

SYSTEMS AND METHODS FOR PROVIDING IDENTITY ASSURANCE FOR DECENTRALIZED APPLICATIONS

CROSS REFERENCE TO RELATED APPLICATIONS

[01] This application claims priority to U.S. Provisional Patent Application No.
5 62/384,980, filed on September 8, 2016, and U.S. Patent Application No. 15/696,996, filed
on September 6, 2017, each of which are hereby incorporated by reference herein in their
entireties.

BACKGROUND INFORMATION

[02] Decentralized applications, in which a client component interfaces with a
10 distributed component stored in a blockchain system, are an increasingly popular form of
application due to benefits provided by the underlying architecture, such as freedom from the
necessity of a centralized clearing authority, increased resistance to subversive attacks, etc.
In some of these applications, the distributed component includes a set of program
instructions, also referred to as a contract or smart contract, stored in the blockchain system,
15 which may be executed in response to a transaction addressed to the contract.

[03] However, implementations of such applications to date have suffered from
problems. Typically, the distributed component, such as the smart contract, accepts
transactions from an address, with the user behind the address being unknown to the
component. This presents potential security and efficiency risks, as a subset of these
20 transactions may be of a malicious or otherwise undesirable form, which may seek to use the
decentralized application in an unauthorized manner, such as to steal assets stored in the
blockchain system, attack other applications or components of the blockchain system, disrupt
operation of the decentralized application or other components of the blockchain system, etc.

[04] Thus, a need exists for systems and methods to enable decentralized application to
25 execute only authorized transactions.

SUMMARY

[04a] According to a first aspect of the present invention, there is provided a method of
providing identity assurance for a decentralized application (DApp), the method comprising:
executing, by at least one distributed node of a blockchain system, program instructions of an
30 entitlement contract, the entitlement contract including the program instructions being stored
on a blockchain of the blockchain system, to perform a read call received by the entitlement
contract from a DApp contract stored on the blockchain, the read call including

an address signing a transaction to the DApp contract, performing the read call including: reading a list of registered addresses stored on the blockchain by the entitlement contract; determining whether the list of registered addresses includes the signing address included with the read call; and providing an output indicating whether the list of registered addresses includes the signing address included with the read call.

[04b] According to a second aspect of the present invention, there is provided a method of providing identity assurance for a decentralized application (DApp), the method comprising: receiving, by at least one distributed node of a blockchain system, a transaction addressed to a contract of the DApp stored on a blockchain in the blockchain system, the transaction including an address signing the transaction; and executing, by the at least one distributed node, the DApp contract to conduct the transaction, the executing including: if the transaction invokes a predetermined blockchain modification function of the DApp contract, executing an entitlement checking function of the DApp contract including a read call to an entitlement contract stored on the blockchain, the read call containing the signing address; receiving an output from the read call to the entitlement contract indicating whether the signing address is authorized to execute the predetermined blockchain modification function; and if the signing address is authorized to execute the predetermined blockchain modification function, executing the predetermined blockchain modification function.

[04c] According to a third aspect of the present invention, there is provided a system for providing identity assurance for a decentralized application (DApp), the system comprising: at least one processor; and at least one non-transitory machine-readable storage medium including program instructions, which when executed by the at least one processor cause the at least one processor to perform a method of providing the identity assurance for the DApp, the method including: receiving, by at least one distributed node of a blockchain system, a transaction addressed to a contract of the DApp stored on a blockchain in the blockchain system, the transaction including an address signing the transaction; and executing, by the at least one distributed node, the DApp contract to conduct the transaction, the executing including: if the transaction invokes a predetermined blockchain modification function of the DApp contract, executing an entitlement checking function of the DApp contract including a read call to an entitlement contract stored on the blockchain, the read call containing the signing address; receiving an output from the read call to the entitlement contract indicating whether the signing address is authorized to execute the predetermined blockchain

modification function; and if the signing address is authorized to execute the predetermined blockchain modification function, executing the predetermined blockchain modification function.

5 **BRIEF DESCRIPTION OF THE DRAWINGS**

[05] 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.

10 [06] FIG. 1 is a schematic diagram depicting an embodiment of a system for conducting authorized transactions by a decentralized application on a blockchain.

[07] FIG. 2 is a schematic diagram depicting an embodiment of the system for conducting authorized transactions, showing in greater detail embodiments of a decentralized application and an identity assurance system.

[08] FIG. 3 is a schematic diagram depicting an embodiment of a blockchain system.

5 [09] FIG. 4 is a schematic diagram depicting an embodiment of a distributed node of the blockchain system.

[10] FIG. 5 is a flow diagram depicting an embodiment of transaction and entitlement checking functions of the decentralized application and identity assurance system.

10 [11] FIG. 6 is a schematic diagram depicting an embodiment of a contract of the decentralized application.

[12] FIG. 7 is a schematic diagram depicting an embodiment of an entitlement contract of the identity assurance system.

[13] FIG. 8 is a schematic diagram depicting an embodiment of an registry contract of the identity assurance system.

15 [14] FIG. 9 is a flow diagram depicting an embodiment of transaction, registry checking, and entitlement checking functions of the decentralized application and identity assurance system.

[15] FIG. 10 is a flow diagram depicting an embodiment of a registration of the decentralized application with the identity assurance system.

20 [16] FIG. 11 is a flow diagram depicting an embodiment of a registration of a user with the identity assurance system.

[17] FIG. 12 is a schematic diagram depicting an embodiment of a server of the identity assurance system.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

25 [18] Embodiments of an identity assurance system provide improved identity services for decentralized applications operating in the context of a blockchain system.

[19] The identity assurance system may include one or more components in the blockchain system to enable the decentralized application to process only authorized transactions. An entitlement contract of the identity assurance system may provide an
30 indication to a contract of the decentralized application of whether an address signing a

transaction is authorized to conduct the transaction. To verify the address, the entitlement contract may store, maintain and consult a list of addresses authorized to conduct transactions with the decentralized application.

[20] A registry contract of the identity assurance system may provide an indication to the contract of the decentralized application of the location of the entitlement contract within the blockchain. To provide the location, the registry contract may store, maintain and consult a list of registered applications and corresponding entitlement contracts.

[21] The identity assurance system thus provides entitlement checking to improve the security and efficiency with which the decentralized application and the blockchain system operate by reducing their exposure to unauthorized transactions and thereby maintaining an operative status of the decentralized application and blockchain system and reserving their processing power for desired transactions. The entitlement checking functions provided by the identity assurance system may operate within the blockchain system itself, again improving security and efficiency of the system by reducing communications off chain and providing immutability to address and application registries. Similarly, the identity assurance system enables a user associated with an authorized address to execute restricted functions of the decentralized application without revealing the identity of the user to the decentralized application, thereby maintaining the privacy of the user while providing identity assurance to the decentralized application.

[22] The identity assurance system also may include a wallet interface to generate transactions for the contract of the decentralized application. The wallet interface may download a decryption module and one or more encrypted keys to a user system, decrypt the keys using the decryption module, and sign transactions using the decrypted keys. The wallet interface thus provides improved security to the user, decentralized application and blockchain system by limiting exposure of the unencrypted keys to only the user's system.

[23] FIG. 1 depicts an exemplary embodiment of a system 20, for conducting transactions by a decentralized application, including a user system 24, the decentralized application 28, an identity assurance system 32, and a blockchain system 36. A user of the user system 24 uses the decentralized application 28 to engage services that the application provides. The decentralized application 28 may provide a wide variety of such services, such as related to conducting financial transactions, providing business productivity tools, gaming, etc. The identity assurance system 32 enables the decentralized application 28 to execute

only authorized transactions, and thus provide increased security, efficiency, etc. for the decentralized application 28 and blockchain system 36. The blockchain system 36 implements a distributed transaction ledger utilized by components of the decentralized application 28 and identity assurance system 32.

5 [24] FIG. 2 depicts an exemplary embodiment of the system 20 for conducting authorized transactions showing embodiments of the decentralized application 28, identity assurance system 32, and blockchain system 36 in greater detail. The decentralized application (DApp) 28 includes client, distributed and server components. A user interface of the decentralized application (DApp user interface) 40 executes on the user, i.e., client,
10 system 24 to provide an interface to services of the decentralized application 28. A smart contract of the decentralized application (DApp contract) 44 is stored on a blockchain of the blockchain system 36. The DApp contract 44 includes program instructions to be executed by distributed nodes of the blockchain system 36 to conduct transactions received from the DApp user interface 40 to perform services of the decentralized application 28. A server of
15 the decentralized application (DApp server) 48 serves components to the user system for execution as the DApp user interface 40. For example, the DApp user interface 40 may execute in a browser program on the user system using Internet technologies such as HTML, Javascript, CSS, etc.

[25] The identity assurance system 32 also includes client, distributed and server
20 components. A wallet user interface 52 provides identity services to the DApp user interface for generating transactions to the decentralized application, such as key decryption, transaction signing, etc. for registered users. One or more registry and/or entitlement contracts of the identity assurance system 56, 60 are stored in the blockchain system 36. The
25 contracts 56, 60 includes program instructions to be executed by distributed nodes of the blockchain system 36 to enable the decentralized application 28 to execute only authorized transactions. An identity assurance server 64 provides encryption/decryption and key services to the wallet user interface 52 for generating the transactions, and contract services for creating and maintaining the contracts 56, 60 in the blockchain system 36.

[26] In other embodiments, the decentralized application 28, identity assurance system
30 32 and blockchain system 36 may include only any subset of the components or features depicted in or discussed in regard to Fig. 2 or otherwise discussed herein.

[27] FIG. 3 depicts an exemplary embodiment of the blockchain system 36 including a plurality of distributed nodes 68. The distributed nodes 68 are organized as a peer-to-peer network, in which each of the nodes 68 may connect to one or more of the other nodes 68 using a peer-to-peer communication protocol. At least one or some of the distributed nodes
5 68 also connect to components of the user system 24 and/or components of the identity assurance system 32.

[28] FIG. 4 depicts an exemplary embodiment of the distributed node 68 of the blockchain system, including a block storage module 72, a communication module 76, a virtual machine module 80 and a block creation module 84. The block storage module 72
10 stores blocks of the blockchain of the blockchain system 36. The communication module 76 performs communications between the distributed node 68 and other distributed nodes 68 and/or components of the user system 24 and/or the identity assurance system 32. The virtual machine module 80 executes contracts stored on the blockchain of the blockchain system 36. The block creation module 84 performs an algorithm to incorporate transactions into blocks
15 of the blockchain of the blockchain system 36, also referred to as mining blocks of the blockchain. In embodiments, one or more of the distributed nodes 68 of the blockchain system include only selected subsets of the components depicted in FIG. 4, so as to perform only corresponding subsets of the functions performed by the included modules.

[29] Components of the identity assurance system 32, decentralized application 28,
20 blockchain system 36 and user system 24 may be implemented as hardware, software, or a mixture of hardware and software. For example, each of the user system 24, identity assurance server 64, DApp server 48, and nodes 68 of the blockchain system 36, 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
25 storage medium includes program instructions, which when executed by the processor perform embodiments of the functions of such components discussed herein.

[30] In one embodiment, the user system 24 may be a computer system, such as, e.g., a desktop computer, laptop computer, mobile computing device, etc. owned or operated the user; the DApp server 48 and identity assurance server 64 may be computer systems, such as,
30 e.g., an Internet server, cloud server, etc., owned or operated by the DApp owner and identity assurance provider, respectively; the distributed nodes 68 of the blockchain system each may be a computing system, such as, e.g., a desktop computer, laptop computer, mobile

computing device, network device, server, cloud server, etc., owned or operated by the identity service provider or another entity or entities.

[31] The identity assurance system 32 provides identity services to the decentralized application 28 to enable the decentralized application 28 to conduct only authorized transactions. Embodiments of the identity assurance system 32 provide identity services in the context of both wallet functions to generate signed transactions and entitlement functions to register and verify authorized addresses and/or users.

[32] FIG. 5 is a flow diagram depicting an exemplary embodiment of identity services provided by the identity assurance system 32 for performing wallet and entitlement functions over time and sorted by component. Initially, the DApp user interface 40 receives an input from the user at the user system 24 requesting a service or function of the decentralized application 28 requiring a signed transaction be sent to the DApp contract 44. A wallet component embedded in the DApp user interface 40 intercepts an as yet unsigned transaction and launches the wallet user interface 52, which may execute in a separate window.

[33] If the user is already registered with the identity assurance system 36, the wallet user interface 52 requests and downloads a decryption module and one or more encrypted keys of the user from a decryption module and key storage components 88, 92 of the identity assurance server 64. If the user is not yet registered with the identity assurance system 32, the wallet user interface 52 first launches a process to register the user, as discussed further below. The wallet user interface 52 then requests that the user enter a password, and if the password is correctly entered, decrypts the encrypted key using the downloaded decryption module, signs the transaction using the decrypted key, and passes the signed transaction back to the DApp user interface 40.

[34] The DApp user interface 40 then transmits the signed transaction addressed to the DApp contract 44 to at least one distributed node 68 of the blockchain system 36. As part of a process of mining blocks of the blockchain system 36, at least one of the distributed nodes 68 executes the DApp contract 44 to process the transaction.

[35] If during execution of the DApp contract 44, the transaction invokes one or more predetermined restricted functions of the DApp contract 44 designated for access only by authorized addresses, the DApp contract 44 invokes an entitlement checking function of the DApp contract 44 to check whether the address signing the transaction is authorized to invoke the predetermined function. The predetermined functions may include functions of

the DApp contract 44 deemed to represent a security risk if executed by unauthorized addresses. For example, in embodiments, the predetermined functions include any functions of the DApp contract 44 that modify data stored by the DApp contract 44 on the blockchain.

[36] The entitlement checking function of the DApp contract 44 executes a read call of an entitlement contract 60 of the identity assurance system 36 to determine if the address signing the transaction is authorized to execute the invoked predetermined function of the DApp contract 44. The read call returns as an output an indication of whether the address is authorized to invoke the predetermined function. If the address is authorized to invoke the predetermined function, the DApp contract 44 continues execution to perform the invoked function.

[37] Execution by the DApp contract of the predetermined restricted function enables performance of selected activities by the decentralized application 28, in embodiments including one or more of: conducting a financial transaction in the name of the address signing the transaction (such as a financial transaction between the user using the address and a second user); transferring ownership of an asset (such as a digital or physical asset) in the name of the address signing the transaction (such as transferring ownership of the asset to or from the user using the address from or to the second user); presenting a pay per view or pay per play media environment (such as a video or game) to the user system; or causing an electronic device (such as a control circuit) to generate an electrical signal to control a mechanical operation of an apparatus (such as an air conditioner, fan, refrigerator, a combustion engine, etc.) or to control an electrical operation of an apparatus (such as one or more of another circuit, a computing device, an electric motor, a hybrid engine, etc.).

[38] The identity services provided by the identity assurance system 32 thus enable a user associated with an authorized address to execute restricted functions of the DApp contract 44 without revealing the identity of the user to the decentralized application 28, thereby maintaining the privacy of the user while providing identity assurance to the decentralized application 28. That is, as long as the user has registered with the identity services, e.g., as discussed further below in regard to the embodiment of FIG. 11, the decentralized application 28 receives identity assurance with respect to the user via the entitlement checking of the identity services, etc., while allowing the user's identity to remain concealed from the decentralized application 28.

[39] In other embodiments, the method(s) of providing identity services depicted in or discussed in regard to Fig. 5 may include only any subset of the steps of such method(s).

[40] FIG. 6 depicts an exemplary embodiment of the DApp contract 44 including one or more restricted functions 136, an entitlement checking function 140, and a registry checking function 144.

[41] As discussed above, the restricted functions 136 may be invoked by a transaction to the DApp contract 44 to invoke the functionality implemented by these functions. The restricted functions 136 are functions of the DApp contract 44 for which execution is restricted to only authorized addresses. The restricted functions 136 may be functions of the DApp contract 44 which represent the greatest risk by unauthorized execution. In embodiments, the restricted functions 136 include at least one function of the DApp contract 44 that modifies data stored by the contract 44 on the blockchain, such as each of the functions of the DApp contract 44 that modify such data.

[42] The entitlement checking function 140 executes a read call to the entitlement contract 60 to determine if the address signing the transaction is authorized to execute the restricted function 136. The entitlement checking function 140 receives the signing address as an input, provides the signing address to the entitlement contract 60 in the read call, and provides an indication of whether the address is authorized to invoke the restricted function 136 as an output.

[43] In embodiments, the identity assurance system 32 provides a separate corresponding entitlement contract 60 for each decentralized application 28. The registry checking function 140 of the DApp contract 44 executes a read call to a registry contract 56 of the identity assurance system 32 to determine an address of the entitlement contract 60 corresponding to the decentralized application 28. The registry checking function 144 receives an identifier identifying the decentralized application 28 as an input, provides the identifier to the registry contract 56 in the read call, and provides an indication of whether a corresponding entitlement contract 60 exists for the decentralized application 28 and, if so, the address of the corresponding entitlement contract 60 on the blockchain as an output.

[44] FIG. 7 depicts an exemplary embodiment of the entitlement contract 60 including an entitlement checking function 148, a registered address list 152, and an address registration function 156.

[45] The entitlement checking function 148 is a read call of the entitlement contract 60 that determines if an address included with the read call is in the list of registered addresses 152. The entitlement checking function 148 receives the address to be checked as an input, reads the list of registered addresses 152 stored by the entitlement contract 60, and provides
5 as an output an indication of whether the address is in the list of registered addresses 152. As a read call, the entitlement checking function 148 may be executed by directly executing the entitlement contract 60 by the distributed node 68 executing the DApp contract 44 without transmitting a transaction containing the read call to a distributed node 68 for execution.

[46] The registered address list 152 is a data structure stored by the entitlement
10 contract 60 on the blockchain containing a list of addresses authorized to execute the restricted functions 136 of the DApp contract 44. Functions of the entitlement contract 60, such as the entitlement checking function 148 and address registration function 156, selectively read or write to the registered address list 152, respectively. Alternatively or in addition to storing a list of authorized addresses, in embodiments, the list 152 may store an
15 indicator of a user generating the transaction other than the address signing the transaction, such as an identifier of the user, and the entitlement checking functions 144, 148 of the DApp contract 44 and entitlement contract 60 may operate on the basis of checking this indicator against the list of registered indicators.

[47] The address registration function 156 is invoked by a transaction addressed to the
20 entitlement contract 60 to add or remove registered addresses to or from the list of registered addresses 152. The address registration function 156 receives an address as an input, and adds or removes the address to or from the list of registered addresses 152 as a result. The address registration function 156 is a restricted function of the entitlement contract 60, accessible only to a wallet or address of the identity assurance system 32. In embodiments,
25 the entitlement contract 60 may include a single address registration function 156 to both add and remove addresses to and from the registered address list 152, or alternatively a plurality of address registration functions 156, such as a first address registration function to add addresses to the list of registered addresses 152, and a second address registration function to remove addresses from the list of registered addresses 152.

[48] In embodiments, the entitlement contract 60 may provide a more detailed
30 indication of whether an address is authorized to use a restricted function 136. For example, the entitlement contract 60 may store in the registered list 152 one or more of a role of a registered address or which selected ones of a plurality of different restricted functions the

address is authorized to invoke. In such embodiments, the entitlement checking functions 144, 148 may return an indication of the one or more of the role of the address or which functions the address is entitled to invoke.

[49] FIG. 8 depicts an exemplary embodiment of the registry contract 56 including a registry checking function 160, a registered application list 164, and an application registration function 168.

[50] The registry checking function 160 is a read call of the registry contract 56 to determine if an address included with the read call is in the list of registered applications 164. The registry checking function 160 receives an identifier of the decentralized application 28, such as an address of the application 28 on the blockchain, as an input, reads a list of registered applications 164 stored by the registry contract 56, and, if the identifier is contained in the list 164, provides as an output an address of an entitlement contract 60 stored in the list 164 corresponding to the application 28. Similar to as discussed above, as a read call, the registry checking function 160 may be executed by the distributed node 68 executing the DApp contract 44 by directly executing the registry contract 56 without transmitting a transaction containing the read call to a distributed node 68 for execution.

[51] The registered application list 164 is a data structure stored by the registry contract 56 on the blockchain containing a list of registered applications and corresponding entitlement contract addresses. Functions of the registry contract 56, such as the registry checking function 160 and application registration function 168, may read from or write to the registered application list 164, respectively.

[52] The application registration function 168 is invoked by a transaction addressed to the registry contract 56 to add or remove applications to or from the list of registered applications 164. The application registration function 168 receives an identifier of an application, and, in the case of adding an application, a corresponding entitlement contract address, as input, and adds or removes the application and corresponding entitlement contract address to or from the list of registered applications as a result. The application registration function 168 is a restricted function, accessible only to a wallet or address of the identity assurance system 32. In embodiments, the registry contract 56 may include a single application registration function 168 to both add and remove applications 28 to and from the registered application list 164, or alternatively a plurality of application registration functions 168, such as a first application registration function to add applications 28 to the list of

registered applications 164, and a second application registration function to remove applications 28 from the list of registered applications 164.

[53] In other embodiments, the DApp contract 44, entitlement contract 60, and registry contract 56 may include only any subset of the components or features depicted in or discussed in regard to Figs. 6, 7 and 8.

[54] FIG. 9 is a flowchart showing an exemplary embodiment of providing identity assurance services to the decentralized application 28, showing functions of the DApp contract 44, entitlement contract 60 and registry contract 56 in greater detail.

[55] The DApp user interface 40 transmits to at least one distribute node 68 of the blockchain system 36 a transaction addressed to the DApp contract 44 invoking one of the restricted functions 136 of the DApp contract 44. As part of the creation of new blocks of the blockchain, the DApp contract 44 is executed by at least one distributed node 68 to process the transaction.

[56] Invoking the restricted function 136 of the DApp contract 44 by the transaction causes the DApp contract 44 to invoke the registry checking function 140 of the DApp contract 44. As discussed above, invoking the registry checking function 140 executes a read call to the registry contract 56 including an identifier of the DApp contract 44. The read call to the registry contract 56 invokes the registry checking function 160 of the registry contract 56, which reads the registered application list 164 of the registry contract 56, and, if the indicated application 28 is contained in the registered application list 164, returns an address of the entitlement contract 60 corresponding to the decentralized application 28, or else returns an indication that the application 28 is not registered.

[57] The DApp contract 44 then invokes the entitlement checking function 144 of the DApp contract 44, with the returned address of the entitlement contract 60 and the signing address of the transaction invoking the restricted function 136 as inputs. As discussed above, invoking the entitlement checking function 144 executes a read call to the entitlement contract 60 including the signing address. The read call invokes the entitlement checking function 148 of the entitlement contract 60, which reads the registered address list 152 of the entitlement contract, and, if the signing address is contained in the registered address list 152, returns an indication that the signing address is authorized to invoke the restricted function 136, or else returns an indication that the signing address is not authorized to invoke the restricted function 136.

[58] If the entitlement checking function 144 returns that the signing address is authorized to invoke the restricted function 136, the DApp contract 44 then proceeds to execute the restricted function 136 to process the transaction. By contrast, if the entitlement checking function 144 returns that the signing address is not authorized to invoke the restricted function 136, the DApp contract 44 aborts processing the transaction.

[59] In other embodiments, the method(s) of providing identity services depicted in or discussed in regard to Fig. 10 may include only any subset of the steps of such method(s).

[60] The identity assurance system 32 may provide an interface for registration of the decentralized application 28 for the identity services discussed herein. FIG. 10 is a flowchart depicting an exemplary embodiment of registration of the decentralized application over time and sorted by component.

[61] An owner of the decentralized application 28 accesses an application registration user interface 172 using an owner system. The application registration user interface 172 may be served to the owner system by the identity assurance server 64. Through the application registration user interface 172, the owner of the decentralized application 28 provides an identification of the decentralized application 28, such as a name of the application 28 and/or an address of the DApp contract 44 of the application 28 on the blockchain, and terms of use for the user of the application 28, to an application registration module 176 of the identity assurance system 32, such as executing on the identity assurance server 64.

[62] In response, the application registration module 176 creates an entitlement contract 60 corresponding to the decentralized application 28, and generates and transmits a transaction to the blockchain system deploying the generated entitlement contract 60 to the blockchain. The registration module 176 also generates and transmits to the blockchain system a transaction addressed to the registry contract 56 to write the identifier of the decentralized application 28 and the address of the corresponding entitlement contract 60 to the registered application list 164 of the registry contract 56.

[63] The registration component 176 then returns a confirmation of registration and an identity assurance module to the decentralized application owner via the registration user interface 172. The identity assurance module includes program instructions to implement aspects of the identity assurance system discussed herein within the DApp user interface 40 and DApp contract 44, such as program instructions to implement triggering the wallet user

interface 52, registry checking function 140, and entitlement checking function 144. The decentralized application owner may incorporate the program instructions from the identity assurance module into the DApp user interface 40 and DApp contract 44.

[64] In embodiments, the identity assurance system 32 may optionally omit the registry contract 56 and the associated functionality discussed above and include only one or more entitlement contracts 60 for a decentralized application 28. In such embodiments, the DApp contract 44 may itself contain an address for the entitlement contract 60 in program instructions of the DApp contract 44, and thus be enabled to address the corresponding entitlement contract 60 without first consulting a registry contract.

[65] In different embodiments, a single decentralized application may consult a corresponding single entitlement contract, a plurality of decentralized applications may consult a single entitlement contract, or a single decentralized application may consult a plurality of entitlement contracts.

[66] The identity assurance system 32 may provide an interface for registration of a user for the identity services discussed herein. FIG. 11 is a flowchart depicting an exemplary embodiment of providing registration of the user over time and sorted by component.

[67] When the DApp user interface 40 triggers the launching of the wallet user interface 52 to sign a transaction, as discussed above, the wallet user interface 52 determines if the user is registered with the identity assurance system 32. If the user is not yet registered, the wallet user interface 52, or another interface launched by the wallet user interface 52, performs a registration of the user.

[68] To register the user within the identity assurance system 32, the wallet interface 52 or other interface requests and transmits identification information from the user to a user registration component 180 of the identity assurance system, such as executed by the identity assurance server 64. The identification information may include an identifier of an identity provided by an identify provider. The user registration component 180 authenticates or trigger authentication of the user information. The authentication of the user information may be performed either by the user registration component 180 or by an identity provider system, such as a government, corporate or other identity provider system, which issued the identity provided by the user which is being authenticated. Upon authentication of the user information, the user registration component 180 registers the user and transmits a confirmation back to the user via the interface 52.

[69] The user also may request to be registered as an authorized user of the restricted functions 136 of the decentralized application 28. To register the user as an authorized user, the user is presented with the terms of use of the decentralized application 28 via the interface 52 from the user registration component 180. In embodiments, other criteria also may be presented for satisfaction by the user for registering with the decentralized application 28. Upon agreeing to the terms of use of the decentralized application 28 and satisfying any other criteria, the user registration component 180 registers the user as an authorized user for the decentralized application 28. To register the user, the user registration component 180 generates and transmits to at least one of the distributed nodes 68 of the blockchain system 36 a transaction to invoke the address registration function 156 of the entitlement contract 60 for the decentralized application 28 to add one or more addresses or other indicators of the user to the registered address list 152.

[70] In other embodiments, the method(s) of registering depicted in or discussed in regard to Figs. 10 and 11 may include only any subset of the steps of such method(s).

[71] FIG. 12 depicts an exemplary embodiment of the identity assurance server 64 including the wallet module 184, the application registration module 176, and the user registration module 180.

[72] The wallet module 184 includes the decryption module 88 and one or more encrypted keys 92. In embodiments, the encrypted keys 92 may be separately stored for use with different decentralized applications 28.

[73] In other embodiments, the identity assurance server may include only any subset of the components or features depicted in or discussed in regard to Fig. 12.

[74] The blockchain stored by the blockchain system includes an ordered plurality of blocks, starting from a genesis block to a most recently created block. Each block may include a header and a list of transactions incorporated into that block. For operation of the identity assurance system, the transactions may include, among others, a transaction deploying the DApp contract, a transaction deploying the registry contract, a transaction deploying the entitlement contract, one or more transactions addressed to execute restricted functions of the DApp contract, one or more transactions addressed to execute application registration functions of the registry contract, one or more transactions addressed to execute user registration functions of the entitlement contract, etc.

[75] In embodiments, a web token system is used to protect access to components of the system for conducting transactions, such as the identity assurance sever and ports to the blockchain system. The web token system issues a time- and identity-limited web token to a user to access the protected components. Once the token has expired, a new token must be issued.

[76] In embodiments, the system for conducting transactions 20 may include a different type of distributed transaction ledger system in place of the blockchain system 36, such as a distributed transaction ledger system having a data organization other than a blockchain, and embodiments of the associated methods of providing identity assurance discussed herein may use such a distributed transaction ledger system in place of the blockchain system 36. Such a distributed transaction ledger system may have a distributed structure similar to as depicted in FIG. 3, including a plurality of distributed nodes that may be organized as a peer-to-peer network. The distributed nodes of such a distributed transaction ledger system may include modules similar to as depicted in FIG. 4, such as a data storage module to store the data structures of the distributed transaction ledger system, a communication module, a virtual machine module, and a data creation module to create the data structures.

[77] Additional embodiments of the decentralized application, identity assurance system, blockchain system, user system and associated methods of providing identity assurance discussed herein are possible. For example, any feature of any of the embodiments of the decentralized application, identity assurance system, blockchain system, user system and associated methods of providing identity assurance described herein may be used in any other embodiment of the decentralized application, identity assurance system, blockchain system, user system or associated methods of providing identity assurance. Also, embodiments of the decentralized application, identity assurance system, blockchain system, user system and associated methods of providing identity assurance may include only any subset of the components or features of the decentralized application, identity assurance system, blockchain system, user system and associated methods of providing identity assurance discussed herein.

WHAT IS CLAIMED IS:

1. A method of providing identity assurance for a decentralized application (DApp), the method comprising:

executing, by at least one distributed node of a blockchain system, program

5 instructions of an entitlement contract, the entitlement contract including the program instructions being stored on a blockchain of the blockchain system, to perform a read call received by the entitlement contract from a DApp contract stored on the blockchain, the read call including an address signing a transaction to the DApp contract, performing the read call including:

10 reading a list of registered addresses stored on the blockchain by the entitlement contract;

determining whether the list of registered addresses includes the signing address included with the read call; and

15 providing an output indicating whether the list of registered addresses includes the signing address included with the read call.

2. The method of claim 1, further comprising:

executing, by the at least one distributed node, a registry contract stored on the blockchain to perform a read call to the registry contract from the DApp contract, the read call including an identifier of the decentralized application, performing the read call

20 including:

reading a list of registered applications stored on the blockchain by the registry contract;

determining whether the list of applications includes the identifier included with the read call; and

25 providing an output indicating whether the list of registered applications includes the identifier included with the read call, and, if list of registered applications including the identifier, an address stored in the list of registered applications corresponding to the entitlement contract.

3. The method of claim 1, wherein an identity of a user corresponding to the address
30 signing the transaction to the DApp contract is concealed from the DApp.

4. The method of claim 1, further comprising:

receiving, by the at least one distributed node of the blockchain system, a transaction addressed to the entitlement contract, the transaction including an address authorized to conduct transactions with the DApp contract; and

5 executing, by the at least one distributed node, the entitlement contract to conduct the transaction, the executing including writing the authorized address to the list of registered addresses stored on the blockchain by the entitlement contract.

5. The method of claim 2, further comprising:

10 receiving, by the at least one distributed node of the blockchain system, a transaction addressed to the registry contract, the transaction including an identifier of the decentralized application; and

executing, by the at least one distributed node, the registry contract to conduct the transaction, the executing including writing the application identifier to the list of registered applications stored on the blockchain by the registry contract.

15 6. The method of claim 1, wherein the read call to the entitlement contract is performed by an entitlement checking function of the DApp contract in response to a transaction to the DApp contract invoking a predetermined blockchain modification function of the DApp contract.

20 7. The method of claim 6, wherein the predetermined blockchain modification function modifies data stored on the blockchain by the DApp contract, the modifying the data stored on the blockchain at least one of:

transferring ownership of assets from a first user to a second user;

conducting a financial transaction involving the first user and the second user; or

causing a device to generate an electrical signal to operate at least one of: a

25 mechanical apparatus, or an electrical apparatus.

8. A method of providing identity assurance for a decentralized application (DApp), the method comprising:

receiving, by at least one distributed node of a blockchain system, a transaction addressed to a contract of the DApp stored on a blockchain in the blockchain system, the

5 transaction including an address signing the transaction; and

executing, by the at least one distributed node, the DApp contract to conduct the transaction, the executing including:

if the transaction invokes a predetermined blockchain modification function of the DApp contract, executing an entitlement checking function of the DApp contract

10 including a read call to an entitlement contract stored on the blockchain, the read call containing the signing address;

receiving an output from the read call to the entitlement contract indicating whether the signing address is authorized to execute the predetermined blockchain modification function; and

15 if the signing address is authorized to execute the predetermined blockchain modification function, executing the predetermined blockchain modification function.

9. The method of claim 8, wherein an identity of a user corresponding to the address signing the transaction to the DApp contract is concealed from the DApp.

10. The method of claim 8, further comprising:

20 executing, by the at least one distributed node, the entitlement contract to execute the read call to the entitlement contract, executing the read call including:

reading a list of registered addresses stored on the blockchain by the entitlement contract;

25 determining whether the list of registered addresses includes the signing address included with the read call; and

providing an output indicating whether the list of registered addresses includes the signing address included with the read call.

11. A system for providing identity assurance for a decentralized application (DApp), the system comprising:

at least one processor; and

at least one non-transitory machine-readable storage medium including program

5 instructions, which when executed by the at least one processor cause the at least one processor to perform a method of providing the identity assurance for the DApp, the method including:

receiving, by at least one distributed node of a blockchain system, a transaction addressed to a contract of the DApp stored on a blockchain in the blockchain
10 system, the transaction including an address signing the transaction; and

executing, by the at least one distributed node, the DApp contract to conduct the transaction, the executing including:

if the transaction invokes a predetermined blockchain modification function of the DApp contract, executing an entitlement checking function of the DApp

15 contract including a read call to an entitlement contract stored on the blockchain, the read call containing the signing address;

receiving an output from the read call to the entitlement contract indicating whether the signing address is authorized to execute the predetermined blockchain modification function; and

5 if the signing address is authorized to execute the predetermined blockchain modification function, executing the predetermined blockchain modification function.

12. The system of claim 11, wherein executing the DApp contract further includes:

10 if the transaction invokes the predetermined blockchain modification function, executing an registry checking function of the DApp contract to perform a read call to a registry contract stored on the blockchain, the read call containing an identifier of the DApp contract;

receiving an output from the read call to the registry contract indicating an address on the blockchain system of the entitlement contract; and

addressing the read call to the entitlement contract using the received address.

15 13. The system of claim 11, the method further comprising:

executing, by the at least one distributed node, the entitlement contract to execute the read call to the entitlement contract, executing the read call including:

reading a list of registered addresses stored on the blockchain by the entitlement contract;

20 determining whether the list of registered addresses includes the signing address included with the read call; and

providing an output indicating whether the list of registered addresses includes the signing address included with the read call.

14. The system of claim 12, the method further comprising:

25 executing, by the at least one distributed node, the registry contract to execute the read call to the registry contract, executing the read call including:

reading a list of registered applications stored on the blockchain by the registry contract;

30 determining whether the list of applications includes the identifier included with the read call; and

providing an output indicating whether the list of registered applications includes the identifier included with the read call, and, conditional upon the list of registered

applications including the identifier, an address stored in the list of registered applications corresponding to entitlement contract associated with the DApp.

15. The system of claim 11, wherein an identity of a user corresponding to the address signing the transaction to the DApp contract is concealed from the DApp.

5 16. The system of claim 11, wherein the modifying the data stored on the blockchain at least one of:

transfers ownership of assets from a first user to a second user;

conducts a financial transaction involving the first user and the second user; or

causes a device to generate an electrical signal to operate at least one of: a mechanical

10 apparatus, or an electrical apparatus.

transmitting the signed transaction to the distributed node of the blockchain system.