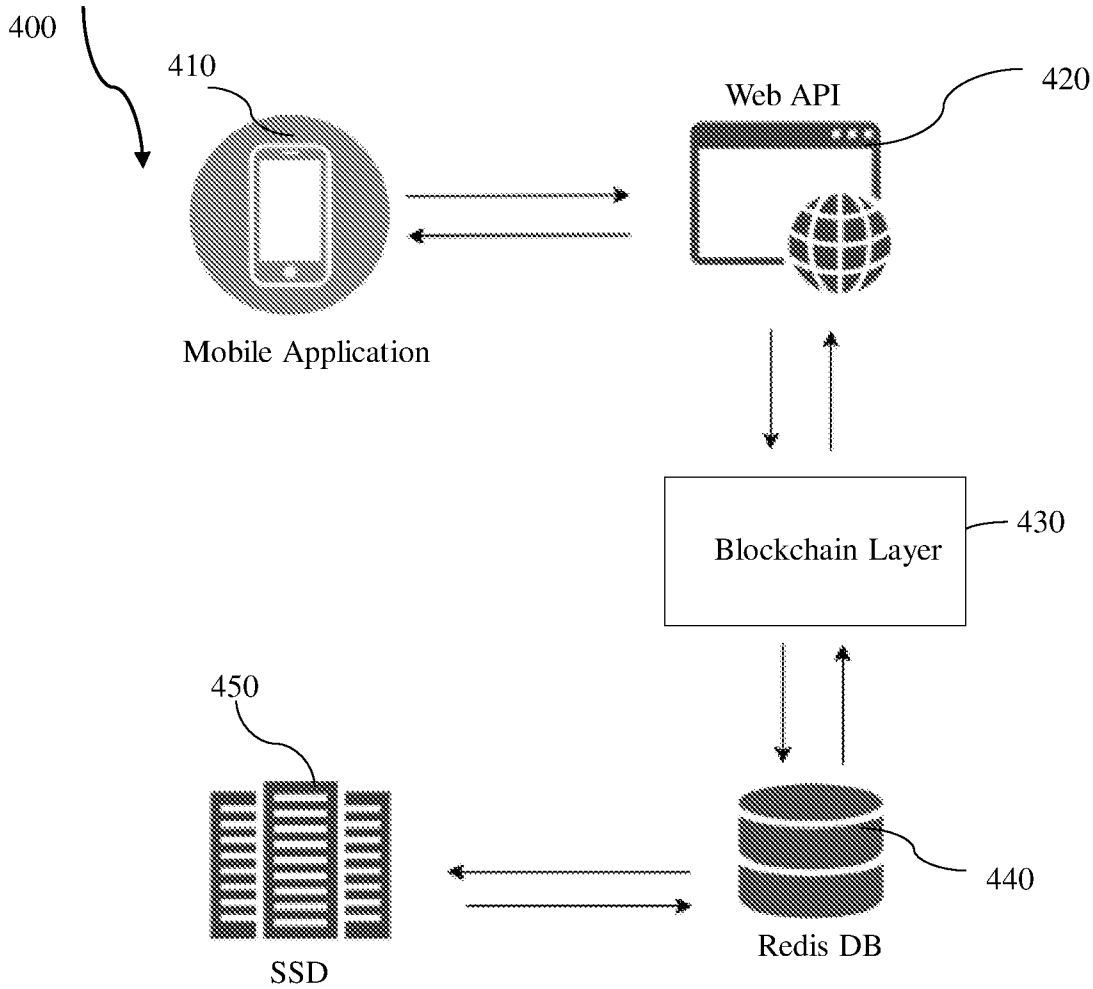
400
410
Web API
420
Mobile Application
Blockchain Layer
430
450
440
SSD
Redis DB

FIG. 10

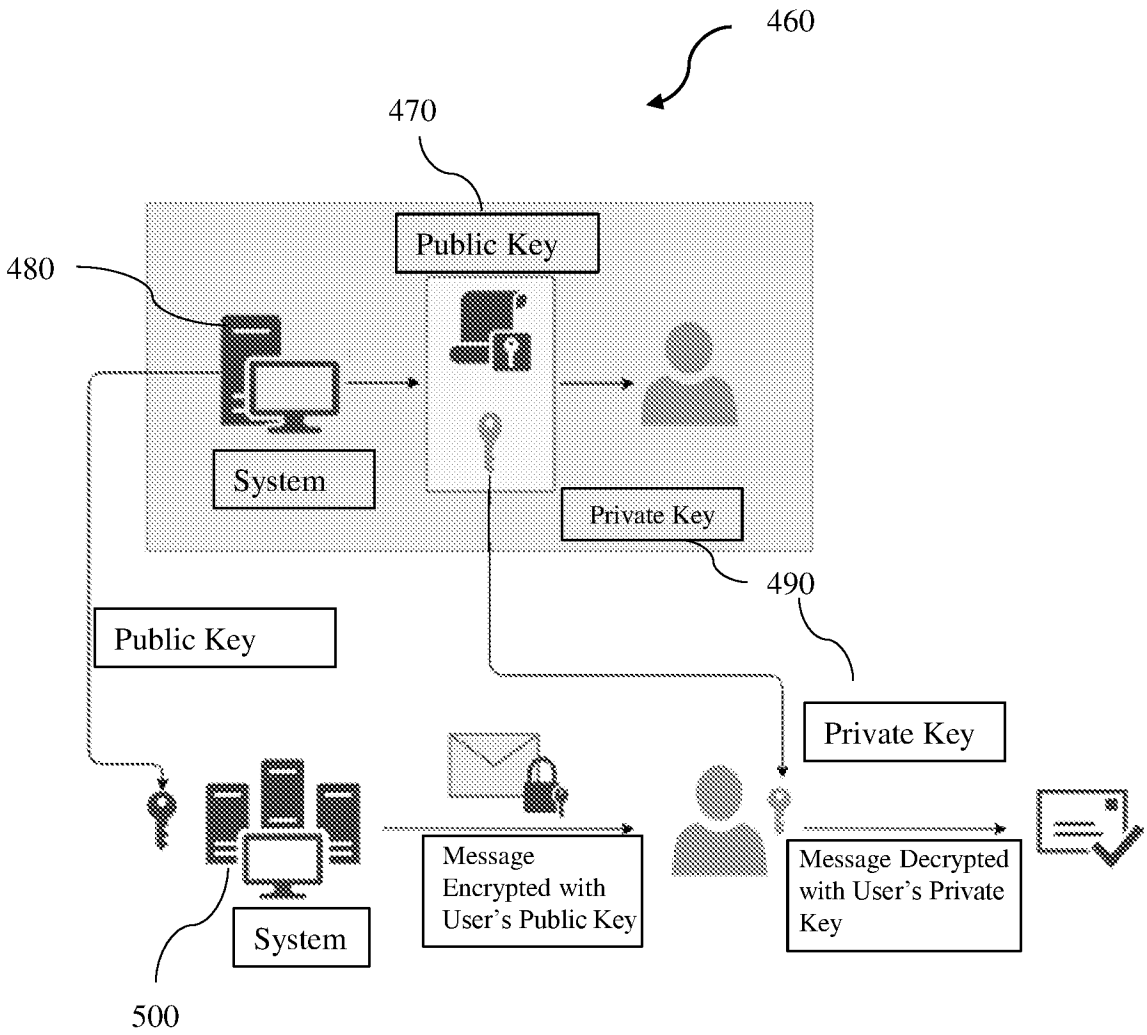
460
470
Public Key
480
System
Private Key
490
Public Key
Private Key
Message Encrypted with User's Public Key
Message Decrypted with User's Private Key
System
500

FIG. 11

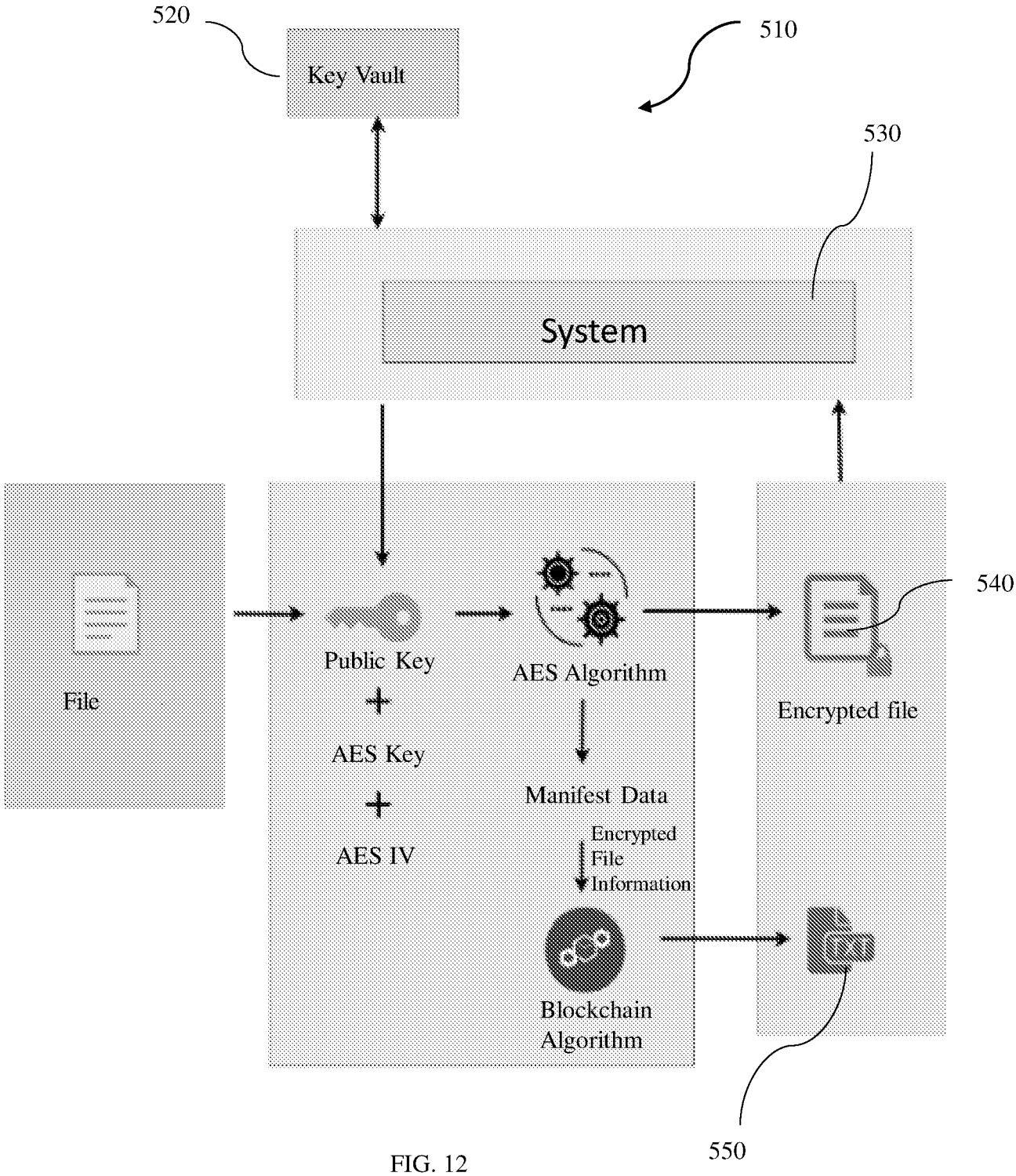
520
Key Vault
510
530
System
File
Public Key
+
AES Key
+
AES IV
AES Algorithm
Manifest Data
Encrypted
File
Information
Blockchain
Algorithm
Encrypted file
540
550

FIG. 12

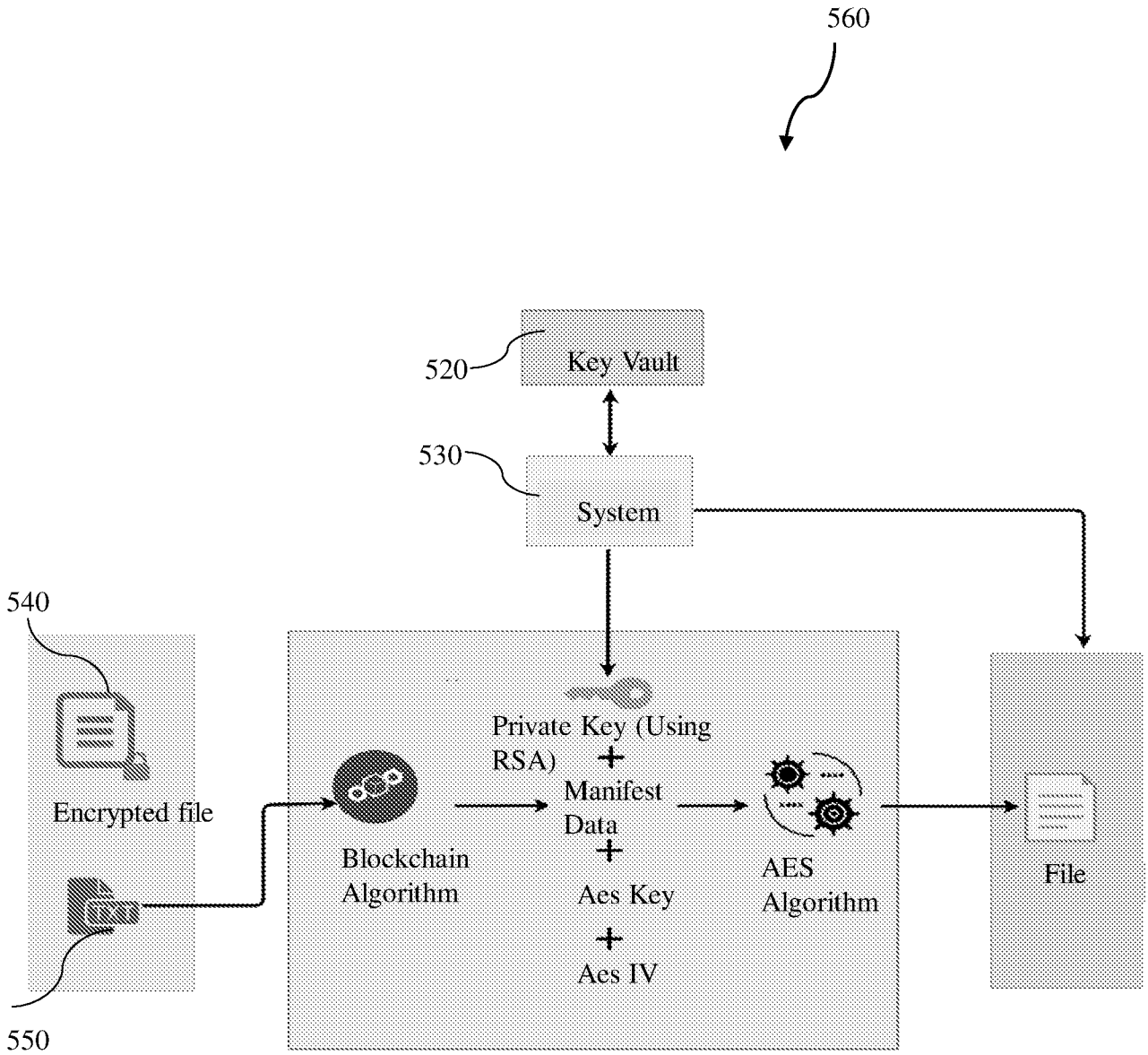
560
520
Key Vault
530
System
540
Encrypted file
550
Private Key (Using RSA)
+
Manifest Data
+
Aes Key
+
Aes IV
Blockchain Algorithm
AES Algorithm
File

FIG. 13

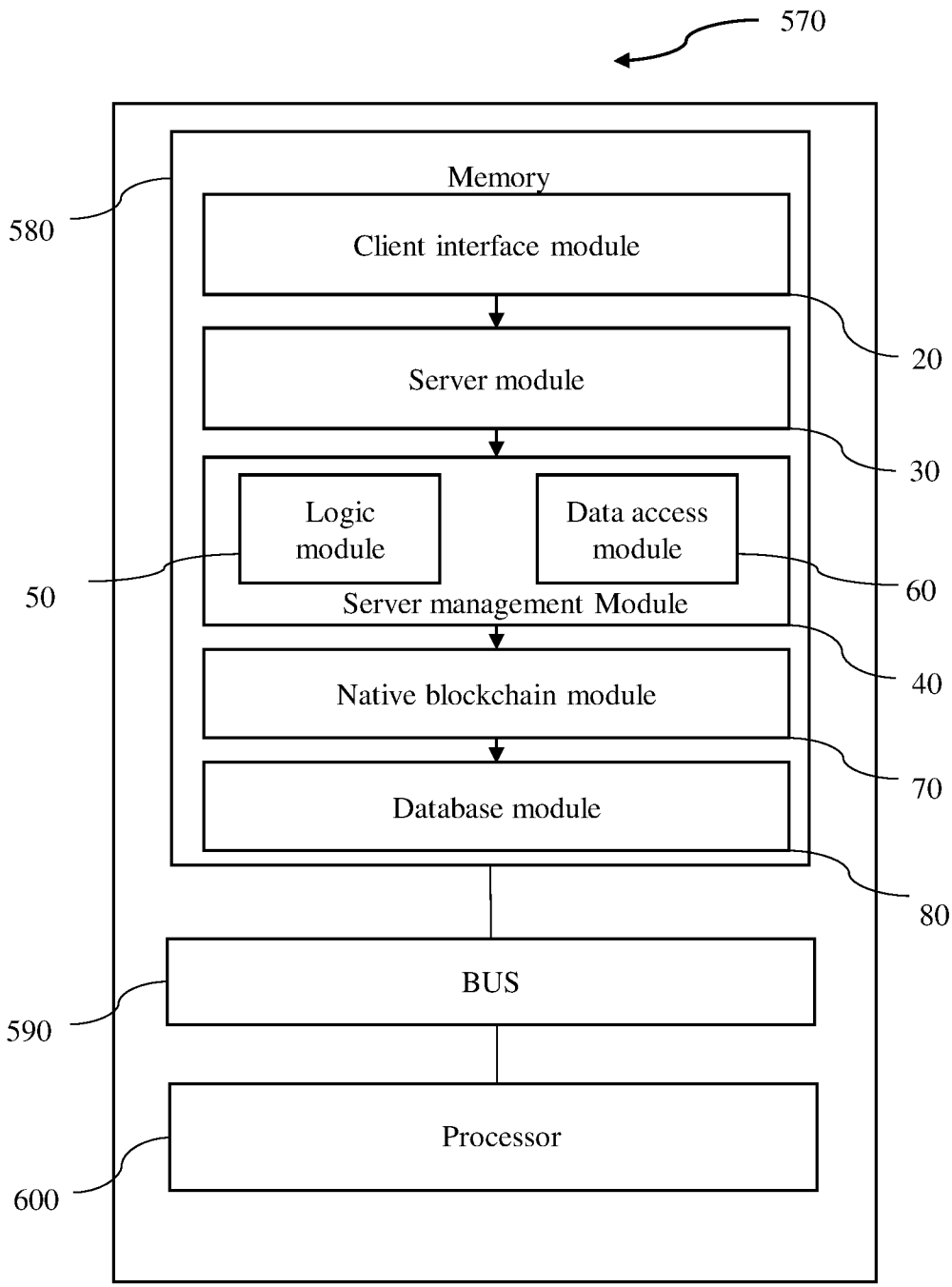
570
Memory
580
Client interface module
20
Server module
30
Logic module
Data access module
50
60
Server management Module
Native blockchain module
40
Database module
70
80
BUS
590
Processor
600

FIG. 14

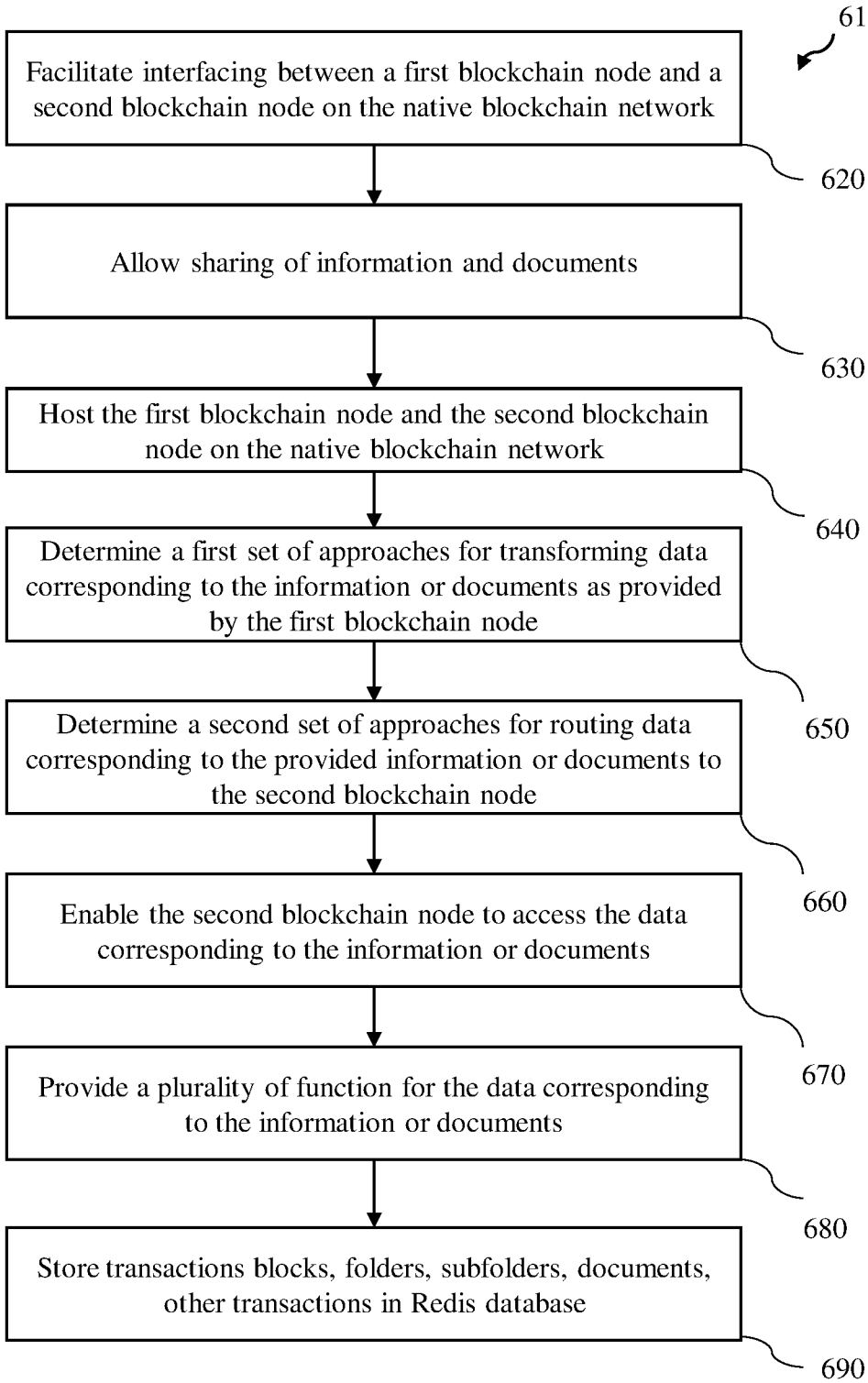
610
Facilitate interfacing between a first blockchain node and a second blockchain node on the native blockchain network
620
Allow sharing of information and documents
630
Host the first blockchain node and the second blockchain node on the native blockchain network
640
Determine a first set of approaches for transforming data corresponding to the information or documents as provided by the first blockchain node
650
Determine a second set of approaches for routing data corresponding to the provided information or documents to the second blockchain node
660
Enable the second blockchain node to access the data corresponding to the information or documents
670
Provide a plurality of function for the data corresponding to the information or documents
680
Store transactions blocks, folders, subfolders, documents, other transactions in Redis database
690

FIG. 15

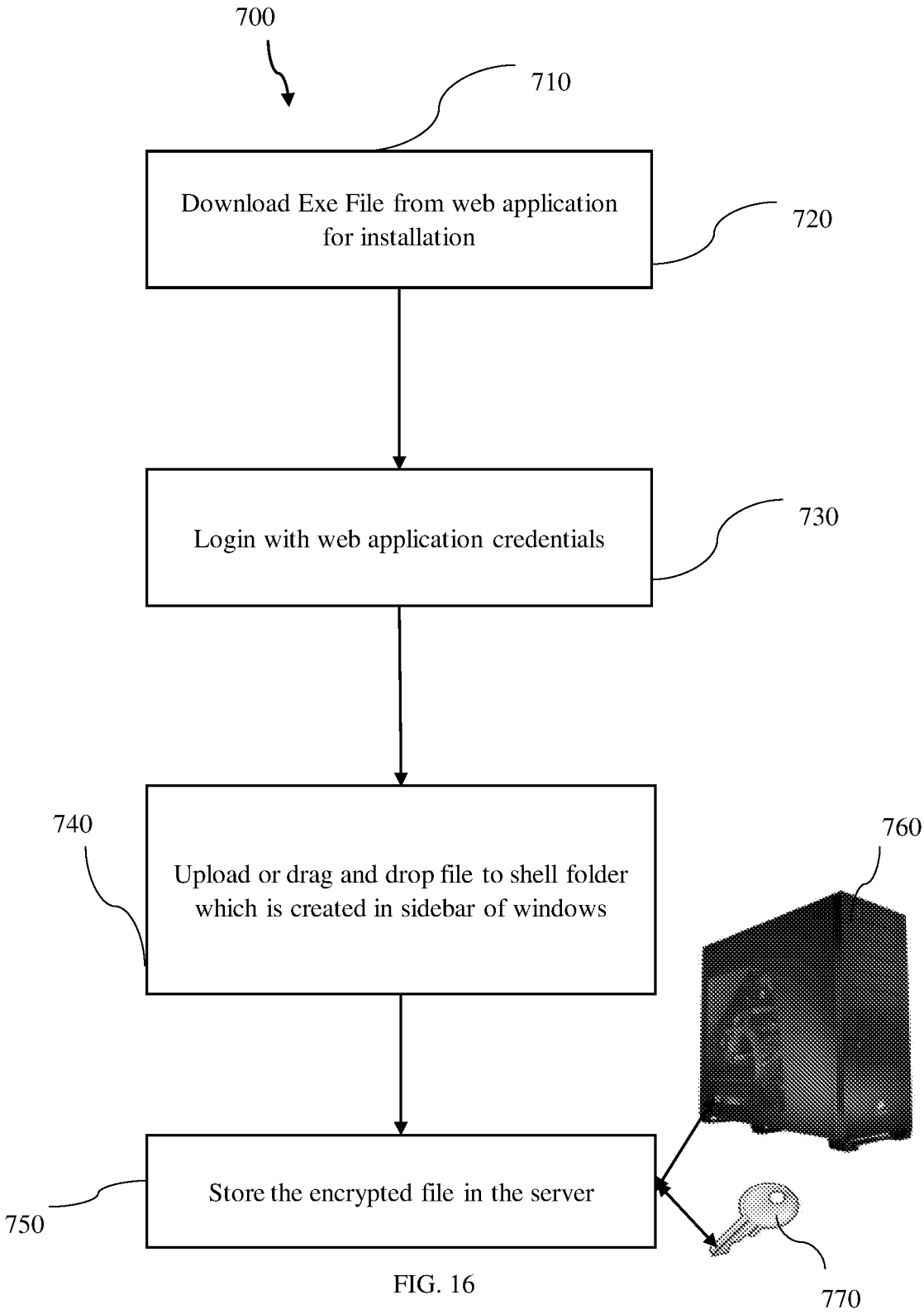
700
710
Download Exe File from web application for installation
720
Login with web application credentials
730
740
Upload or drag and drop file to shell folder which is created in sidebar of windows
760
Store the encrypted file in the server
750
770

FIG. 16

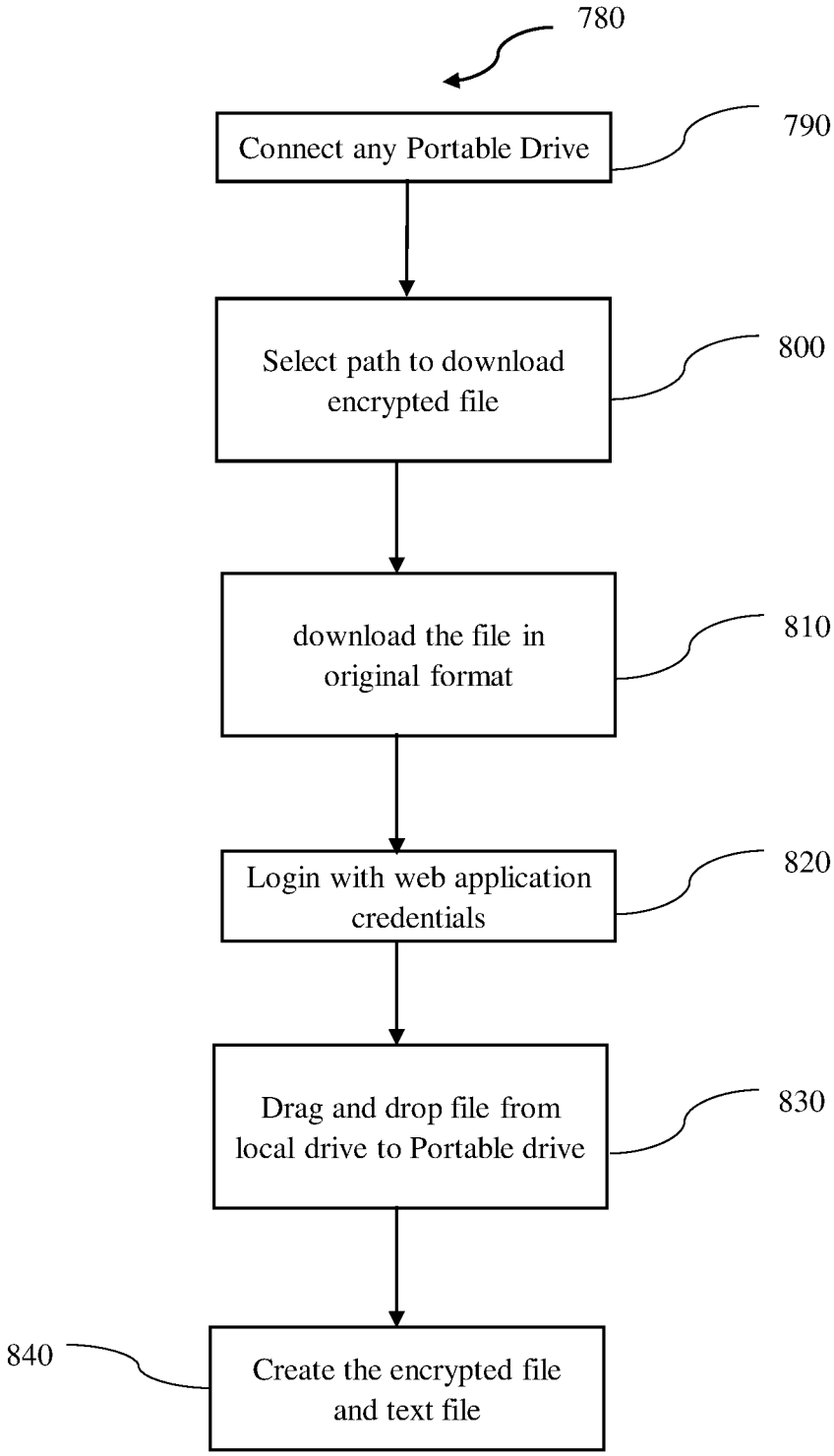
780
Connect any Portable Drive
790
Select path to download encrypted file
800
download the file in original format
810
Login with web application credentials
820
Drag and drop file from local drive to Portable drive
830
840
Create the encrypted file and text file

FIG. 17

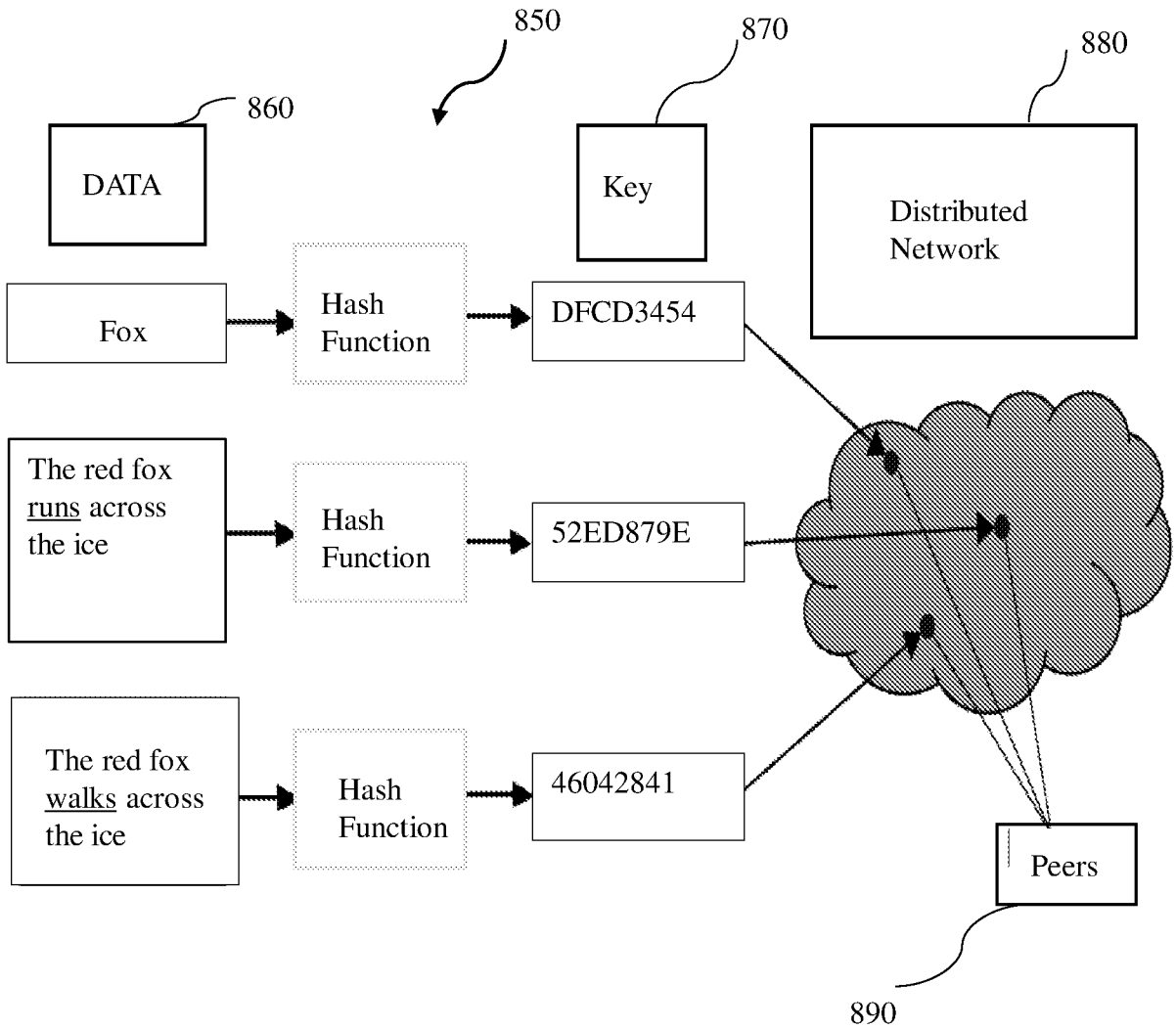
850
870
880
860
DATA
Key
Distributed Network
Fox
Hash Function
DFCD3454
The red fox runs across the ice
Hash Function
52ED879E
The red fox walks across the ice
Hash Function
46042841
Peers
890

FIG. 18

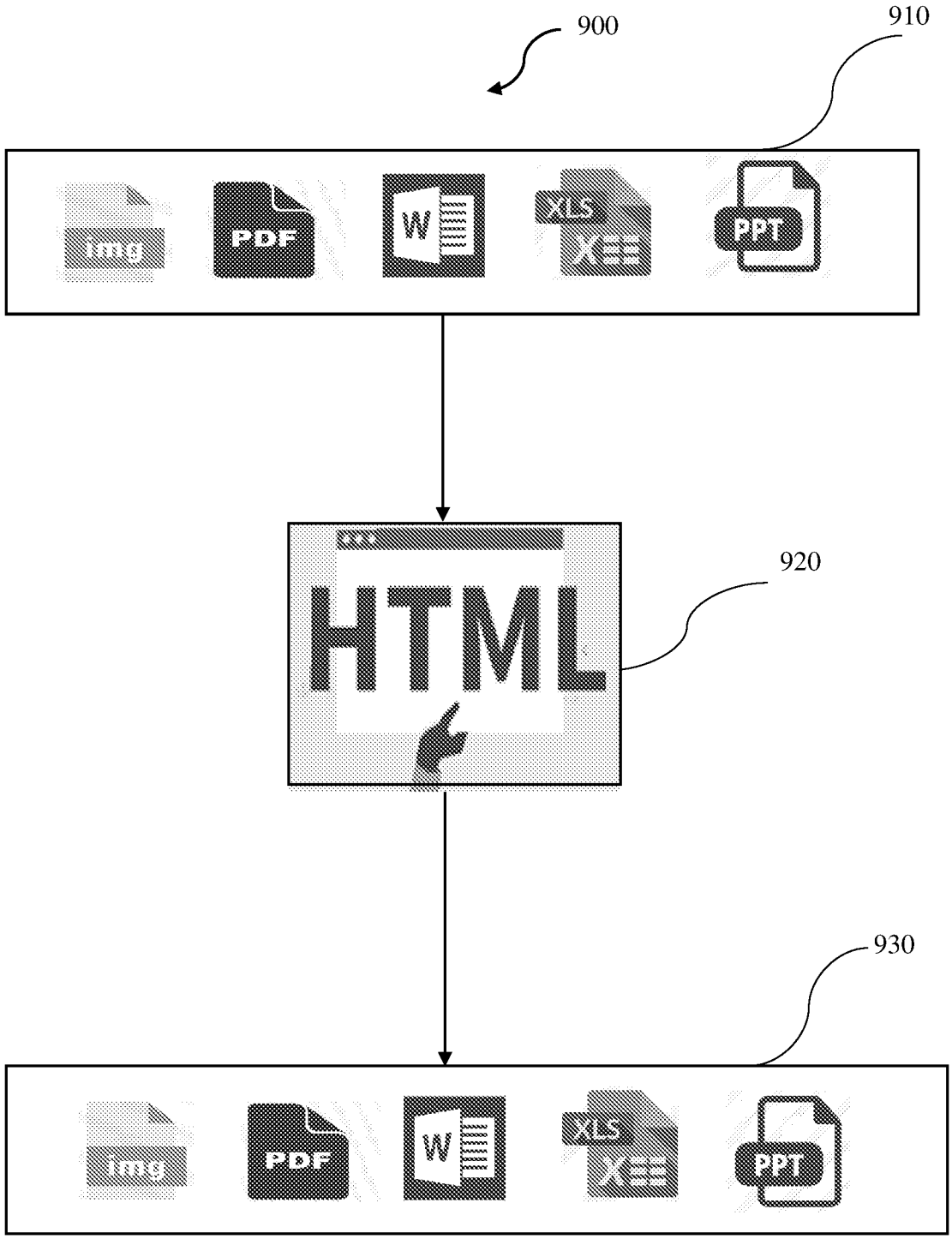

FIG. 19

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2020/056892

A. CLASSIFICATION OF SUBJECT MATTER
H04L9/06,H04L12/24,H04L12/26,H04L29/08 Version=2020.01

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Databases - Total Patent One, IPO Internal Database.
Searched Terms - Blockchain, Smart contract, Public-Private key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2020036657 A1 (THE ASSAY DEPOT, INC.); 20 February 2020 (20/02/2020) Whole Document	1-18
A	WO 2018064645 A1 (3DPP, LLC); 05 April 2018 (05/04/2018) Whole Document	1-18

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26-10-2020	26-10-2020
Name and mailing address of the ISA/ Indian Patent Office Plot No.32, Sector 14,Dwarka,New Delhi-110075 Facsimile No.	Authorized officer Nikhil Katiyar Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2020/056892

Citation	Pub.Date	Family	Pub.Date
WO 2018064645 A1	05-04-2018	US 20180096175 A1	05-04-2018
		EP 3519159 A4	25-03-2020