

SYSTEMS AND METHODS FOR IDENTITY ATOMIZATION AND USAGE

CROSS REFERENCE TO RELATED APPLICATIONS

[01] This application claims priority to U.S. Provisional Patent Application No. 62/519,440, filed on June 14, 2017; is a continuation-in-part of and claims priority to U.S. Patent Application No. 15/922,505, filed on March 15, 2018, which claims priority to U.S. Provisional Patent Application No. 62/490,818, filed on April 27, 2017, and U.S. Provisional Patent Application No. 62/506,090, filed on May 15, 2017; and claims priority to U.S. Patent Application No. 16/007,581, filed on June 13, 2018; each of the above applications being hereby incorporated by reference herein in their entireties.

BACKGROUND INFORMATION

[02] In many types of transactions, and other scenarios, an individual must provide identity information to conduct the transaction. For example, to purchase alcohol, an individual is typically required to display a state identification, such as a state driver's license.

[03] One problem prevalent in many such transactions is that the required identification actually contains more information about the individual than is legally or otherwise required to conduct the transaction. For example, to purchase alcohol, all that is legally required to be presented is proof that the person is over the legal age to purchase alcohol, an image associated with the proof that looks like the person, and proof that the presented information is valid. However, when a person displays a state driver's license, they are actually displaying much more personal information than this, including their address, their exact date of birth, the driver's license number, an organ donor status, etc.

[04] Presenting more personal information than is necessary during a transaction has many privacy-related downsides that are desirable to avoid. Continuing the above example, when the person presents the state driver's license to purchase alcohol, oversharing of personal information such as their address may subject them to negative consequences such as stalking.

[05] Thus, a need exists for systems and methods to provide identity information for conducting transactions that limits the provided information to only that required by the transaction.

SUMMARY OF THE INVENTION

[05a] According to a first aspect of the invention, there is provided a method of providing identity information, the method comprising: receiving, from a transaction system, a request regarding a transaction; retrieving at least one rule defining a compliant identity based on a type of the transaction associated with the request, the at least one rule identifying a plurality of identity attributes; retrieving an identity index map including a list of pointers to available identity components; selecting pointers from the identity index map pointing to identity components corresponding to the plurality of identity attributes identified by the at least one rule; outputting, to the transaction system, the compliant identity including information corresponding to each of the identity components, the information including for each of the identity components at least one of: the selected pointers to the identity component, or the identity component.

[05b] According to a second aspect of the invention, there is provided a method of providing identity information, the method comprising: receiving identity information for an individual; processing the received identity information to identify an identity type of the identity information; retrieving a stored identity type definition based on the identity type, the identity type definition including a plurality of attributes of the identity type; atomizing the identity information into a plurality of identity information components based on the plurality of attributes of the identity type; distributing the identity information components to a plurality of different storage locations; and generating and storing an identity information index including a list of pointers to the distributed identity information components.

[05c] According to a third aspect of the invention, there is provided a system, comprising: at least one non-transitory machine readable storage medium having program instructions, mapping rules, and an index map; and at least one processor configured to execute the program instructions to perform a method of providing identity information, the method including: receiving, from a transaction system, a request regarding a transaction; retrieving at least one rule defining a compliant identity based on a type of the transaction associated with the request, the at least one rule identifying a plurality of identity attributes; retrieving an identity index map including a list of pointers to available identity components; selecting pointers from the identity index map pointing to identity components corresponding to the plurality of identity attributes identified by the at least one rule; outputting, to the transaction system, the compliant identity including information corresponding to each of the identity components, the information including for each of the identity components at least one of: the selected pointers to the identity component, or the identity component.

[05d] According to a fourth aspect of the invention, there is provided a system, comprising:
at least one non-transitory machine readable storage medium having program instructions;
and at least one processor configured to execute the program instructions to perform a
method of providing identity information, the method including: receiving identity
5 information for an individual; processing the received identity information to identify an
identity type of the identity information; retrieving a stored identity type definition based on
the identity type, the identity type definition including a plurality of attributes of the identity
type; atomizing the identity information into a plurality of identity information components
based on the plurality of attributes of the identity type; distributing the identity information
10 components to a plurality of different storage locations; and generating and storing an
identity information index including a list of pointers to the distributed identity information
component.

BRIEF DESCRIPTION OF THE DRAWINGS

[06] 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.

5 [07] FIG. 1 is a schematic diagram depicting an embodiment of a distributed data architecture.

[08] FIG. 2 is a schematic diagram depicting an embodiment of a data map of the distributed data architecture.

10 [09] FIG. 3 is a schematic diagram depicting an embodiment of a data fragment of the distributed data architecture.

[10] FIG. 4 is a schematic diagram depicting an embodiment of system for conducting a transaction, and processing data, according to the distributed data architecture.

[11] FIG. 5 is a schematic diagram depicting an embodiment of a computing system for implementing components of the system for conducting a transaction and processing data.

15 [12] FIG. 6 is a flowchart depicting an embodiment of a method of processing data according to the distributed data architecture.

[13] FIG. 7 is a flowchart depicting an embodiment of a method of conducting a transaction using data stored according to the distributed data architecture.

20 [14] FIG. 8A is a data flow diagram depicting an embodiment of data flows during the method of conducting a transaction using data stored according to the distributed data architecture.

[15] FIG. 8B is a data flow diagram depicting another embodiment of data flows during the method of conducting a transaction using data stored according to the distributed data architecture.

25 [16] FIG. 8C is a data flow diagram depicting another embodiment of data flows during the method of conducting a transaction using data stored according to the distributed data architecture.

30 [17] FIG. 8D is a data flow diagram depicting another embodiment of data flows during the method of conducting a transaction using data stored according to the distributed data architecture.

[18] FIG. 9 is a schematic diagram depicting another embodiment of the distributed data architecture.

[19] FIG. 10A is a schematic diagram depicting an embodiment of an identity document including data elements.

5 [20] FIG. 10B is a schematic diagram depicting an embodiment of a representation of the identity document stored using the distributed data architecture.

[21] FIG. 11 is a flowchart depicting an embodiment of a method of providing access to a data map.

10 [22] FIG. 12 is a flowchart depicting an embodiment of a method of refreshing data stored according to the distributed data architecture.

[23] FIG. 13 is a schematic diagram depicting an embodiment of a system for conducting transactions.

[24] FIG. 14 is a schematic diagram depicting an embodiment of an identity atomization and mapping system.

15 [25] FIG. 15 is a flowchart depicting an embodiment of a method of providing a compliant identity for a transaction.

[26] FIG. 16 is a flowchart depicting an embodiment of a method of conducting an atomization process to atomize information for use in providing a compliant identity for a transaction.

20 [27] FIG. 17 depicts an embodiment of identity information.

[28] FIG. 18 is a schematic diagram depicting an embodiment of an identity definition module of the identity atomization and mapping system.

[29] FIG. 19 is a schematic diagram depicting embodiments of an identity index module and an identity component module of the identity atomization and mapping system.

25 [30] FIG. 20 is a flowchart depicting an embodiment of a method of providing a compliant identity for a transaction based on atomized identity components.

[31] FIG. 21 is a schematic diagram depicting an embodiment of a communication interface of an identity mapping module of the identity atomization and mapping system.

30 [32] FIG. 22 is a schematic diagram depicting an embodiment of a mapping rules module of the identity atomization and mapping system.

[33] FIG. 23 depicts an embodiment of a compliant identity provided by the identity atomization and mapping system.

[34] FIG. 24 is a data flow diagram depicting embodiments of distribution of components of the identity atomization and mapping system and corresponding data flows among a user device, identity server system(s), and a transaction system.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

Distributed Data Architecture

[35] FIG. 1 depicts an embodiment of a distributed data architecture providing improved security for storage of and access to at least one data element. The distributed data architecture includes a data map 20 stored on a first system, such as a user device or data service system, and a plurality of fragments 24 of the at least one data element stored at a plurality of different storage locations on one or more second systems separate from the first system. For each data element stored by the distributed data architecture, the data map 20 includes a plurality of pointers 28 pointing to the plurality of the data fragments 24 making up the data element.

[36] Storing and/or accessing the data element may include a two stage process. In a first stage, or set of steps, the data map 20 is created, stored and/or accessed to create, store and/or access the pointers 28 identifying the locations of the data fragments 24 making up the data element. In a second stage, or set of steps, the plurality of the data fragments 24 are created, stored and/or accessed to create, store and/or access the data element. Both the data map 20 and/or first system, on the one hand, and the data fragments 24 and/or second system(s), on the other hand, provide security measures including, e.g., encryption, access policies, etc. Additionally, the first and second systems may be separate computing systems, the second system(s) may be located remotely from the first system, and the plurality of different storage locations may each be on separate second systems, be on separate devices, or require separate access transactions.

[37] The distributed data architecture thus provides improved security for storage of and access to data elements. The distributed, multi-tiered structure of the data architecture, including the data map 20 as a first tier, and the data fragments 24 constituting the data elements as a second tier distributed remotely from the first tier, provides improved security by limiting the damage that may be immediately done by an improper breach of the system, as such a breach likely results only initially in access of the first tier, enabling further protection of the actual data elements, which must be accessed through both the first and

second tiers. Moreover, to breach both tiers of the distributed data architecture, one must potentially breach a plurality of different data security systems, including, e.g., the encryption, etc. of the data map 20 and first system on the one hand, and the encryption, etc. of the data fragments 24 and second system(s) on the other hand, before the data elements may be accessed.

[38] FIG. 2 depicts an embodiment of a data map 20, including a plurality of map elements 32 for each data element stored according to the distributed data structure. Each map element 32 corresponds to a different respective data fragment 24 of the data element, and includes a pointer data field 36 and optionally one or more additional data fields. The pointer data field 36 includes a pointer 28 to the location of the data fragment 24 corresponding to that map element 32 on a corresponding second system. The pointer 28 may take a variety of different forms, such as an Internet address, an FTP address, a web address, etc.

[39] The additional data fields may include one or more of a fragment order field 42, a fragment identifier field 44, a fragment signature field 48, a fragment encryption field 52, or a fragment time-to-live field 56. The fragment order field 42 includes an order number of the fragment within the fragments 24 of the data element, for use in ordering the fragments 24 to reconstruct the data element. The fragment identifier field 44 includes an identifier uniquely identifying the fragment 24, such as a hash of a payload of the fragment 24. The fragment signature field 48 includes a cryptographic signature of the entire data fragment 24, such as for use in validating a fragment 24 upon retrieval to verify that it has not been altered. The fragment encryption field 52 includes information about the encryption scheme used to encrypt the fragment 24, such as for use in decrypting the fragment 24. The time-to-live field 56 includes a time-to-live attribute of the data fragment 24, such as indicating a time period for which the data fragment 24 is valid

[40] In embodiments, a data map may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 2.

[41] FIG. 3 depicts an embodiment of a data fragment 24. The depicted fragment 24 includes a fragment payload field 60 and optionally one or more additional data fields. The fragment payload field 60 includes the fragment of the data element. The fragment may be stored in the fragment payload field 60 in an encrypted form.

[42] The depicted additional fields include a fragment identifier field 64 and fragment metadata fields 68. The fragment identifier field 64 includes the unique identifier of the

data fragment 24. The depicted metadata fields 68 include a time-to-live field 72, a payload signature field 76, a master key identifier field 80, an originator context field 84, a nonce field 88, and a decryption key field 92. The time-to-live field 72 includes the time-to-live attribute of the data fragment 24. The payload signature field 76 includes a cryptographic signature of the payload field 60, such as for use in validating the payload upon retrieval to verify that it has not been altered. The master key identifier field 80 includes a unique identifier of the master key used to encrypt a fragment decryption key, such as for use, along with the nonce field 88, to generate a unique encryption key per fragment 24. The originator context field 84 includes any relevant information about the issuer of the fragment 24. The nonce field 88 includes a nonce value, such as may be used to randomize an encryption key used to encrypt the fragment decryption key. The fragment decryption key field 92 includes a randomized encryption key that can decrypt the fragment 24, encrypted with the master key.

[43] In embodiments, a data fragment may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 3.

[44] FIG. 4 depicts an embodiment of a system 100 for conducting a transaction using the distributed data architecture. The depicted system 100 includes a user device 104, a transaction system 108, a data service system 112, one or more data storage systems 116, and one or more communication networks 120.

[45] The user device 104 is used by a user to communicate with the transaction system 108 via one or more communication networks 120 to conduct a transaction with the transaction system 108. The user device 104 includes a transaction module 124, a distributed data module 128, and a communication interface 132. The transaction module 124 communicates, using the communication interface module 132 via the one or more communication networks 120, with the transaction system 108 to conduct the transaction. The distributed data module 128 communicates, using the communication interface module 132 via the one or more communication networks 120, with the data service system 112 and/or data storage systems 116 to retrieve data elements according to the distributed data architecture, for provision, such as via the transaction module 124, to the transaction system 108 for conducting the transaction.

[46] The transaction system 108 is a system used by a person or organization to conduct the transaction with the user. The transaction system 108 includes a transaction module 136 and a communication interface module 140. The transaction module 136 communicates, using the communication interface module 140 via the one or more

communication networks 120, with the user device 104, data service system 112 and/or data storage systems 116 to conduct the transaction.

[47] The data service system 112 is a system providing data services to the user device 104 via one or more communication networks 120 to enable conducting the transaction with the transaction system 108 using the distributed data architecture. The data service system 112 includes a distributed data module 144 and a communication interface module 148. The distributed data module communicates 144, using the communication interface module 148 via the one or more communication networks 120, with the user device 104 and/or data storage systems 116 to provide data services according to the distributed data architecture.

[48] The data storage systems 116 store and provide access via one or more communication networks 120 to the data fragments of the distributed data architecture. The data storage systems 116 each include a data storage module 152 and a communication interface 156. The data storage module 152 stores and provides selective access for the user device 104 and/or data service system 112, using the communication module 156 via the one or more communication networks 120, to one or more data fragments of the distributed data architecture.

[49] In embodiments, the plurality of different storage locations at which the plurality of data fragments 24 are stored may include locations at a corresponding plurality of different data storage systems 116. For example, each of the plurality of the data fragments 24 may be stored using a respective different one of a plurality of the data storage systems 116. In such embodiments, each data storage system 116 may be implemented using a separate computing system.

[50] In embodiments, the plurality of different data storage locations at which the plurality of data fragments 24 are stored may include locations at a corresponding plurality of different data storage devices. For example, each of the data fragments 24 may be stored using a respective different one of a plurality of different data storage devices. The data storage devices may include hard drives, flash memories, etc. In such embodiments, a plurality of the different data storage devices may be part of a single data storage system 116 (for example, the data storage module 152 may include a plurality of different data storage devices), part of a plurality of different data storage systems 116 (for example, a plurality of data storage systems 116 may each include a data storage module 152 including a respective one of the plurality of data storage devices), or combinations thereof.

[51] In embodiments, the plurality of different data storage locations at which the plurality of data fragments 24 are stored may include multiple locations on a single data storage system 116 or device. For example, each of the data fragments may be stored at different data storage locations on a single data storage system 116 or device. In such
5 embodiments, each of the plurality of different data storage locations may require a separate storage and/or access transaction, or set of steps, to store and/or access data at that data storage location. In one example, the plurality of different data storage locations may be within a plurality of different partitions of, or behind other logical or physical boundaries within, the data storage system 116 or device that would require such separate
10 transactions or sets of steps.

[52] The one or more communication networks 120 may include one or more of the Internet, a cellular network, a wireless local area network (e.g., WiFi), etc.

[53] The transaction may include one or more of: a security transaction, in which the user provides identity data to prove his or her identity to the transaction system 108; a financial
15 transaction, in which the user conducts a financial transaction, such as purchasing an item, making a payment, exchanging currency, etc.; other types of transactions in which the user provides some type of data regarded as sensitive to the transaction system 108; or variations and/or combinations of these types of transactions.

[54] Embodiments of the user device 104, transaction system 108, data service system
20 112, data storage systems 116, and/or any individual one, subset, or all of the components thereof, such as the communication interfaces, transaction modules, distributed data modules, data storage modules, etc., may be implemented as hardware, software, or a mixture of hardware and software. For example, each of the user device, transaction system, data service system, data storage systems, and/or any individual one, subset, or all
25 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. In exemplary embodiments, each of the user device, transaction system, data service system, data storage systems, and/or any
30 individual one, subset, or all of the components thereof, may be implemented using one or more computer systems, such as, e.g., a mobile computing device, a desktop computer, laptop computer, network device, server, Internet server, cloud server, etc.

[55] In embodiments, a system for conducting a transaction using the distributed data architecture may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 4.

[56] FIG. 5 depicts an embodiment of a computer system 160 that may be used to implement the user device 104, transaction system 108, data service system 112, data storage systems 116, and/or any individual one, subset, or all of the components thereof. The computer system 160 includes a processor 164, a non-transitory machine-readable storage medium 168, a communication circuit 172, and optionally other components 176.

The processor 164 executes program instructions stored in the non-transitory machine-readable storage medium 168 to perform the functionality of the component it is implementing as discussed herein. The communication circuit 172 can be controlled by the processor 164 to communicate with other devices, such as the user device 104, transaction system 108, data service system 112 and/or data storage systems 116, to perform the functionality of the component it is implementing as discussed herein. The optional other components 176 may include any further components required by the computer system 160 to perform this functionality.

[57] In embodiments, a computer system that may be used to implement the user device, transaction system, data service system, data storage systems, and/or any individual one, subset, or all of the components thereof may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 5.

[58] FIG. 6 depicts an embodiment of a method 600 of processing and storing data according to the distributed data architecture to enable use of the data in conducting transactions according to the distributed data architecture. The method provides the improved security of the distributed data architecture for data stored according to the distributed data architecture. The method may be performed by or involving components of the system of FIG. 4, such as by the user device 104 and/or data service system 112. The method begins at step 602.

[59] At step 604, data elements to be stored according to the distributed data architecture are encrypted. The data elements may be encrypted using various different encryption techniques. For example, the data element may be encrypted using one or more of an encryption standard, such as the Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA), etc.; a hashing standard, such as the Secure Hash Algorithm 2 (SHA-2), etc.; a digital signature, such as RSA, elliptic curve Digital Signature Algorithm (ECDSA),

Digital Signature Standard (DSS), etc.; public key infrastructure (PKI); secret sharing algorithms, such as Shamir's Secret Sharing, etc.

[60] At step 606, each encrypted data element is split apart into a plurality of separate data fragments 24. Each data fragment 24 is a different portion of the data element, and together, the plurality of data fragments 24 constitute the entire data element. The data element may be split using various different splitting techniques. For example, the data element may be split to create a plurality of data fragments 24 having contiguous portions of the original data element. Alternatively, the data element may be split into a plurality of sub-fragments, numbering a multiple of an eventual number of the plurality of data fragments 24, and then the subfragments combined to form the fragments 24, such as by combining interleaved slices of the data element, or by combining random slices of the data element, into each fragment 24. In another alternative, the data element may be split into a plurality of data fragments 24 or subfragments as part of an encryption process using a secret sharing algorithm such as Shamir's Secret Sharing.

[61] In embodiments, the order of steps 604 and 606 may be reversed. That is, the data element may first be split into the plurality of fragments 24, and then each of the fragments 24 encrypted using the encryption techniques.

[62] At step 608, time-to-live (TTL) attributes are assigned to each of the data fragments 24. The TTL attributes provide even further security for the distributed data architecture by controlling the time period for which the data fragments 24, and thus the data elements constituted by the data fragments 24, are valid and accessible. Controlling access to the data as a function of time further limits the potential for improper access to the data, as it prevents the presence of data from being forgotten and left in a potentially unsecure state by automatically invalidating and preventing access to the data once the selected lifetime of the data has expired. The TTL attribute may indicate the validity time period in a variety of ways. For example, the TTL attribute may include one or more of a time at which the validity of the fragment expires, a time at which the validity of the fragment begins, etc.

[63] At step 610, a distribution of the plurality of encrypted fragments 24 among a plurality of separate data storage locations is determined. In embodiments, the plurality of different data storage locations may be at separate data storage systems 116, separate storage devices, or combinations thereof. The plurality of different data storage systems 116 or devices also may be located at physically separate locations, such as at separate storage facilities. Storing a plurality of data fragments 24 on a plurality of different data storage systems 116 or devices generally may require a separate data storage transaction or set of

steps for each system 116 or device. In embodiments, the plurality of different data storage locations may include a plurality of different locations on a single data storage system 116 or device, where the plurality of different locations may each require a separate transaction or set of steps to store or access data at such location. The distribution may be determined
5 in a variety of ways. For example, the distribution may be determined by distributing each fragment 24 to a different one of the plurality of separate data storage locations.

[64] At step 612, a data map 20 including a map element 32 corresponding to each of the data fragments 24 is generated and stored. The data map 20 and constituent map elements 32 may take a variety of forms, such as, e.g., the form of the exemplary data map
10 20 and map elements 32 depicted in FIG. 2. Each map element 32 includes a field containing a pointer 28 to the location at the data storage system 116 on which the corresponding data fragment 24 is stored. The map elements 32 may include one more further fields containing further information, such as one or more of the TTL attribute and the other information discussed above in regard to FIG. 2. The generated data map 20 may
15 be stored in various locations. For example, the data map 20 may be stored on the user device 104 or the data service system 112, according to various different embodiments, such as discussed further below in regard to FIGS. 8A-8D.

[65] At step 614, the encrypted data fragments 24 are distributed to and stored at the plurality of separate data storage locations according to the determined distribution. The
20 distribution and storage of each data fragment 24 at a corresponding one of the plurality of different data storage locations may include executing a corresponding separate transaction or set of steps to perform the distribution and storage. The method ends at step 616.

[66] In embodiments, a method of a method of processing and storing data according to the distributed data architecture may include only any subset of, or an alternative ordering
25 of, the features depicted in or discussed above in regard to FIG. 6.

[67] FIG. 7 depicts an embodiment of a method 700 of conducting a transaction using data stored according to the distributed data architecture. The method provides the improved security provided by the distributed data architecture for data involved in the transaction. The method may be performed by or involving components of the system of
30 FIG. 4, such as by the user device 104 and/or data service system 112. The method begins at step 702.

[68] At step 704, a request for one or more data elements for conducting a transaction is received. In embodiments in which a user uses the user device 104 to conduct a

transaction with the transaction system 108, the request may be received by the user device 104 from the transaction system 108. For example, the user device 104 may be a mobile device, such as a smart phone, tablet, etc., and the transaction system 108 may be an Internet-based system, such as a website, application server, etc. Other types of user devices 104 and transaction systems 108 are also possible.

[69] As discussed above, the transaction may include one or more of a variety of different types of transactions, such as a security authorization, a purchase transaction, a credit application, etc. The requested one or more data elements include data relevant to the transaction. For a security authorization or other identity verifying transaction, the requested data element may include identity data, such as one or more of a name of the user, an mailing address of the user, an email address of the user, a picture of the user, a driver's license number of the user, a passport number of the user, etc. For a purchase, credit application or other financial transaction, the requested data element may include financial data, such as one or more of a currency amount, an account number, a fund transfer authorization, etc. Other types of transactions may involve other types of data elements.

[70] At step 706, a data map 20 corresponding to the requested data element is accessed to determine the pointers 28, TTL attributes, etc. for the data fragments 24 constituting the data element. The data map 20 may be accessed in a variety of ways. In embodiments, the data map 20 is accessed by the user device 104, while in other embodiments, the data map 20 is accessed by the data service system 112. In embodiments, an existing data map 20 is accessed, while in other embodiments, a data map 20 that is generated specifically for conducting the transaction in response to the data element request is accessed. Further details of such embodiments are discussed below in regard to FIGS. 8A-8D.

[71] At step 708, the TTL attributes for the data fragments 24 constituting the requested data element in the data map 20 are examined to determine whether the data fragments 24 are all still currently valid. If the TTL attributes for all of the data fragments 24 constituting the data element indicate that the fragments 24 are currently valid, the method proceeds to step 710 to retrieve the fragments 24, etc. If the TTL attributes for any of the data fragments 24 constituting the data element indicate that fragments 24 are currently invalid, the method proceeds to step 718, where the method ends, to enforce the TTL expiration by preventing retrieval of the expired fragments 24. In embodiments, the examining and enforcing of the TTL attributed may be performed by the user device 104, while in other embodiments it may be performed by the data service system 112. The TTL attributes may

be found in the TTL field 56 of the data map 20 and/or the TTL field 72 of the fragment metadata 68 of the fragment 24.

[72] At step 710, the pointers 28 in the data map 20 for the data fragments 24 constituting the requested data element are used to retrieve the data fragments 24. The data fragments 24 may be retrieved in a variety of ways, such as via communication with the data storage system(s) 116 or device(s) containing the data fragments 24 over the one or more communication networks 120 according to data transfer protocols used by such system(s) 116 or device(s). The retrieval of each data fragment 24 from a corresponding one of the plurality of different data storage locations may include executing a corresponding separate transaction or set of steps to perform the retrieval. In embodiments, the retrieving of the data fragments 24 may be performed by the user device 104, while in other embodiments it may be performed by the data service system 112.

[73] At step 712, the data element is reconstructed from the retrieved data fragments 24. In embodiments in which the data element was first encrypted and then fragmented, the data fragments 24 may be directly combined to reconstruct the encrypted data element. In embodiments in which the data element was first fragmented and then the fragments 24 encrypted, the retrieved data fragments 24 may first be decrypted and then combined to reconstruct the data element. The data fragments 24 may be combined to form the data element according to the recipe used to fragment the data element, such as creating contiguous, interleaved or randomized combinations of portions of the data element, which may be determined, e.g., from the map elements 32 corresponding to the data fragments, such as the fragment encryption field 52. In embodiments, the reconstruction of the data element from the retrieved data fragments 24 may be performed by the user device 104, while in other embodiments it may be performed by the data service system 112. In embodiments, prior to reconstruction, or as part of the reconstruction process, each of the retrieved data fragments 24 may be validated, such as using the fragment signature 48, in order to protect against malicious manipulation of the fragments 24 by other parties.

[74] At step 714, the reconstructed data element is decrypted and validated. The data element may be decrypted according to the encryption used to encrypt the data element, which may be determined from the map elements corresponding to the data fragments 24, such as the fragment encryption field 52, and fragment metadata 68. In embodiments in which the data element was first fragmented and then the fragments 24 encrypted, and the retrieved data fragments 24 are decrypted and then combined to reconstruct the data

element, the data element may simply be validated, or decrypted and validated if further encryption is still present on the reconstructed data element.

[75] At step 716, the requested data element is provided for conducting the transaction. In embodiments in which a user uses the user device 104 to conduct a transaction with the transaction system 108, the requested data element may be provided by the user device 104 to the transaction system 108. The data element may be encrypted as part of a communication protocol between the user device 104 and the transaction system 108. As a result of providing the data element, the transaction may be performed to completion.

[76] In embodiments, a method of a conducting a transaction using data stored according to the distributed data architecture may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 7.

[77] As discussed above, accessing the data map 20, retrieving the data fragments 24, and/or reconstructing the data element may selectively be performed by the user device 104, the data service system 112, or the transaction system 108. FIG. 8A-8D depict embodiments of the method of conducting a transaction of FIG. 7, showing further details with respect to the flow of data between devices during the method.

[78] In FIG. 8A, the user device 104 receives the request for the data element from the transaction system 108, then requests the data map from the data service system 112. The data service system 112 may either retrieve an existing data map 20 in its possession, or generate a new data map 20 specifically for the transaction. Storing or generating the data map 20 on the data service system 112 may provide further benefits of the distributed data architecture by limiting the degree of exposure of exposure of the data map 20 to the user device 104, which may involve greater security risks than the data service system 112. Generating a data map 20 specifically for a transaction may also provide further benefits of the distributed data architecture by limiting the degree of exposure of the data map 20 to even the data service system 112, which may even itself entail security risks. Moreover, generating a data map 20 specifically for a transaction may also provide even further benefits of the distributed data architecture by tailoring the content of the data map 20 to the transaction, and thereby not pointing to, and thus not exposing to risk, data in the data map 20 unnecessary for the transaction.

[79] Returning to FIG. 8A, the data service system 112 provides the data map 20 to the user device 104, which then uses the data map 20 to request the data fragments 24 corresponding to the requested data element from the data storage systems 116. Upon

receiving the data fragments 24 from the data storage systems 116, the user device 104 reconstructs the data element, and provides the data element to the transaction system 108. Note that in FIGS. 8A-8D, although two data fragments 24 and two data storage systems 116 are shown, the number of the plurality of data fragments 24 and the one or more data storage systems 116 may vary in various different embodiments.

[80] In FIG. 8B, the user device 104 again receives the request for the data element from the transaction system 108, but then requests the data element from the data service system 112. The data service system 112 then either retrieves an existing data map 20 in its possession or generates a new data map 20 specifically for the transaction. The data service system 112 then uses the data map 20 to request the data fragments 24 corresponding to the requested data element from the data storage systems 116. Upon receiving the data fragments 24 from the data storage systems 116, the data service system 112 reconstructs the data element, and provides the reconstructed data element to the user device 104, which provides it to the transaction system 108. Such an embodiment may provide the benefit of locating many of the manipulations of the distributed data architecture on the potentially more secure data service system 112 instead of the user device 104.

[81] In FIG. 8C, during the transaction the user device 104 provides a link to the data service system 112 to the transaction system 108, such as in response to a request for a data element or other aspect of the transaction. The data service system 112 then receives the request for the data element from the transaction system 108. The data service system 112 then either retrieves an existing data map 20 in its possession, or generates a new data map 20 specifically for the transaction. The data service system 112 then uses the data map 20 to request the data fragments 24 corresponding to the requested data element from the data storage systems 116. Upon receiving the data fragments 24 from the data storage systems 116, the data service system 112 reconstructs the data element, and provides the reconstructed data element to the transaction system 108. This embodiment also may provide the benefit of locating many of the manipulations of the distributed data architecture on the potentially more secure data service system 112 instead of the user device 104.

[82] In FIG. 8D, the user device 104 receives the request for the data element from the transaction system 108, and then itself either retrieves an existing data map 20 in its possession or generates a new data 20 map specifically for the transaction. The user device 104 then uses the data map 20 to request the data fragments 24 corresponding to the requested data element from the data storage systems 116. Upon receiving the data fragments 24 from the data storage systems 116, the user device 104 reconstructs the data

element, and provides the reconstructed data element to the transaction system 108. This embodiment may provide a benefit of a simplified system by eliminating the need for the data service system 112.

[83] In embodiments, the retrieved data fragments 24 may be provided to the transaction system 108 instead of the reconstructed data element, with the transaction system 108 then reconstructing the data element from the data fragments 24. For example, the embodiments of FIGS. 8A-8D may be adapted to have the device that retrieves the data fragments, such as the user device 104 or data service system 112, forward the retrieved data fragments 24 to the transaction system 108, for reconstruction by the transaction system 108, instead of that device reconstructing the data element and forwarding the reconstructed data element to the transaction system 108.

[84] The distributed data architecture may provide a hierarchy of multiple layers of data maps. FIG. 9 depicts an embodiment of the distributed data architecture having a first data map 20-1 including a plurality of pointers 28-1 to a plurality of data fragments 24-1 (only one of which is shown in the figure). The plurality of fragments 24-1 are distributed to a plurality of different data storage locations as discussed above. The first map 20-1 may map at least one data element to the plurality of data fragments 24-1. However, in the depicted embodiment, the payload 60-1 of at least one of the data fragments 24-1 contains a second data map 20-2 instead of an actual fragment of the data element. The second data map 20-2 itself includes a plurality of pointers 28-2 to a plurality of second data fragments 24-2 (only one of which is shown in the figure). The plurality of second fragments 24-2 are again distributed to a plurality of different data storage locations as discussed above. The second map 20-2 may map the corresponding data fragment 24-1 into the plurality of further data fragments 24-2. The plurality of further fragments 24-2 include payloads 60-2 that contains actual data fragments of the first data fragment 24-1. Providing such a hierarchy of data maps 20 further improves the security of the distributed data architecture by increasing the number of layers of the distributed data architecture that must be traversed by a breach before actual data fragments are reached. Note that, although FIG. 9 shows two layer of data maps 20, other hierarchical embodiments may include any number of layers of data maps 20.

[85] The distributed data architecture may selectively map data elements of a data set in data maps 20 in various ways. For example, the distributed data architecture may map a related set of data elements into a single data map 20. FIG. 10A depicts an embodiment of an identity document 180 such as a driver's license, passport, etc. The identity document

180 has a number of data elements 184, such as a first attribute 184-1 including a name of a person identified by the identity document 180, a second attribute 184-2 including an identification number of the identity document 180, a third attribute 184-3 including a mailing address of the person, and a fourth attribute 184-4 including a picture of the person.

5 FIG. 10B depicts an embodiment of a data map 20 mapping the data elements of the identity document 180. The data map 20 includes a first set of map elements 32-1 including a first set of pointers to map the first attribute to a plurality of first data fragments, a second set of map elements including a second set of pointers to map the second attribute to a plurality of second data fragments, a third set of map elements including a third set of pointers to map the third attribute to a plurality of third data fragments, and a fourth set of map elements 32-4 including a fourth set of pointers to map the fourth attribute to a plurality of fourth data fragments. Other embodiments may selectively map data sets in other ways. For example, in other embodiments, a related data set may be mapped to a plurality of data maps 20, such as mapping the data elements of the identity document 180 to a plurality of different data maps 20. In other embodiments, several data sets may be mapped to a single data map 20.

[86] A hierarchical embodiment of the distributed data architecture may also be used to selectively map data elements of a data set. For example, the identity document 180 depicted in FIG. 10A may be mapped using a hierarchical embodiment of the distributed data architecture such as depicted in FIG. 9. In such embodiments, one or more of the map elements 32-1 ... 32-4 depicted in FIG. 10B may instead point to data fragments 24 containing an additional data map 20 mapping the actual data fragments 24 or even further layers of data maps 20.

[87] The distributed data architecture may be used to provide selective access to a data map 20 or copy of a data map 20 to enable various functions, such as recovery of a lost data map 20, providing legally authorized access to data for legal authorities, etc. FIG. 11 depicts an embodiment of a method of providing access to a data map of the distributed data architecture. The method may be performed by or involving components of the system of FIG. 4, such as by the data service system. The method begins at step 1102.

30 [88] At step 1104, a data map 20 and corresponding data fragments 24 according to the distributed data architecture are generated and distributed. The data map 20 and corresponding data fragments 24 may be generated and distributed as discussed above, such as according to embodiments of the method of FIG. 6.

[89] At step 1106, a copy of the data map 20 is encrypted using a multi-party encryption technique that requires multiple parties to participate in the encryption and any subsequent decryption, and the encrypted data map stored. The multi-party encryption technique may be based on, e.g., Shamir's Secret Sharing, etc. The multiple parties may take various forms, and may include an operator of the data service system 112, operators of the data storage systems 116, other entities, etc. The encrypted data map 20 may be stored by one or more of the data service system 112, one of the data storage systems 116, etc.

[90] At step 1108, a request for the data map 20 is received. A request for access to the data map 20 may be received from various different entities. In one example, the user may lose the user device 104 and any data maps 20 stored on the user device 104, and may need to request to retrieve a copy of the data maps 20 for restoring them onto a new device in order to continue using them. In another example, the user may be the subject of a legal or regulatory proceeding or investigation, and a legal entity, such as a lawyer, law enforcement agency, court, or regulator may have a legal right to access to the data elements, even without authorization by the user. The request may be received by an entity facilitating providing the access to the copy of the data map 20, such as the data service system 112, the data storage system 116, etc.

[91] At step 1110, it is determined whether consensus exists among the multiple parties involved in the encryption and needed to perform the decryption of the copy of the data map 20 as to whether to provide access to the data map 20 for the requesting entity. For example, if the user is requesting access to his or her own data map 20, the multiple parties may be satisfied by the user undergoing an authentication process, such as entering a password, etc. If a legal entity is requesting access to a user's data, such as without authorization by the user, the multiple parties may require the requesting entity to provide a legally sufficient authorization, such as a valid search warrant, etc. If it is determined at step 1110 that there is no consensus to provide access to the data map 20 in response to the request, such as if sufficient the authentication and/or legal authorization has not been provided, the method proceeds to step 1114, where the method ends. If it is determined at step 1110 that there is consensus to provide access in response to the request, such as if sufficient authentication and/or legal authorization has been provided, the method proceeds to step 1112.

[92] At step 1112, the data map 20 is decrypted with input from the multiple parties according to the multi-party decryption scheme, and provided to the requesting entity. The decryption and providing may be performed by an entity facilitating providing the access to

the copy of the data map 20, such as the data service system 112, the data storage system 116, etc., with the requisite input from the multiple parties.

[93] In embodiments, a method of providing access to a data map of the distributed data architecture may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 11.

[94] The distributed data architecture may be refreshed when one or more data fragments 24 are found to be invalid. FIG. 12 depicts an embodiment of a method of refreshing the distributed data architecture. The method may be performed by or involving components of the system of FIG. 4, such as by the user device 104, data service system 112, data storage systems 116, etc. The method begins at step 1202.

[95] At step 1204, it is determined that one or more data fragments 24 pointed to by one or more data maps 20 are invalid. The determination may be made based on to various factors, such as expiration, revoking, or corruption of the data fragments 24. For example, data fragments 24 of a data element of identity data may be deem to be invalid when an identity document, of which the data element is a part, expires, is revoked, etc. In another example, data fragments 24 may have been found to have suffered corruption of their integrity, such as due to a system failure, unauthorized access, etc. In embodiments, the determination may be performed by the data service system 112, such as either on its own or in response to information that it receives.

[96] At step 1206, any data maps 20 pointing to fragments determined to be invalid are invalidated. The invalidation may be performed in various ways. For example, the data service system 112 may suspend functionality related to the invalidated data maps 20 in its system, delete such data maps 20, and/or communicate such suspension and/or deletion instructions to the user device 108.

[97] At step 1208, one or more replacement data fragments 24 may be generated and distributed. The replacement data fragments 24 may cure the deficiencies of the invalid data fragments 24, such as no longer being a part of an expired, revoked or corrupt data set. The data service system 112 may generate the new data fragments 24 from a valid replacement data element that it receives. The data fragments 24 may be generated and distributed as discussed above, such as using embodiments of the method of FIG. 6.

[98] At step 1210, replacement data maps 20 pointing to the replacement data fragment 24 may be generated and distributed. The replacement data maps 20 may be generated and distributed as discussed above, such as using embodiments of the method of FIG. 6.

[99] In embodiments, a method of refreshing the distributed data architecture may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 12.

[100] Additional embodiments of the user device 104, transaction system 108, data service system 112, data storage systems 116, and associated methods, as 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.

Identity Atomization and Mapping

[101] FIG. 13 depicts an embodiment of another system for conducting transactions 220, including an identity atomization and mapping system 224 and a transaction system 236.

[102] The identity atomization and mapping system 224 receives identity information, such as from a user device 228 or an identity provider system 232, atomizes the identity information into corresponding identity components, and generates an index to the atomized identity information. The identity atomization and mapping system 224 also provides a compliant identity containing a sufficient set of identity the atomized identity components to the transaction system 236, such as via communication interfaces 240, 244. The compliant identity may include a subset of identity information from a single identity, such as a state or national identity, or may combine information from a plurality of identities, such as a subset of information from each of the plurality of identities.

[103] The identity atomization and mapping system 224 is thus able to provide identity information sufficient to satisfy the requirements of the transaction, while protecting from exposure identity components not required for the transaction, greatly enhancing the privacy of the identity information of the individual. The identity atomization and mapping system 224 also enhances the efficiency of the system for conducting transactions 220 by providing only the sufficient identity information and thereby reducing the transmission and processing costs of providing identity information not required for a transaction.

[104] The identity atomization and mapping system 224 may be implemented on one or more of the user device 28 and the identity server system 232. The user device 228 may be any device used by a user to communicate with the transaction system 236, such as via one or more communication networks or other communication interfaces, to conduct a

transaction with the transaction system 236. The identity server system 232 is a system providing identity data services as discussed herein.

[105] The transaction system 236 is a system used by a person or organization to conduct the transaction with the user. The transaction may be one or more of: a security

5 transaction, in which the user provides identity information to prove his or her identity to the transaction system, such as at an airport, building, venue, border, etc., to gain access to a restricted area, such as to a secure area, across a border, etc.; a financial transaction, in which the user provides identity information to conduct and/or authorize conducting of a financial transaction, such as purchasing an item, making a payment, exchanging currency,
10 etc., such as either in person at a point-of-sale location or online; or other types of transactions in which the user provides identity data to the transaction system.

[106] In exemplary embodiments, each of the identity atomization and mapping system 224, user device 228, identity server system 232, transaction system 236, and/or any individual one, subset, or all of the components thereof, may be implemented using one or

15 more computer systems, such as, e.g., a mobile computing device, a desktop computer, laptop computer, network device, server, Internet server, cloud server, etc. For example, the user device 228 may be a mobile computing device, a desktop computer, laptop computer, etc.; the identity server system 232 may be a network device, server, Internet server, cloud server, etc.; and the transaction system 236 may be a kiosk computing device,
20 point-of-sale device, network device, server, Internet server, cloud server, etc.

[107] In embodiments, a system for conducting transactions may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 13.

[108] FIG. 14 depicts an embodiment of the identity atomization and mapping system 224, including an identity atomization module 248, an identity definition module 252, an identity
25 index module 256, an identity component module 260, an identity mapping module 264, a mapping rules module 268, and an interface module 272.

[109] The identity atomization module 248 receives identity information for an individual; atomizes, or separates, the identity information into a plurality of separate identity
30 components corresponding to separate attributes of the identity; and stores the atomized identity components in the identity component module 260. The atomization module 248 atomizes identity information into components based on the type of the identity information, using an identity type definition retrieved from the identity type definition module 252. The

atomization module 248 also generates one or more identity index maps containing pointers to the atomized identity components stored in the identity component module 260, and stores the identity index maps in the identity index module 256.

[110] The mapping module 264 receives a request associated with a transaction of the transaction system 236, and in response provides a compliant identity to the transaction system 336 having selected atomized identity components satisfying the request. The mapping module 264 determines the selected components for the compliant identity based on the type of transaction being conducted, using a compliant identity definition for that transaction type retrieved from the mapping rules module 268. The provided compliant identity may include the selected identity components themselves, pointers to the selected identity components, or a combination thereof. The mapping module 264 retrieves pointers to the selected components from the one or more index maps of the identity index module 256, and retrieves the selected components from the identity component module 260 using the pointers. The mapping module 64 may be controlled by the user using the interface module 272.

[111] In embodiments, an identity atomization and mapping system may include only any subset of, or an alternative connection or ordering of, the features depicted in or discussed herein in regard to FIG. 14.

[112] FIG. 15 depicts an embodiment of a method 1500 of providing a compliant identity for conducting a transaction. The method may be performed by or involving components of the system for conducting transactions 220 of FIG. 13, such as by embodiments of the identity atomization and mapping system 224 and the transaction system 236. The method provides identity information sufficient to satisfy the requirements of the transaction, while protecting from exposure identity information not required for the transaction. The method also enhances the efficiency of conducting transactions by providing the sufficient identity information, while reducing transmission and processing costs that would otherwise be associated with surplus identity information not provided. The method begins at step 1502. At step 1504, an atomization process is conducted to atomize identity information received for an individual into a plurality of separate components corresponding to separate attributes of the identity information, and generate an index to the atomized components. At step 1506, a compliant identity is provided for a transaction, the compliant identity containing a sufficient set of the atomized identity components for the transaction being conducted. At step 1508, the transaction is conducted based on the provided compliant identity. The method ends at step 1510.

[113] In embodiments, a method of a method of providing a compliant identity for conducting a transaction may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 15.

[114] FIG. 16 depicts an embodiment of a method of conducting an atomization process to atomize identity information for use in providing a compliant identity for a transaction.

Embodiments of the method 1600 may be used to perform the atomization process of step 1504 of the method 1500 of FIG. 15. The method may be performed by or involving components of the identity atomization and mapping system 224 of FIG. 14, such as by one or more of the atomization module 248, identity definition module 252, identity index module 256 or identity component module 260. The method separates identity information into attribute-level identity components, generates an index to the atomized components, and stores the index and atomized components to enable the provision of compliant identities that protect the privacy of identity information while also enhancing the efficiency of conducting transactions. The method begins at step 1602.

[115] At step 1604, identity information is received. The identity information may be received by the atomization module 248. The receiving the identity information may include creating an electronic copy of the identity information, such as by one or more of capturing an image of the identity information, receiving the identity information electronically, or inputting the identity information via an interface. For example, one or more images of one or more identity documents containing the identity information may be captured, such as by an image sensor of the atomization module 248. In another example, an electronic form of one or more identity documents containing the identity information, such as one or more files or packets, etc., may be received, such as by an electronic communication module of the atomization module 248.

[116] The received identity information may include identity information of a variety of different forms. For example, the identity information may include one or more state or nationally issued identity documents, such as a driver's license, passport, social security card, etc. The identity information also may include biometric information, such as one or more of a DNA segment, fingerprint, iris pattern, retina pattern, facial pattern, voice pattern, signature pattern, etc. The identity information also may include characteristics of an electronic or online identity of an individual, such as an electronic mail or social media identity, such as one or more of an e-mail address, a social media account name (such as for one or more social media platforms such as Twitter, Facebook, Instagram, Snapchat,

etc.), a social media account password, etc. The identity information also may include a legal assertion being made by an individual.

[117] At step 1606, the received identity information is processed to determine a type of the identity information. The identity information may be processed by the atomization module 248. The identity information may be processed in a number of different ways to determine the type of identity information. For example, the identity information may be processed to recognize an indicator that corresponds to a specific type of identity information, such as by one or more of performing text recognition on a captured image of the identity information, performing image analysis on a captured image of the identity information, performing text parsing of an electronic form of the received identity information, etc. The indicator may include one or more of an identity type identifier in the identity information that directly identifies the identity type, such as text of the identity information identifying it as driver's license, passport, fingerprint, account name, etc., or an indicator that indirectly identifies the information as the identity type, such as particular format, graphical element, etc. of the identity information.

[118] At step 1608, a definition for the determined identity type is retrieved. The identity type definition may be retrieved by the atomization module 248 from the identity definition module 252. The identity definition module 252 stores definitions for a plurality of different identity types that the identity atomization and mapping system 224 may receive, such as one or more of the types of identity information discussed above.

[119] The identity type definition defines a plurality of attributes for an identity type into which the identity atomization and mapping system 224 will atomize identity information of that identity type. The identity type definition may further define an organization into which the identity atomization and mapping system 224 will place the plurality of attributes, such as ordered collection of elements representing the attributes, such as an ordered collection of one or more of fields, files, etc. representing the attributes.

[120] In one example, for a state driver's license, the identity type definition may define a plurality of attributes of the driver's license as including one or more of a name, a date of birth, an expiration date, a restriction, a Real ID compliant status, etc. Further, the definition for a drivers license may define a structured collection of files to contain the attributes as a follows:

```
/state/driverlicense/name.xml  
/state/driverlicense/DOB.xml
```

/state/driverlicense/EXP.xml
/state/driverlicense/restrictions.xml
/state/driverlicense/RealIDCompliant.xml

where each attribute is to be contained in a corresponding separate file, such as an XML file,
5 and stored in a corresponding location of a file directory structure.

[121] In another example, for a social media identity, such as a Twitter account, the identity type definition may define a plurality of attributes of the Twitter account as including one or more of a handle, a verified status, a profile image, a background image, and a description, etc. Further, the definition for the Twitter account may define a structured
10 collection of files to contain the attributes as a follows:

/electronic/socialmedia/Twitter/handle.xml
/electronic/socialmedia/Twitter/verified.xml
/electronic/socialmedia/Twitter/profileimage.xml
/electronic/socialmedia/Twitter/backgroundimage.xml
15 /electronic/socialmedia/Twitter/description.xml

where each attribute is to be contained in a separate corresponding file, such as an XML file, in a corresponding location of a file directory structure.

[122] In other examples, the identity type definition may define different types of structures to organize the attributes, such as in a plurality of structured fields of a single file
20 or data object, etc.

[123] At step 1610, the identity information may be atomized, i.e., separated, into a plurality of identity components, each corresponding to a different one of the identity attributes defined for that identity type. The identity information may be atomized by the atomization module 48. The atomization may process the identity information in a number
25 of different ways to identify the identity components, such as including one or more of capturing an image of the identity information, recognizing text in an image of the identity information, processing text of the identity information, etc. For example, a captured image of the identity information may be processed to recognize text in the image, and then the recognized text may be parsed and analyzed to identify the components corresponding to
30 the defined attributes for that identity type. The atomization may store the identified components in an organized data structure defined for that identity type. For example, for a state driver's license, the atomized components may be stored in a data structure such as the exemplary file structure discussed above.

[124] At step 1612, the atomized identity components may be distributed to storage locations. The atomized identity components may be distributed by the atomization module 48 to storage locations of the identity component module 260. The atomized identity components may be stored in a distributed fashion. Storing the atomized identity components in a distributed fashion, and accessing the components through the use of the index data maps as discussed below, may increase the security of the stored components and any transaction relying on the stored components, such as by increasing the difficulty of violating the integrity of the stored components. To store the atomized identity components in a distributed fashion, the identity component module 260 may include a plurality of distributed storage modules 276, and the atomized identity components for a set of identity information may be stored across the plurality of distributed storage modules 276. The atomized identity components may be stored across a plurality of separate storage modules 276 in a number of different ways, such as including one or more of storing each atomized component at a different distributed storage module 276, storing each atomized component across a plurality of different distributed storage modules 276, storing a plurality of atomized components across a plurality of different distributed storage modules 276, etc.

[125] The plurality of storage modules 276 may be structured in different ways. In embodiments, the plurality of different storage modules 276 may include locations at a corresponding plurality of different data storage systems. In such embodiments, each data storage system may be implemented using a separate computing system. In embodiments, the plurality of different data storage modules 276 may include locations at a corresponding plurality of different data storage devices. The data storage devices may include hard drives, flash memories, etc. In such embodiments, a plurality of the different data storage devices may be part of a single data storage system, part of a plurality of different data storage systems, or combinations thereof. In embodiments, the plurality of different data storage modules 276 may include multiple locations on a single data storage system or device. In such embodiments, each of the plurality of different data storage modules 276 may require a separate storage and/or access transaction, or set of steps, to store and/or access data at that data storage location. In one example, the plurality of different data storage modules 276 may be within a plurality of different partitions of, or behind other logical or physical boundaries within, the data storage system or device that would require such separate transactions or sets of steps.

[126] At step 1614, an index map to the stored atomized identity components is generated. The index map may be generated by the atomization module 248. The index

map includes a list of pointers to storage locations of the atomized identity components, for use by other components of the identity atomization and mapping system, such as the identity mapping module 264, to retrieve, and potentially create further data maps to, the identity components. The index map may be organized to enable access to selected sets of one or more pointers. For example, the index map may be organized one or more of as a searchable list, according to types of identity information, using a file directory structure of the stored components, etc. Optionally, a plurality of different index maps may be generated at step 1614, such as one or more index maps for each type of identity information having stored atomized identity components.

[127] At step 1616, the generated index map is stored. The index map may be stored by the atomization module 248 in the index module 256. The method ends at step 1618.

[128] In embodiments, a method of conducting an atomization process to atomize identity information for use in providing a compliant identity for a transaction may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 16.

[129] FIG. 17 depicts an embodiment of identity information that may undergo an atomization process of FIG. 16. In FIG. 17 the identity information is a state-issued driver's license. The driver's license has a plurality of attributes including a state that issued the driver's license, an identifier identifying the document as a driver's license, a picture of the individual for whom the driver's license was issued, a signature of the individual, a driver's license number, a class of the driver's license, a date of birth of the individual, an expiration date of the driver's license, a name of the individual, an address of the individual, restrictions on the driver's license, an issue date of the driver's license, a gender of the individual, an eye color of the individual, a height of the individual, a weight of the individual, a Real ID compliant status, etc. The atomization process may perform one or more of capturing an image of the driver's license, recognizing text and graphical elements in the captured image, processing the recognized text to identify the above attributes, creating identity components containing the attributes according a retrieved identity type definition for a driver's license, storing the atomized identity components in a distributed fashion, creating an index map to the stored components, and storing the index map.

[130] FIG. 18 depicts an embodiment of the identity definition module 252 storing exemplary identity type definitions for a plurality of different identity types, including for a driver's license such as depicted in FIG. 17. Each of the identity type definitions includes an attribute element 254 for each attribute of the corresponding identity type, having an

indication of what information that attribute includes, and optionally of a data structure in which to store the attribute as an atomized component. An attribute element 254 may be stored in a variety of formats, such as as a data field in a file, as a separate file, etc. In the depicted example, for the driver's license, the identity type definition includes elements 254
5 corresponding to each of the plurality of attributes of the driver's license, such as the state of issue, picture of the individual, signature of the individual, driver's license number, class of the driver's license, date of birth of the individual, expiration date of the driver's license, name of the individual, address of the individual, restrictions on the driver's license, issue date of the driver's license, gender of the individual, eye color of the individual, height of the
10 individual, weight of the individual, RealID compliant status, etc. The exemplary identity definition module also includes definitions for other identity types, such as for a passport, fingerprint, Twitter account, etc.

[131] FIG. 19 depicts embodiments of the identity index module 256 storing an exemplary index map and the identity component module 60 storing exemplary atomized identity
15 components produced by the atomization process, including as produced for the driver's license of FIG. 17. The index map includes a plurality of pointer elements 258 for each set of identity information, such as the driver's license, each pointer element 258 pointing, for a respective one of the attributes of the identity information, to the storage location in the identity component module 260 of a corresponding atomized identity component. A pointer
20 element 258 may be stored in a variety of formats, such as as a data field in a file, as a separate file, etc. The exemplary index map also includes pointer elements 258 to other atomized identity information, such as for a passport, for a fingerprint, for a Twitter account, etc. Alternatively, exemplary index maps may include pointers for only a single set of
25 identity information.

[132] The identity component module 260 stores at least one atomized identity component 262 for each of the attributes of the sets of identity information, such as the driver's license. An atomized identity component 262 may store the corresponding attribute in a variety of formats, such as as a data field in a file, as a separate file, etc. An atomized identity
30 component also may store the corresponding attribute as processed in a variety of ways, such as one or more of encoded, encrypted, etc. The atomized identity components may be distributed among the distributed storage modules 276 of the identity component module 260 in a variety of ways. On the right hand side and bottom of FIG. 19, several different possibilities for distribution of the plurality of identity components 262 across a plurality of

the distributed storage modules 276-1 ... 276-N of the identity component module are shown. In a first scenario shown at the right, each atomized identity component is stored in a different distributed storage module 276-1 ... 276-N. In a second scenario shown at the right, the atomized identity components for each particular set of identity information, such as for a particular identity document, are stored in a different distributed storage module 276-1 ... 276-N. In a third scenario shown at the bottom, the atomized identity components for a particular set of identity information, such as for a particular identity document, are each stored across a plurality of different distributed storage module 276-1 ... 276-N. Other distributions are also possible.

[133] In embodiments, the identity information may be atomized into and stored as identity components, and the index map generated and stored, according to the systems and methods of the distributed data architecture discussed above. For example, the identity information may correspond to a data element that is to be fragmented and distributed, with the atomization process fragmenting and distributing the identity information as atomized identity components, the atomized identity components corresponding to data fragments, and the generated index map corresponding to a data map.

[134] In this regard, the systems and methods of the distributed data architecture discussed above may be utilized for the atomization process, atomized identity components, index map, etc. The atomization process may incorporate the method of FIG. 6, or any individual one or any combination and/or order of the steps of the method of FIG. 6. For example, step 1610 through step 1616 of the method of FIG. 16 may be performed using the method of FIG. 6, or any individual one or any combination and/or order of the steps of the method of FIG. 6. An index map may be stored as a data map as depicted in FIG. 2, or using any individual one or any combination and/or order of the features of the data map of FIG. 2. For example, the index map may be stored with each pointer element 258 using the structure and/or fields of the map element 32, or using any individual one or any combination and/or order of the structure and/or fields of the map element 32, of the embodiment of FIG. 2. An atomized identity component may be stored a data fragment as depicted in FIG. 3, or using any individual one or any combination and/or order of the features of the data fragment of FIG. 3. For example, the atomized identity component may be stored using the structure and/or fields of the data fragment 24, or using any individual one or any combination and/or order of the structure and/or fields of the data fragment 24, of the embodiment of FIG. 3.

[135] In exemplary embodiments, an individual atomized identity component may correspond to an individual data fragment. That is, the atomization process may break the identity information into individual attributes, and store each of the individual attributes as a respective atomized identity component, with each atomized identify component being
5 treated as a data fragment in the distributed data architecture.

[136] In other exemplary embodiments, an individual atomized identity component may correspond to a plurality of data fragments. That is, the atomization process may break the identity information into individual attributes as a respective atomized identity component, with an individual atomized identity component being fragmented into a plurality of data
10 fragments in the distributed data architecture. For example, an atomized identity component corresponding to an individual attribute may be fragmented into a plurality of data fragments, the fragments distributed across a plurality of different distributed storage modules 276, and the index map may include a plurality of pointers for each atomized identity component, each pointer pointing to a corresponding fragment of the identity
15 component.

[137] FIG. 20 depicts an embodiment of a method 2000 of providing a compliant identity for a transaction based on atomized identity information. Embodiments of the method 2000 may be used to perform the providing of the compliant identity of step 1506 of the method 1500 of FIG. 15. The method may be performed by or involving components of the identity
20 atomization and mapping system 224 of FIG. 14, such as by one or more of the identity mapping module 264, mapping rules module 268, identity index module 256, identity component module 260 or interface module 272. The method provides a compliant identity having identity information sufficient to satisfy the requirements of the transaction, but protects from exposure identity information not required for the transaction, while also
25 enhancing the efficiency of the transaction by reducing transmission and processing costs that would otherwise be associated with providing surplus identity information. The method begins at step 2002.

[138] At step 2004, a request associated with a transaction is received. The request may be received by the identity mapping module 264. The request may be for identity
30 information sufficient to enable the transaction to proceed. For example, for a transaction to purchase alcohol, the request may be for identity information sufficient to prove that the purchasing individual is of the correct age to purchase alcohol. For a transaction to enter a secure passenger area at an airport, the request may be for REAL ID compliant identity information for the passenger seeking to enter the area. For a transaction to purchase an

item, the request may be for credit or banking account information for the purchasing individual.

[139] The request may be received by the identity mapping module 264 in a number of different ways. The request may be received by the communication interface 240 of the identity mapping module. The communication interface may include one or more components configured to receive the request. FIG. 21 depicts an embodiment of the communication interface 240 including at least one component configured to receive the request, including one or more of: a display 280, a radio-frequency identification (RFID) tag 284, an image sensor 288, a wireless electronic transmission transceiver 292, a wired electronic transmission receiver 296, an application programming interface (API) 300, or other component 304. In exemplary embodiments, the identity mapping module 264, or one or more components of the identity mapping module 264 such as the communication interface 240, may be located on the user device 228, and may receive the request as an input to the user device 228 during the transaction via the communication interface 240.

For example, the communication interface 240 may include in input element on a touchscreen embodiment of the display 280 configured to receive the request as a touch input, the radio-frequency identification (RFID) tag 284 configured to receive the request as an RF transmission, the image sensor 288 configured to take an image of an element representing the request, or the wireless or wired electronic transmission transceivers 292, 296 configured to receive transmissions including the request. Alternatively or additionally, in other exemplary embodiments, the identity mapping module 264, or one or more components of the identity mapping module 264 such as the communication interface 240, may be located on the identity server system 232, and may receive the request as part of the server system 232 being involved with the transaction. For example, the communication interface 240 may include the API 300 configured to receive a program input as the request.

[140] The request may take a variety of different forms. The request may include one or more of information identifying the type of transaction being conducted, information identifying the type of transaction being conducted, or information signaling the performance of a transaction, etc.

[141] At step 2006, a rule defining a compliant identity responsive to the request is retrieved. The rule may be retrieved by the identity mapping module 664 from the mapping rules module 68. The rule defines identity components needed to be included in the compliant identity.

[142] FIG. 22 depicts an embodiment of the mapping rules module 268 storing exemplary mapping rules for compliant identities for a plurality of different transaction types. Each of the rules is associated with a corresponding transaction type and includes one or more rule elements 270 identifying one or more atomized identity components necessary to compose a compliant identity for that transaction type. A rule element 270 may be stored in a variety of formats, such as as a data field in a file, as a separate file, etc. In the depicted example, for an alcohol purchase, a rule may include rule elements 270 identifying an image of the individual, an assertion that the individual is over the age to purchase alcohol, and an assertion that the utilized identity is valid. For a REAL ID compatible security check, a rule may include rule elements 270 identifying an image of the individual, a name of the individual, a date of birth of the individual, a country of origin of the passport of the individual, a passport number of the passport, a legal assertion that the passport is valid, a height of the individual, a weight of the individual, an eye color of the individual.

[143] To identify an identity component, a rule element 270 may identify one or more of the atomized identity component itself, the attribute corresponding to the atomized identity component, the set of identity information from which the attribute or atomized component is generated, etc.

[144] A rule may include rule elements 270 identifying components taken from or based on a single set of identity information, such as a single identity document. The creation of a compliant identity using such a rule may enable the provision of identity information from the identity document sufficient for conducting the transaction, but protect from exposure identity information from the identity document unnecessary for the transaction. For example, the components identified by the exemplary rule for an alcohol purchase depicted in FIG. 19 are taken from or based on a single state or country issued identity document, such as a state driver's license. The creation of a compliant identity based on such an exemplary rule provides information from the driver's license sufficient to enable the purchase of alcohol, but protects from exposure other identity information from the driver's license, such as, e.g., the address of the individual.

[145] A rule may include rule elements 270 identifying components taken from or based on multiple different sets of identity information, such as multiple different identity documents. For example, the components identified by the exemplary rule for a REAL ID compliant security check depicted in FIG. 19 may be taken from or based on multiple different state or country issued identity documents, such as a passport and a driver's license. The creation of a compliant identity using such a rule may leverage identity information from multiple

identity documents to provide of identity information sufficient for conducting the transaction, where any single one of the identity documents may have not contained sufficient information, but protect from exposure identity information from the identity documents unnecessary for the transaction. For example, the components identified by the exemplary rule for a REAL ID compatible security check depicted in FIG. 19 are taken from or based on both a passport and a driver's license (that is, the image of the individual, the name of the individual, the date of birth of the individual, the height of the individual, the weight of the individual, and the eye color of the individual are taken from the driver's license; and the country of origin of the passport, the passport number, and the legal assertion that the passport is valid are taken from or based on the passport). The creation of a compliant identity based on such an exemplary rule provides information sufficient to comply with REAL ID requirements, where one of the documents by itself may have been insufficient, such as in the case of using passport information to supplement, and make REAL ID compatible, a driver's license that is not REAL ID compatible, while protecting from exposure other identity information from the driver's license and passport.

[146] Returning to FIG. 20, at step 2008, the identified identity components, pointers to the identified identity components, or combinations thereof, are retrieved for composing the compliant identity. The pointers and identity components may be retrieved by the identity mapping module 264 from the identity index module 256 and the identity component module 260. Pointers to storage locations of the identity components in the identity component module 260 are retrieved from the identity index map of the identity index module 256. Identity components are retrieved by first retrieving pointers to the storage locations of the identity components in the identity component module 60 from the identity index map of the identity index module 256, and then retrieving the identity components from their storage locations in the identity component module 260 using the pointers.

[147] At step 2010, the compliant identity having information for the identified identity components is output. Providing the compliant identity provides identity information sufficient to satisfy the requirements of the transaction, while protecting from exposure identity information not required for the transaction, and also enhances the efficiency of conducting transactions by providing the sufficient identity information, while reducing transmission and processing costs that would otherwise be associated with surplus identity information not provided.

[148] The compliant identity may include elements for each of the atomized identity components identified by the rule being utilized, each element including the identity

component itself or the pointer to the identity component. Thus the compliant identity may include atomized identity components, pointers to identity components, or combinations thereof.

[149] The compliant identity may be output by the identity mapping module 264 to the transaction system 236. The compliant identity may be output by the identity mapping module 264 in a number of different ways. The compliant identity may be output by the communication interface 240 of the identity mapping module. For example, in exemplary embodiments where the identity mapping module 264, or one or more components of the identity mapping module 264 such as the communication interface 240, is located on the user device 228, the communication interface 240 may output the compliant identity as part of an output by the user device during the transaction. For example, the communication interface may include the display 280 configured to display the compliant identity, the RFID tag 284 configured to communicate signals representing the compliant identity, the wireless or wired electronic transmission transceiver 292, 296 configured to output signals representing the compliant identity. Alternatively or additionally, in other exemplary embodiments where the identity mapping module 264, or one or more components of the identity mapping module 264 such as the communication interface 240, is located on the identity server system 232, the communication interface 240 may output the compliant identity as part of the server system 232 being involved with the transaction. For example, the communication interface 240 may include the API 300 configured to provide a program output representing the compliant identity. The method ends at step 2012.

[150] In embodiments, a method of providing a compliant identity for a transaction based on atomized identity information may include only any subset of, or an alternative ordering of, the features depicted in or discussed above in regard to FIG. 20.

[151] FIG. 23 depicts an exemplary embodiment of a compliant identity output by the identity mapping module 64 corresponding to the exemplary rule discussed above for creating a REAL ID compatible identity. The compliant identity is output as a display by a display of the communication interface of the identity mapping module. The compliant identity includes textual and graphical elements corresponding to the image of the individual, name of the individual, date of birth of the individual, country of origin of the passport of the individual, passport number of the passport, legal assertion that the passport is valid, height of the individual, weight of the individual, eye color of the individual, a legal assertion that the presented identity is REAL ID compliant.

[152] The compliant identity may also include an encoding of some or all of the included identity components. Such an encoding may take the form of a bar code, two dimensional bar code, hash, etc. of the identity components. The encoding of the identity components may serve a variety of purposes, such as communication the identity components themselves, providing a means of checking the validity of the compliant identity, etc. In the embodiment of FIG. 20, the compliant identity includes a two dimensional bar code serving one or more of the above purposes.

[153] The identity atomization and mapping system may be encapsulated in a number of different ways within systems such as user, transaction and sever systems and/or devices.

FIG. 23 depicts embodiments of different encapsulations of the identity atomization and mapping system within user, transaction and sever systems and/or devices, and corresponding data flows for performance of embodiments of the method of FIG. 20. As shown at the top of FIG. 23, in embodiments, the identity mapping module 264 and mapping rules module 268 are located in the user device 228, and the identity index module 256 and identity component module 260 are located in one or more identity server systems 232. The identity mapping module 264 receives a request from the transaction system 236 related to a transaction, generates and sends to the mapping rules module 268 a request for a compliant identity definition for the type of transaction identified according to the request, receives at least one rule from the mapping rules module 268 defining a corresponding compliant identity, generates and sends to the identity index module 256 a request for pointers corresponding to components of the defined compliant identity, receives the pointers to the identity components from the identity index module 256, generates and sends to the identity component module 260 requests for the components of the defined compliant identity, receives the identity components from the identity component module 260, and outputs the compliant identity to the transaction system 236. For example, the user device 228 may receive the request and provide the compliant identity to and from the transaction system 36 using the communication module 240 of the identity mapping module 264, such as including one or more of the display, RFID tag, image sensor, wireless communication connection, wired communication connection, etc.

[154] Returning to step 1508 of the method of FIG. 15, the transaction may be conducted based on the provided identity by or involving components of the system for conducting transactions 220 of FIG. 13, such by one or more of the identity atomization and mapping system 224 or transaction system 236. Conducting the transaction based on the provided compliant identity may include one or more of verifying that the provided identity is indeed

compliant with an expected identity type, verifying that the individual identified by the provided identity is an authorized individual for the transaction, implementing any necessary steps to conduct the mechanics of the transaction, etc. For example, for security transactions, in which the user provides identity information to prove his or her identity to the transaction system, such as at an airport, building, venue, border, etc., to gain access to a restricted area, such as to a secure area, across a border, etc., the conducting the transaction may include one or more of verifying that the provided identity is compliant with an expected identity type for the type of restricted area being accessed (e.g., having components from a state or country issued license or passport for entering an airport security area, or from a building identification for entering a building, etc.), verifying that the individual identified by the provided identity is an authorized individual for the transaction (e.g., being a citizen of certain countries for entering an airport security area or crossing a border, a person on a list of authorized personnel for entering a building, etc.), or implementing any necessary steps to conduct the mechanics of the transaction (e.g., providing a signal indicating that access is allowed, or providing a signal enabling a machine to provide the access, such as open a gateway and/or release a lock, etc.). In another example, for financial transactions, in which the user provides identity information to conduct and/or authorize conducting of a financial transaction, such as purchasing an item, making a payment, exchanging currency, etc., such as either in person at a point-of-sale location or online, etc., the conducting the transaction may include one or more of verifying that the provided identity is compliant with an expected identity type for the type of financial transaction being conducted (e.g., having identity and/or account information components expected for that type of financial transaction), verifying that the individual identified by the provided identity is an authorized individual for the transaction (e.g., a person meeting certain legal or financial requirements for conducting that type of financial transaction), or implementing any necessary steps to conduct the mechanics of the transaction (e.g., providing a signal to transfer an item or ownership of an item, such as currency, securities, or other financial instruments, etc.).

[155] FIG. 24 also depicts at the bottom of the figure alternative encapsulations of the identity atomization and mapping system 224 and corresponding function and data flows within user, transaction and sever systems and/or devices. In the first alternative, the entire identity atomization and mapping system 224, including the identity mapping module 264, mapping rules module 268, identity index module 256, and identity component module 260, are located in one or more identity server systems 232. In such embodiments, the server system(s) 232 may receive the request and provide the compliant identity to and from the

transaction system 236 using a communication module 240 of the identity mapping module 264, such as including one or more of an API, wireless communication connection, wired communication connection, etc. In the second alternative, the identity mapping module 264, mapping rules module 268, and identity index module 256 are located on the user device 228, while the identity component module 60 is located in one or more identity server systems 232.

[156] The computer system depicted in FIG. 5 also may be used to implement the identity atomization and mapping system, transaction system, user system, identity provider system, and/or any individual one, subset, or all of the components thereof. The processor 164 executes program instructions stored in the non-transitory machine-readable storage medium 168 to perform the functionality of the component it is implementing as discussed herein. The communication circuit 172 can be controlled by the processor 164 to communicate with other devices, such as the identity atomization and mapping system, transaction system, user system, identity provider system, to perform the functionality of the component it is implementing as discussed herein. The optional other components 176 may include any further components required by the computer system 160 to perform this functionality.

[157] Additional embodiments of the identity atomization and mapping system 224, transaction system 236, user device 228, identity server system 232, and associated methods, as 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.

WHAT IS CLAIMED IS:

1. A method of providing identity information, the method comprising:
receiving, from a transaction system, a request regarding a transaction;
retrieving at least one rule defining a compliant identity based on a type of the transaction associated with the request, the at least one rule identifying a plurality of identity attributes;
retrieving an identity index map including a list of pointers to available identity components;
selecting pointers from the identity index map pointing to identity components corresponding to the plurality of identity attributes identified by the at least one rule;
outputting, to the transaction system, the compliant identity including information corresponding to each of the identity components, the information including for each of the identity components at least one of: the selected pointers to the identity component, or the identity component.
2. The method of claim 1, wherein the outputting the compliant identity outputs information for identity components corresponding to attributes of plurality of different sets of identity information.
3. The method of claim 1, wherein the outputting the compliant identity outputs information for identity components corresponding to at least some but less than a complete set of attributes of a set of identity information.
4. The method of claim 1, further comprising conducting the transaction based on the provided compliant identity.
5. The method of claim 4, wherein conducting the transaction includes at least one of:
conducting the transaction to gain access to a restricted area based on the provided compliant identity, or
conducting a financial transaction based on the provided compliant identity.

6. The method of claim 1, wherein receiving the request includes receiving the request at a communication interface from the transaction system.
7. The method of claim 1, wherein outputting the compliant identity includes outputting the compliant identity to the transaction at a communication interface.
8. The method of claim 1, wherein outputting the compliant identity includes at least one of: transmitting the compliant identity to the transaction system, or displaying the compliant identity to the transaction system.
9. The method of claim 1, wherein the request includes an identification of the type of the transaction.
10. The method of claim 1, further comprising determining the type of the transaction based on the request.
11. The method of claim 1, further comprising:
 - receiving identity information for a plurality of different sets of identity information of a plurality of different identity types;
 - retrieving stored identity type definitions for the plurality of different identity types, the identity type definitions defining a plurality of attributes of the corresponding identity type;
 - atomizing the identity information of each of the plurality of different sets of identity information into the available atomized identity information based on the plurality of identity type definitions.
12. The method of claim 1, further comprising:
 - receiving identity information for an individual;
 - processing the received identity information to identify an identity type of the identity information;
 - retrieving a stored identity type definition based on the identity type, the identity type definition defining a plurality of attributes of the identity type;
 - atomizing the identity information into a plurality of identity components based on the plurality of attributes of the identity type;

encrypting the identity information components;
distributing the identity information components to a plurality of different storage systems;
generating and storing an identity information index including a list of pointers to the distributed identity information components.

13. The method of claim 12, wherein receiving the identity information includes at least one of: capturing an image of the identity information, or receiving a transmission of the identity information.

14. A method of providing identity information, the method comprising:
receiving identity information for an individual;
processing the received identity information to identify an identity type of the identity information;
retrieving a stored identity type definition based on the identity type, the identity type definition including a plurality of attributes of the identity type;
atomizing the identity information into a plurality of identity information components based on the plurality of attributes of the identity type;
distributing the identity information components to a plurality of different storage locations; and
generating and storing an identity information index including a list of pointers to the distributed identity information components.

15. The method of claim 14, wherein receiving the identity information includes at least one of: capturing an image of the identity information, or receiving a transmission of the identity information.

16. The method of claim 14, further comprising:
receiving, from a transaction system, a request regarding a transaction;
retrieving at least one rule defining a compliant identity based on a type of the transaction associated with the request, the at least one rule identifying a plurality of identity attributes;
retrieving an identity index map including a list of pointers to available identity components;
selecting pointers from the identity index map pointing to identity components corresponding to the plurality of identity attributes identified by the at least one rule;

outputting, to the transaction system, the compliant identity including information corresponding to each of the identity components, the information including for each of the identity components at least one of: the selected pointers to the identity component, or the identity component.

17. The method of claim 16, wherein at least one of:

the outputting the compliant identity outputs information for identity components corresponding to attributes of plurality of different sets of identity information.

the outputting the compliant identity outputs information for identity components corresponding to at least some but less than a complete set of attributes of a set of identity information.

18. A system, comprising:

at least one non-transitory machine readable storage medium having program instructions, mapping rules, and an index map; and

at least one processor configured to execute the program instructions to perform a method of providing identity information, the method including:

receiving, from a transaction system, a request regarding a transaction;

retrieving at least one rule defining a compliant identity based on a type of the transaction associated with the request, the at least one rule identifying a plurality of identity attributes;

retrieving an identity index map including a list of pointers to available identity components;

selecting pointers from the identity index map pointing to identity components corresponding to the plurality of identity attributes identified by the at least one rule;

outputting, to the transaction system, the compliant identity including information corresponding to each of the identity components, the information including for each of the identity components at least one of: the selected pointers to the identity component, or the identity component.

19. The system of claim 18, wherein at least one of:

the outputting the compliant identity outputs information for identity components corresponding to attributes of plurality of different sets of identity information.

the outputting the compliant identity outputs information for identity components corresponding to at least some but less than a complete set of attributes of a set of identity information.

20. A system, comprising:

at least one non-transitory machine readable storage medium having program instructions; and

at least one processor configured to execute the program instructions to perform a method of providing identity information, the method including:

receiving identity information for an individual;

processing the received identity information to identify an identity type of the identity information;

retrieving a stored identity type definition based on the identity type, the identity type definition including a plurality of attributes of the identity type;

atomizing the identity information into a plurality of identity information components based on the plurality of attributes of the identity type;

distributing the identity information components to a plurality of different storage locations; and

generating and storing an identity information index including a list of pointers to the distributed identity information component.