SYSTEMS AND METHODS FOR PROVIDING A PERSONAL DISTRIBUTED LEDGER

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 62/346,097, filed on June 6, 2016, and U.S. Patent Application No. 15/612,288, filed on June 2, 2017, each of which are hereby incorporated by reference herein in their entireties.

BACKGROUND INFORMATION

[0002] Distributed ledger systems are increasingly used to implement currency, financial transactions and other functions. Often, it is desirable for a user to conduct transactions with such systems anonymously, in order to conceal their financial positions, etc. One way for users to conduct transactions anonymously within a distributed ledger system is to generate and use a different public address and private key for each transaction. As users typically store private keys used to sign transactions in a secure location outside the distributed ledger system, it is usually not possible to connect the identity of the user to the transaction through the private key.

[0003] In tension with this, know-your-customer regulatory mandates typically require banks, financial institutions and other entities to verify the identity of their customers. Know-your-customer mandates help prevent criminals from using transactions conducted through these institutions to perform money laundering and other crimes. However, a user conducting transactions anonymously through the use of newly generated addresses and keys for every transaction stifles satisfaction of know-your-customer mandates.

[0004] Thus, a need exists for systems and methods to securely but reliably link transactions conducted anonymously in a distributed ledger system and the user conducting the transactions.

SUMMARY

[0004a] According to a first aspect of the present invention, there is provided a method of conducting transactions, the method comprising: generating, by a wallet distributed application, an address in a public distributed ledger system and a private key; signing, by the wallet distributed application, a first transaction to a distributed application in the public distributed ledger system from the generated address using the private key; generating, by the wallet distributed application, a second transaction to a personal

distributed ledger system, the generated second transaction containing information identifying the first transaction to the public distributed ledger system; and transmitting, by the wallet distributed application, the second transaction to at least one distributed node of the personal distributed ledger system.

[0004b] According to a second aspect of the present invention, there is provided a system for conducting transactions, the system comprising: at least one processor; and a non-transitory machine-readable storage medium having program instructions, which when executed by a processor perform a method of conducting transactions, the method comprising: generating, by a wallet distributed application, an address in a public distributed ledger system and a private key; signing, by the wallet distributed application, a first transaction to a distributed application in the public distributed ledger system from the generated address and using the private key; generating, by the wallet distributed application, a second transaction to a personal distributed ledger system, the generated second transaction containing information identifying the first transaction to the public distributed ledger system; and transmitting, by the wallet distributed application, the second transaction to at least one distributed node of the personal distributed ledger system.

[0004c] According to a third aspect of the present invention, there is provided a method of authorizing a transaction in a public distributed ledger system using a personal distributed ledger system, the method comprising: reading, by a distributed application of a financial institution system hosting a node of the personal distributed ledger system, information identifying at least one first transaction conducted in the public distributed ledger system stored in the personal distributed ledger system; determining, by the financial institution system, whether to authorize a user associated with the personal distributed ledger system to conduct further transactions in the public distributed ledger system using the financial institution system based on the retrieved information identifying the transaction; and upon authorization of the user, generating, by the financial institution system, a transaction to a distributed application of the public distributed ledger system on behalf of the user.

[0004d] As disclosed, there is provided a method of storing a digital object using a personal distributed ledger system, further comprising: receiving, via a distributed application of the personal distributed ledger system, the digital object; determining whether a storage size of the digital object is above a predetermined threshold; upon determining that the storage size of the digital object is below the predetermined threshold: generating a

transaction to store the digital object in the personal distributed ledger system, and transmitting the transaction to at least one distributed node of the personal distributed ledger system; and upon determining that the storage size of the digital object is above the predetermined threshold: determining a representation of the digital object, storing the digital object in a secure storage system outside of the personal distributed ledger system, generating a transaction to store the representation of the digital object in the personal distributed ledger system, and transmitting the transaction to at least one distributed node of the personal distributed ledger system.

[0004e] As disclosed, there is provided a method of distributing a digital object using a persona distributed ledger system, the method comprising: receiving, via a distributed application of the personal distributed ledger system, a request for the digital object; determining whether the digital object or a representation of the digital object is stored in personal distributed ledger system; upon determining that the digital object is stored in the personal distributed ledger system: reading via the access distributed application the personal distributed ledger system to retrieve the digital object, and providing the digital object to a requesting entity; and upon determining that the encoding of the digital object is stored in the personal distributed ledger system: reading via the access distributed application the personal distributed ledger system to retrieve the representation of the digital object, retrieving the digital object from an asset store, determining a representation of the retrieved digital object, determining whether the determined representation matches the retrieved representation, and upon determining that the determined representation matches the retrieved representation, providing the digital object to the requesting entity.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] So that the features of the present invention can be understood, a number of drawings are described below. However, the appended drawings illustrate only particular embodiments of the invention and are therefore not to be considered limiting of its scope, for the invention may encompass other equally effective embodiments.

[0006] FIG. 1 is a schematic diagram depicting an embodiment of a system for providing a personal distributed ledger.

[0007] FIG. 2 is a schematic diagram depicting an embodiment of a distributed ledger system.

[0008] FIG. 3 is a schematic diagram depicting an embodiment of a distributed node of a distributed ledger system.

[0009] FIG. 4 is a schematic diagram depicting another embodiment of a system for providing a personal distributed ledger.

[0010] FIG. 5 is a flowchart depicting an embodiment of a method of using a personal distributed ledger system to record details of transactions conducted in a public distributed ledger system.

[0011] FIG. 6 is a schematic diagram depicting another embodiment of a system for providing a personal distributed ledger.

[0012] FIG. 7 is a flowchart depicting an embodiment of a method of using a personal distributed ledger system to confirm a record of transactions conducted in a public distributed ledger system.

[0013] FIG. 8 is a schematic diagram depicting another embodiment of a system for providing a personal distributed ledger.

[0014] FIG. 9 is a schematic diagram depicting another embodiment of a system for providing a personal distributed ledger.

[0015] FIG. 10 is a flowchart depicting an embodiment of a method of using a personal distributed ledger system to store digital assets, identity data and/or custom data.

[0016] FIG. 11 is a flowchart depicting an embodiment of a method of using a personal distributed ledger system to distribute digital assets, identity data and/or custom data.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0017] Embodiments of a system for providing a personal distributed ledger record of details of transactions conducted in a public distributed ledger system in a personal distributed ledger system, enabling an immutable, secure and reliable link between the transactions conducted in the public ledger and the user conducting the transactions.

[0018] To conduct transactions in the public distributed ledger system, a wallet distributed application may be invoked by a distributed application of the public distributed ledger system. The wallet application may generate an address in the public distributed

ledger system and a private key, sign the transaction using the generated address and key, and transmit the signed transaction to a distributed node of the public distributed ledger system. To record details of the transaction in a personal distributed ledger, the wallet application also may generate a transaction invoking a smart contract in the personal distributed ledger system and containing details of the transaction conducted in the public distributed ledger system, such as the private key used to sign the transaction and an identifier identifying the transaction, and transmit the generated transaction to a distributed node of the personal distributed ledger system. Upon execution, the transaction may invoke a storage function of the contract, which may store the contained transaction information in a data structure of the personal distributed ledger system associated with the contract.

[0019] To satisfy know-your-customer mandates, a validator system, such as operated by a financial institution, may be allowed selective access to the personal distributed ledger system to confirm to the operator of the validator system details of the record of transactions conducted in the public distributed ledger system stored in the personal distributed ledger system. The financial institution may utilize this information to determine whether to authorize the user to use the financial institution's systems to conduct further transactions in the public distributed ledger system.

[0020] The personal distributed ledger system also may be used to store personal data of the user, such as digital assets, identity data, and custom data, or encodings of such, and subsequently distribute such data to authorized entities.

[0021] FIG. 1 depicts an embodiment of a system 20 for providing a personal distributed ledger for recording transactions conducted in a public distributed ledger to create an immutable, secure and reliable link between the transactions conducted in the public ledger and the user conducting the transactions. The system 20 includes a personal distributed ledger system (DLS) 24, one or more distributed applications (DAPPs) 28 based in the personal DLS 24, one or more public DLSs 32, and one or more DAPPs 36 based in the public DLSs 32. The personal DLS 24 and personal DLS DAPPs 28 provide a distributed ledger and corresponding interfaces separate from the public DLSs 32, with access restricted to only the user associated with the personal DLS 24 and entities authorized by the user, to provide the transaction recording and other personal data functions discussed herein. The public DLSs 32 and public DLS DAPPs 36 provide distributed ledgers and corresponding interfaces open to use by the public at large, or at least more than the user, to provide the

various functions that DLSs may provide, such as monetary systems, financial transactions, etc.

[0022] FIG. 2 depicts an exemplary embodiment of a DLS 40 that may be used to individually implement each of the personal DLS 24 and public DLSs 32. The DLS 40 includes a plurality of distributed nodes 44. The distributed nodes 44 are organized as a peer-to-peer network, in which each of the nodes 44 may connect to one or more of the other nodes 44 using a peer-to-peer communication protocol. At least one of the distributed nodes 44 also connects to a front end of a DAPP based in the DLS 40, such as a wallet DAPP front end in the case of the personal DLS 24 or a public DLS DAPP front end in the case of the public DLS 32. As a peer-to-peer network, the configuration of connections between individual distributed nodes 44 may change over time according to operation of the peer-to-peer protocol.

[0023] FIG. 3 depicts an exemplary embodiment of the distributed node 44 of the DLS, including a ledger structure storage module 48, a communication module 52, a virtual machine module 56 and a ledger structure creation module 60. The ledger structure storage module 48 stores data structures into which the ledger of the DLS 40 is organized. The communication module 52 performs communications between the distributed node 44 and other distributed nodes 44 and other systems or components connected to the distributed node 44, such as DAPP front ends. The virtual machine module 56 executes contracts stored on the distributed ledger of the DLS 40. The ledger structure creation module 60 performs an algorithm to incorporate new transactions and other data into ledger structures of the DLS 40, such as a data encryption algorithm of a selected complexity.

[0024] In embodiments, the distributed node 44 may include only any subset of, or an alternative connection of, the features depicted in or discussed herein in regard to Fig. 3.

[0025] In embodiments, the DLS 40 may be a blockchain system in which the ledger implemented by the DLS 40 is in the form of a sequence of structured data blocks, also referred to as a blockchain. In such embodiments, the ledger structure storage module 48 is a block storage module that stores the blocks of the blockchain system, and the ledger structure creation module 60 is a block creation module performing an algorithm to incorporate new transactions and other data into blocks of the blockchain of the blockchain system, also referred to as mining blocks of the blockchain system. In other embodiments, the DLS 40

may be another type of DLS 40 instead of a blockchain system, in which the ledger may be organized into a structure other than blocks of a blockchain.

[0026] In embodiments, the personal DLS 24 and public DLS 32 may be implemented using the same or different types of DLSs.

[0027] FIG. 4 depicts an embodiment 20B of the system of FIG. 1 showing embodiments of the personal DLS DAPP 28 and a public DLS DAPP 36 in greater detail. In the depicted embodiment, the personal DLS DAPP 28 is a wallet DAPP 30. The wallet DAPP 30 is configured to provide wallet functions for conducting transactions in the public DLSs 32 and recording functions for recording the conducted transactions in the personal DLS 24.

[0028] The wallet DAPP 30 includes a front end component 64 and a DLS component 68. The front end component 64 includes one or more program modules to provide functions of the wallet DAPP. The program modules may include a transaction module 72, an address and key generator module 76, and a control module 80. The transaction module 72 is configured to receive a request to sign a transaction from a public DLS DAPP 36, invoke the address and key generator module 76 to generate an address and/or public-private key pair for the transaction, sign the transaction using the generated address and private key, and transmit the transaction to at least one distributed node of the public DLS 32. The transaction module 72 is also configured to generate a transaction to the personal DLS 24 containing information related to the transaction conducted in the public DLS 32, and transmit the transaction to at least one distributed node of the personal DLS 24. The address and key generator module 76 is configured to generate new addresses in a public DLS address space and new public-private key pairs in response to a request from the transaction module 72, and provide the generated address and key pairs to the transaction module 72. The control module 80 is configured to control operation of the modules of the wallet DAPP 30 to provide the functions discussed herein.

[0029] The DLS component 68 of the wallet DAPP 30 includes one or more smart contracts 84 stored in the distributed ledger of the personal DLS 24 and one or more associated data storage structures 88, 92. The wallet DAPP contract 84 includes one or more program functions to be invoked by a transaction or call to the contract 84 to perform the functions of the wallet DAPP contract 84. The program functions may include a transaction information storage function and a transaction information read function. The transaction information storage function is invoked by a transaction to the contract 84, and is configured

to receive as an input information related to a transaction conducted in the public DLS 32, such as a private key used to sign the transaction and an identification of the transaction, and in response store the transaction information in corresponding data structure(s) in the personal DLS, such as a key storage data structure 88 and a transaction identifier storage data structure 92. The transaction information read function is invoked by a call to the contract 84, and configured to output the stored transaction information in response to an authorized request. The associated data structures include data structures to store the transaction information stored by the wallet DAPP contract 84, including a key storage data structure 88 to store the keys such as the private keys, or public-private key pair, used to sign the transaction, and a transaction identifier storage data structure 92 to store the transaction information identifying the transaction.

[0030] The public DLS DAPP 36 also includes a front end component 96 and a DLS component 100. The front end component 96 includes one or more program modules to provide functions of the DAPP. The DLS component 100 includes one or more smart contracts and one or more associated data storage structures stored in the distributed ledger of the public DLS. The DAPP contract includes one or more program functions to be invoked by a transaction to the contract to perform the functions of the DAPP contract.

[0031] In embodiments, the wallet DAPP 30 may include only any subset of, or an alternative connection of, the features depicted in or discussed herein in regard to Fig. 4.

[0032] FIG. 5 depicts an embodiment of a method 500 of using the personal DLS 24 to record transactions conducted in one or more of the public DLSs 32. The method begins at step 502. A first portion of the method conducts a transaction with the public DLS DAPP 32.

[0033] At step 504, the wallet DAPP front end 64 is invoked by the public DLS DAPP front end 96 to sign a transaction that the user wishes to conduct with the public DLS DAPP 36. In invoking the wallet DAPP front end 64, the public DAPP front end 96 may pass the unsigned transaction to the transaction module 72 of the wallet DAPP front end 64. At step 506, the address and key generator module 76 of the wallet DAPP front end 64 generates a new address in the address space of the public DLS 32 and a public-private key pair for signing the transaction. At step 508, the transaction module 72 of the wallet DAPP front end 64 signs the transaction using the generated address and private key. At step 510, the transaction module 72 transmits the signed transaction, addressed to the DAPP contract, to at least one distributed node of the public DLS 32. As part of the ledger structure creation

process in the public DLS 32, in due course the transaction will be incorporated into a new ledger structure and thus executed by the DAPP contract.

[0034] A second portion of the method then records the transaction conducted in with the public DLS DAPP 36 in the personal DLS 24.

[0035] At step 512, the transaction module 72 of the wallet DAPP front end 64 generates a transaction to the wallet DAPP contract 84 containing information related to the conducted transaction and configured to invoke the transaction information storage function of the wallet DAPP contract 84. The information related to the transaction may include information that may be used to link the user conducting the transaction to the transaction. In embodiments, the transaction information includes the private key used to sign the transaction. In embodiments, the transaction additionally or alternatively includes an identifier identifying the transaction. The identifier may include one or more of: an execution receipt generated by execution of the transaction, or a portion of the transaction.

[0036] At step 514, the transaction module 72 transmits the transaction containing the public DLS transaction information to at least one distributed node of the personal DLS 24. As part of the ledger structure creation process of the personal DLS 24, the transaction will be incorporated into a new ledger structure and executed by the wallet DAPP contract 84. Executing the transaction by the wallet DAPP contract 84 invokes the transaction information storage function, which stores the transaction information contained in the transaction in the corresponding data structures. For example, for transaction information including the private key and transaction identifier, the transaction information storage function stores the private key in the key storage data structure 88 and the transaction identifier in the transaction identifier storage data structure 92. The method ends at step 516.

[0037] In embodiments, a method of using a personal DLS 24 to record transactions conducted in one or more public DLSs 32 may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to Fig. 5.

[0038] In embodiments, other embodiments of conducting a transaction with the public DLS DAPP 36 using a wallet DAPP 30 may be paired with the second portion of the method 500 of FIG. 5 of recording the transaction conducted in with the public DLS DAPP 36 in the private DLS 24.

[0039] The personal DLS 24 may be configured to have a selected number of distributed nodes. While public DLSs 32 typically include a large number of distributed nodes in order

to provide an immutable, secure and reliable system without necessitating a trusted authority, in contrast, in embodiments the personal DLS 24 may include a relatively small number of distributed nodes, while still providing an immutable, secure and reliable system, due to a degree of trust that may be naturally present in the distribution of the system. In embodiments, the personal DLS 24 may include a single node. In other embodiments, the personal DLS 24 may include at least two nodes.

[0040] The personal DLS 24 also may be configured to have at least one node and at least a portion of the personal DLS DAPP(s) 28 hosted and operated for the user by a host system provider. The node hosted by the host system provider may be a master node of the personal DLS 24 that controls whether other nodes are granted permission to enter the personal DLS. The personal DLS DAPP(s) 28 are configured to provide only user designated systems, such as the host system and/or a user system, with access to control features of the DLS DAPP(s) 28 and secure information, such as the private keys. It may be advantageous to the user for such a provider to host at least a portion of the personal DLS 24 in order to free the user from performing such operations.

[0041] The personal DLS 24 also may be configured to have at least one node and at least a portion of the personal DLS DAPP(s) 28 hosted and operated by a validator system. The node hosted by the validator system may increase the immutability, security and reliability of the personal DLS, and provides a way to selectively share the contents of the personal DLS 24, such as the record of public DLS transactions stored in the personal DLS 24, with an entity outside the user or host system, such as to satisfy know-your-customer mandates.

[0042] FIG. 6 depicts an embodiment 20C of the system for providing the personal DLS 24, the personal DLS 24 including two distributed nodes, including a first, master node 44A hosted by a host system 104 and a second, validator node 44B hosted by a validator system 108. The host system 104 provides one or more computing devices to host the master node 44A of the personal DLS 24 and at least a portion of the wallet DAPP front end 64, such as a wallet DAPP front end server 112 to serve the front end 64. The validator system 108 provides one or more computing devices to host the validator node 44B of the personal DLS 24 and at least a portion of a validator DAPP front end 116. The validator DAPP front end 116 is a front end to the wallet DAPP contract 84 for providing the selective access to the personal DLS 24 for the validator system 108.

[0043] FIG. 7 depicts an embodiment of a method 700 of using the personal DLS 24 to confirm to a validator system 108 the record of public DLS transactions stored in the personal DLS 24. The method begins at step 702. At step 704, the validator DAPP front end 116 executes a read call to the wallet DAPP contract 84 to invoke the transaction information read function to read at least a portion of the transaction information stored in the personal DLS 24. The read call may be configured provide a variety of different outputs to the validator DAPP front end 116. For example, the read call may provide a list of transaction identifiers associated with the user. In another example, the read call may accept as an input an indication of the selected portion of the record to be read, such as one or more transaction identifiers, and provide as an output a confirmation that the transactions are stored with corresponding private keys in the personal DLS.

[0044] At step 706, the validator system 108 determines whether to authorize the user to conduct transactions utilizing the validator system 108 or another system controlled or authorized by the validator system 108 based on the outcome of the executed read call. In one example, the validator system 108 may be part of a financial institution system having DAPP components to conduct transactions in one or more public DLSs, and the financial institution system may decide based on the outcome of the read call whether to authorize the user to utilize the DAPP components to conduct transactions in the public DLSs. For instance, the financial institution system may enable trading in a public DLS by users using tokens that the financial institution system mints on behalf of the users, and may determine whether to enable such minting on behalf of a user based on a review the transaction record in the personal DLS 24, such as based on a review of whether the transaction record includes any fraudulent or otherwise suspicious or undesirable transactions.

[0045] If at step 706, it is determined to authorize the user to conduct transactions using the validator 108 or another system, the method proceeds to step 708. At step 708, the validator system 108 or another system controlled or authorized by the validator system conducts a transaction on behalf of the user in the public DLS 32. The transaction may include one or more of: minting tokens in the public DLS 32 on behalf of the user, conducting a financial transaction in the public DLS 32 on behalf of the user, etc. By contrast, if at step 706, it is determined to deny authorization to the user to conduct transactions using the validator 108 or another system controlled or authorized by the validator system 108, the method proceeds to step 710. The method ends at step 710.

[0046] In embodiments, a method of using a personal DLS 24 to confirm to a validator system 108 the record of public DLS transactions stored in the personal DLS 24 may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to Fig. 7.

[0047] FIG. 8 depicts an embodiment 20D of a specific implementation of the system for providing a personal DLS 24 depicted in FIG. 7, including a personal DLS 24, host system 104, validator system 108, client system 120, DAPP operator system 124, and public DLS 32. The host system 104 hosts a master node 44A of the personal DLS 24 and a front end server 112 of the wallet DAPP 30. The host system 104 may be provided on behalf of the user by a service provider. The validator system 108 runs a validator node 44B of the personal DLS 24, the validator DAPP front end 116, a minting module 128, a minting DAPP front end 132, and a node of the public DLS 44C. The validator system 108 may be provided by a financial institution, such as a bank, investment bank, trading institution, etc. The DAPP operator system 124 runs a front end server 136 for a DAPP 36 based in the public DLS 32, and a node 44D of the public DLS. The DAPP operator system 124 may be provided by the operator of the DAPP 36. The client system 120 runs a front end client 140 of the wallet DAPP 30 and a front end client 144 of a public DLS DAPP 36. The client system 120 may be a system of the user.

[0048] In the embodiment of FIG. 8, the host and validator systems 104, 108 operate as discussed above in regard to FIGS. 6 and 7 to use the personal DLS 24 to confirm the record of public DLS transactions stored in the personal DLS 24. The validator system 108 determines whether to authorize, based on review of the transaction record in the personal DLS 24, the minting module 128 to respond to requests by the user, via the wallet DAPP 30, to utilize the minting DAPP 132 to mint tokens for the user in the public DLS 32 in exchange for user funds. Upon authorization, the user engages in transactions with the DAPP 36 in the public DLS 32, via the DAPP front end 136, using the minted tokens now attached to a user account in the public DLS 32.

[0049] In embodiments, the system for providing a personal distributed ledger may include only any subset of, or an alternative connection of, the features depicted in or discussed herein in regard to Fig. 8.

[0050] The personal DLS 24 also may be utilized to provide personal data archiving and associated functionalities for the user. FIG. 9 depicts an embodiment 20E of an expansion of

the system to also provide immutable, secure and reliable personal data archiving and retrieval functions. In the depicted embodiment, the system includes the personal DLS 24, one or more personal DLS DAPPs 28, and one or components interfacing with the personal DLS DAPPS 28. For purposes of clarity of illustration, public DLSs 32 are omitted from FIG. 8, although the one or components interfacing with the personal DLS DAPPS 28 may be based in such public DLSs 32.

[0051] The personal DLS DAPPs 28 include one or more of the wallet DAPP 30, an asset DAPP 148, an access DAPP 152, and identity DAPP 150 and a custom DAPP 160. Each of the DAPPs 28 is configured to provide associated functions for personal data archiving and retrieval, and includes a front end component and a DLS component. The wallet DAPP 30 has already been discussed above.

[0052] The asset and access DAPPs 148, 152 may provide an immutable, secure and reliable way to store, and ultimately distribute, digital assets of the user. The asset DAPP 148 is configured to receive and process digital assets, such as images, audio, video, and other data types, and store the assets or a representation of the assets in the personal DLS 24. The asset DAPP 148 includes an asset DAPP front end 164, the personal DLS DAPP contract 84B, an asset storage data structure 168 and an asset storage 172. The asset DAPP front end 164 is configured to provide a user interface for the user to load digital assets to the asset DAPP 148 from a user system 120. The personal DLS DAPP contract 84B is the smart contract stored in the personal DLS 24, and is configured to be invoked by a transaction containing the uploaded assets or representation and store the contained assets or representation in a corresponding asset storage data structure 168 in the personal DLS 24. The asset storage data structure 168 is a data structure in the personal DLS 24 associated with the personal DLS DAPP contract 84B to store the asset or representation stored by the personal DLS DAPP contract 84B. The asset storage 172 is a secure storage outside of the personal DLS 24 to store the uploaded asset when the asset is too large to itself be stored in the personal DLS 24.

[0053] Note that, in embodiments, each of the personal DAPPs 28 may include their own distinct front end and DLS components. In other embodiments, at least some of the personal DAPPS 28, or even all of the personal DAPPs 28, may share either or both of front end and DLS components. For example, a plurality of different ones of the personal DAPPs 28 may each have their own distinct DAPP front end, which may enable the front end to be adapted to the specific user interaction that that DAPP entails, such as specific types of data

collection. In another example, a plurality of different ones of the personal DAPPs may share a common personal DAPP contract 84B, which implements the personal DAPP contract functions for each of those personal DAPPs.

[0054] FIG. 10 depicts an embodiment of a method 1000 of using the personal DLS 24 to store digital assets. The method begins at step 1002. At step 1004, the digital asset is received by the asset DAPP front end 164 from the user via the user system 120. At step 1006, it is determined whether the digital asset has a storage size above a predetermined storage size. For digital assets below the predetermined size, the method stores the digital asset itself in the personal DLS 24, whereas for digital assets above the predetermined size, the method stores a representation of the digital asset in the personal DLS 24 and stores the asset itself in the secure storage 172. Thus, if at step 1006 it is determined that the digital asset size is below the predetermined storage size, the method proceeds to step 1008. At step 1008, the asset DAPP front end 164 generates a transaction to the personal DLS DAPP contract 84B, containing the digital asset, to store the digital asset in the asset storage data structure 168. At step 1016, the asset DAPP front end 164 transmits the transaction to at least one distributed node of the personal DLS 24. As part of the ledger structure creation process in the personal DLS 24, the transaction will be incorporated into a new ledger structure and executed by the personal DLS DAPP contract 84B. Executing the transaction by the personal DLS DAPP contract 84B invokes an asset storage function of the personal DLS DAPP contract 84B, which stores the asset contained in the transaction in the asset storage data structure 168.

[0055] By contrast, if at step 1006 it is determined that the digital asset size is above the predetermined storage size, the method proceeds to step 1010. At step 1010, the asset DAPP front end 164 generates a representation of the asset have a reduced size in comparison to the asset. In embodiments, generating the representation includes performing at least one cryptographic hash function on the asset, with the representation of the asset being the output of at least one cryptographic hash function. At step 1012, the asset DAPP front end 164 then stores the digital asset in the secure asset storage 172 outside of the personal DLS 24. At step 1014, the asset DAPP front end 164 generates a transaction to the personal DLS DAPP contract 84B, containing the representation of the digital asset, to store the representation of the digital asset in the asset storage data structure 168. At step 1016, the asset DAPP front end 164 transmits the transaction to at least one distributed node of the personal DLS 24. As part of the ledger structure creation process in the personal DLS 24, the transaction will be

incorporated into a new ledger structure and executed by the personal DLS DAPP contract 84B, and executing the transaction by the personal DLS DAPP contract 84B invokes the asset storage function of the personal DLS DAPP contract 84B, which stores the representation contained in the transaction in the asset storage data structure 168. The method ends at step 1018.

[0056] In embodiments, a method of using a personal DLS 24 to store digital assets may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to Fig. 10.

[0057] The user may distribute the digital assets stored using the personal DLS 24 to authorized entities, such as an entity entering into a license agreement for the digital asset. The access DAPP 152 is configured to enable selective access to digital assets stored using the personal DLS 24. The access DAPP 152 includes an access DAPP front end 176, the personal DLS DAPP contract 84B, the asset data structure 168 and the asset storage 172. The access DAPP front end 176 is configured to provide a user interface to the entity to license, request and/or receive the digital assets. The personal DLS DAPP contract 84B is the smart contract stored in the personal DLS 24, and is configured to be invoked by a read call to selectively retrieve the assets or asset representations stored in the corresponding asset storage data structure 168 in the personal DLS 24.

[0058] FIG. 11 depicts a method of distributing a digital asset stored using the personal DLS 24 to an authorized entity. The method begins at step 1102. At step 1104, the access DAPP front end 176 receives a request for a digital asset from an authorized requestor. At step 1106, the access DAPP front end 176 determines if the asset itself or a representation of the asset is stored in the personal DLS 24. For example, the access DAPP 152 may consult a log of assets stored using the personal DLS 24 to make the determination. If at step 1106 it is determined that the requested asset itself is stored in the personal DLS 24, the method proceeds to step 1108. At step 1108, the access DAPP front end 176 performs a read call to the personal DLS DAPP contract 84B to invoke an asset read function of the contract, which reads an indicated asset from the asset storage data structure 168 and outputs the retrieved asset. At step 1118, the retrieved asset is provided to the requesting authorized entity via the access DAPP front end 176. As part of the providing, the access DAPP front end 176 may optionally encrypt the retrieved asset, such as by using a public key of the requesting entity.

[0059] By contrast, if at step 1106 it is determined that the representation of the requested asset is stored in the personal DLS 24, and the asset itself is stored in the secure asset store 172, the method proceeds to step 1110. At step 1110, the access DAPP front end 176 performs a read call to the personal DLS DAPP contract 84B to invoke the asset read function of the contract, which reads an indicated asset representation from the asset storage data structure 168 and outputs the retrieved asset representation. At step 1112, the access DAPP front end 176 retrieves the asset itself from the secure asset store 172. At step 1114, the asset DAPP front end 176 generates a representation of the asset from the retrieved asset. In embodiments, generating the representation includes performing at least one cryptographic hash function on the asset, with the representation of the asset being the output of at least one cryptographic hash function. At step 1116, it is determined whether the representation retrieved from the personal DLS 24 matches the representation determined at step 1114, such as by comparing the representations. The determination may be performed in order to ensure that the asset retrieved from the secure asset store 172 has not changed, as will generally be discernable by comparison to the representation stored in the personal DLS 24, which has a relatively higher degree of immutability, security and reliability. If at step 1116, it is determined that the representations match, the method proceeds to step 1118, where the asset is provided to the requesting authorized entity, as discussed above. If at step 1116, it is determined that the representations do not match, the asset retrieved from the asset store 172 is not provided to the requesting entity, as the asset in the secure asset store 172 may have become corrupted, and the method instead proceeds to step 1120. At step 1120, the method ends.

[0060] In embodiments, a method of using a personal DLS 24 to distribute digital assets may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to Fig. 11.

[0061] The personal DLS 24 also may provide an immutable, secure and reliable way to store, and ultimately distributed, identity data of the user. The identity DAPP 156 is configured to receive and process identity data, such as identities (e.g., passports, driver's licenses, etc.), personal identifiers (social security numbers, etc.), biometric information (fingerprints, iris scans, facial patterns, DNA patterns, etc.), etc., and store the identity data or an encoding of the identity data in the personal DLS. The identity DAPP 156 includes an identity DAPP front end 180, the personal DLS DAPP contract 84B, an identity data storage data structure 184 and the asset storage 172. The identity DAPP front end 180 is configured

to provide a user interface for the user to load identity data from a user system 120. The personal DLS DAPP contract 84B is the smart contract stored in the personal DLS 24, and is configured to be invoked by a transaction containing the uploaded identity data or representation of the identity data and store the contained identity data or representation in a corresponding identity data storage data structure 184 in the personal DLS 24. The identity data storage data structure 184 is a data structure in the personal DLS 24 associated with the personal DLS DAPP contract 84B to store the identity data or representation stored by the personal DLS DAPP contract 84B. The asset storage 172 is the secure storage outside of the personal DLS 24 to store the uploaded identity when the identity data is too large to itself be stored in the personal DLS 24.

[0062] The identity data may be stored in the personal DLS 24 using the identity data according to the method of FIG. 10, adapted to receive, process and store the identity data instead of digital assets. Likewise, the identity data may be distributed to authorized requesting entities using the access DAPP and method of FIG. 11, adapted to retrieve and distributed the identity data instead of digital assets.

[0063] The personal DLS also may provide an immutable, secure and reliable way to store, and ultimately distributed, custom data related to third parties. The custom DAPP 160 may be configured by the user in conjunction with third parties to provide a variety of different data storage and distribution scenarios. The custom DAPP includes a custom DAPP front end 188, the personal DLS DAPP contract 84B, a custom data storage data structure 192 and the asset store 172. The custom DAPP 160 may be configured to function, such as by itself or in conjunction with the access DAPP 152, much like the asset, identity and access DAPPs and the methods of FIGS. 10 and 11, to receive, process and store custom data in the personal DLS 24, and ultimately distribute the custom data to the third parties or other entities, as agreed between the user and the third party.

[0064] In embodiments, the system for providing a personal distributed ledger may include only any subset of, or an alternative connection of, the features depicted in or discussed herein in regard to FIG. 9.

[0065] Components of the embodiments of the system for providing a personal distributed ledger, such as nodes of the personal DLS, components of the personal DLS DAPPs, nodes of the public DLS, components of the public DLS DAPPs, the validator system, etc., and/or any individual one, subset, or all of the components thereof, may be

implemented as hardware, software, or a mixture of hardware and software. For example, each of the nodes of the personal DLS, components of the personal DLS DAPPs, nodes of the public DLS, components of the public DLS DAPPs, the validator system, etc., and/or any individual one, subset, or all of the components thereof, may be implemented using a processor and a non-transitory storage medium, where the non-transitory machine-readable storage medium includes program instructions that when executed by the processor perform embodiments of the functions of such components discussed herein.

[0066] In embodiments, each of the nodes of the personal DLS, components of the personal DLS DAPPs, nodes of the public DLS, components of the public DLS DAPPs, the validator system, etc., and/or any individual one, subset, or all of the components thereof, may be implemented using a computer system, such as, e.g., a desktop computer, laptop computer, mobile computing device, network device, server, Internet server, cloud server, etc.

[0067] Additional embodiments of the personal DLS, personal DLS DAPP(s), public DLS, public DLS DAPP(s) and associated methods 500, 700, 1000, 1100 discussed herein are possible. For example, any feature of any of the embodiments of these systems and methods described herein may be used in any other embodiment of these systems and methods. Also, embodiments of these systems and methods may include only any subset of the components or features of these systems and methods discussed herein.

CLAIMS

1. A method of conducting transactions, the method comprising:

generating, by a wallet distributed application, an address in a public distributed ledger system and a private key;

signing, by the wallet distributed application, a first transaction to a distributed application in the public distributed ledger system from the generated address using the private key;

generating, by the wallet distributed application, a second transaction to a personal distributed ledger system, the generated second transaction containing information identifying the first transaction to the public distributed ledger system; and

transmitting, by the wallet distributed application, the second transaction to at least one distributed node of the personal distributed ledger system.

2. The method of claim 1, wherein the information identifying the first transaction to the public distributed ledger system includes the private key used to sign the first transaction.

3. The method of claim 1, wherein the information identifying the first transaction to the public distributed ledger system includes an execution receipt for the first transaction.

4. The method of claim 1, wherein the information identifying the first transaction to the public distributed ledger system includes at least a portion of the first transaction.

5. The method of claim 1, further comprising:

receiving, by the wallet distributed application, an invocation by the distributed application of the public distributed ledger system to sign the first transaction to the distributed application, the invocation including the first transaction; and

transmitting, by the wallet distributed application, the first transaction to at least one distributed node of the public distributed ledger system.

6. The method of claim 1, wherein the personal distributed ledger system is separate from the public distributed ledger system, and provides access to only the user and entities authorized by the user

7. The method of claim 1, further comprising:

reading, by a distributed application of a financial institution system hosting a node of the personal distributed ledger system, at least a portion of the information identifying the first transaction stored in the personal distributed ledger system;

determining whether to authorize, by the financial institution system, a user associated with the personal distributed ledger system to conduct transactions in the public distributed ledger system using the financial institution system based on the retrieved information identifying the first transaction;

upon authorization of the user, generating, by the financial institution system, a transaction to a distributed application of the public distributed ledger system on behalf of the user, wherein the transaction by the financial institution system mints tokens in the public distributed ledger system on behalf of the user.

8. The method of claim 7, further comprising:

generating, by the user, a transaction to a distributed application of the public distributed ledger system utilizing the minted tokens in the public distributed ledger system

9. The method of claim 1, further comprising:

transmitting the first transaction to the public distributed ledger system for execution;

wherein the information identifying the first transaction includes at least a portion of the executed first transaction and the private key used to sign the first transaction, and

the personal distributed ledger system is separate from the public distributed ledger system and includes only distributed nodes authorized by the user, the authorized distributed nodes including a validator node hosted by a financial institution system.

10. The method of claim 9, further comprising:

executing, by the validator node hosted by the financial institution system, a read of the personal distributed ledger system to retrieve the identifying information; and

generating, by the at least one financial institution system based on the retrieved identifying information, tokens of the public distributed ledger system for use by the user to conduct further transactions in the public distributed ledger system.

11. The method of claim 1, further comprising:

receiving, via a distributed application of the personal distributed ledger system, a digital object;

determining whether a storage size of the digital object is above a predetermined threshold;

upon determining that the storage size of the digital object is below the predetermined threshold: generating a transaction to store the digital object in the personal distributed ledger system, and transmitting the transaction to at least one distributed node of the personal distributed ledger system;

upon determining that the storage size of the digital object is above the predetermined threshold: determining a representation of the digital object, storing the digital object in a secure storage system outside of the personal distributed ledger system, generating a transaction to store the representation of the digital object in the personal distributed ledger system, and transmitting the transaction to at least one distributed node of the personal distributed ledger system.

12. The method of claim 11, wherein the digital object is identity data, the identity data including at least one of: driver's license data, passport data, a social security number, a fingerprint, an iris pattern, a facial pattern, or a DNA pattern.

13. The method of claim 11, wherein determining the representation includes performing at least one cryptographic hash on the digital asset.

14. The method of claim 11, further comprising:

receiving, via an access distributed application of the personal distributed ledger system, a request for the digital object;

determining whether the digital object or the representation of the digital object is stored in personal distributed ledger system;

upon determining that the digital object is stored in the personal distributed ledger system: reading via the access distributed application the personal distributed ledger system to retrieve the digital object, and providing the digital object to a requesting entity; and

upon determining that the representation of the digital object is stored in the personal distributed ledger system: reading via the access distributed application the personal distributed ledger system to retrieve the representation of the digital object, retrieving the digital object from an asset store, determining a representation of the retrieved digital object, determining whether the determined representation matches the retrieved representation, and upon determining that the determined representation matches the retrieved representation, providing the digital object to the requesting entity.

15. A system for conducting transactions, the system comprising:

at least one processor; and

a non-transitory machine-readable storage medium having program instructions, which when executed by a processor perform a method of conducting transactions, the method comprising:

generating, by a wallet distributed application, an address in a public distributed ledger system and a private key;

signing, by the wallet distributed application, a first transaction to a distributed application in the public distributed ledger system from the generated address and using the private key;

generating, by the wallet distributed application, a second transaction to a personal distributed ledger system, the generated second transaction containing information identifying the first transaction to the public distributed ledger system; and

transmitting, by the wallet distributed application, the second transaction to at least one distributed node of the personal distributed ledger system.

16. The system of claim 15, wherein the information identifying the first transaction to the public distributed ledger system includes the private key used to sign the transaction.

17. The system of claim 15, wherein the personal distributed ledger system is separate from the public distributed ledger system, and provides access to only the user and entities authorized by the user.

18. A method of authorizing a transaction in a public distributed ledger system using a personal distributed ledger system, the method comprising:

reading, by a distributed application of a financial institution system hosting a node of the personal distributed ledger system, information identifying at least one first transaction conducted in the public distributed ledger system stored in the personal distributed ledger system;

determining, by the financial institution system, whether to authorize a user associated with the personal distributed ledger system to conduct further transactions in the public distributed ledger system using the financial institution system based on the retrieved information identifying the transaction; and

upon authorization of the user, generating, by the financial institution system, a transaction to a distributed application of the public distributed ledger system on behalf of the user.

19. The method of claim 18, wherein the reading executes by the distributed application of the financial institution system a read call to a contract of the personal distributed ledger system to retrieve the information identifying the first transaction, the identifying information including at least a portion of the conducted first transaction and a private key used to sign the first transaction, the personal distributed ledger system being separate from the public distributed ledger system and including only distributed nodes authorized by the user, the authorized distributed nodes including the validator node hosted by the financial institution system; and

wherein the transaction generated by the financial institution system is executed to generate tokens of the public distributed ledger system for use by the user to conduct the further transactions in the public distributed ledger system

20. The method of claim 18, further comprising generating, by the user, a transaction to a distributed application of the public distributed ledger system utilizing an output of the transaction by the financial institution,

wherein the transaction by the financial institution system mints tokens in the public distributed ledger system on behalf of the user, and

the transaction by the user utilizes the minted tokens in the public distributed ledger system.