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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 September 2016 (29.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/149806 A1

(51) International Patent Classification:
*G06Q 30/00* (2012.01) *H04L 12/16* (2006.01)
*G06Q 40/02* (2012.01)

(21) International Application Number:
PCT/CA2016/050305

(22) International Filing Date:
18 March 2016 (18.03.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/664,285 20 March 2015 (20.03.2015) US

(72) Inventors; and
(71) Applicants : **MAWJI, Ashif** [CA/CA]; 615 Magrath View NW, Edmonton, Alberta T6R 0H2 (CA). **CHAN, Leo M.** [CA/CA]; #30, 4755 Terwillegar Common, Edmonton, Alberta T6R 3V6 (CA). **CHRAPKO, Shane** [CA/CA]; #358 10654 82 Avenue, Edmonton, Alberta T6E 2A7 (CA). **MARSH, Stephen** [CA/CA]; 21670 Breadalbane Road, Dalkeith, Ontario K0B 1E0 (CA). **CHRAPKO, Evan V** [CA/CA]; #128 14-9977-178 Street, Edmonton, Alberta T5T 6J6 (CA).

(74) Agent: **SMART & BIGGAR**; 900-55 Metcalfe Street, Ottawa, Ontario K1P 6L5 (CA).

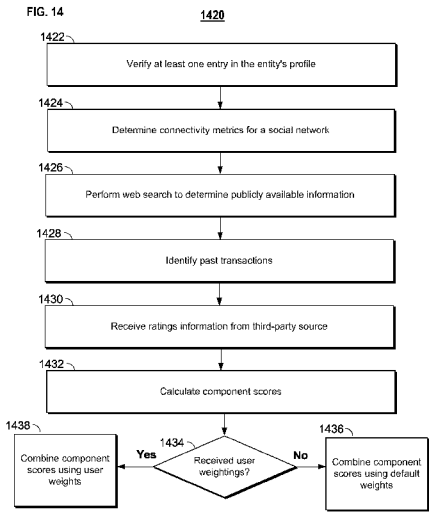
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP,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

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

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

(54) **Title**: CALCULATING A TRUST SCORE

FIG. 14




(57) **Abstract**: Systems, devices, and methods are described herein for calculating a trust score. The trust score may be calculated between entities including, but not limited to, human users, groups of users, organizations, businesses/corporations, and locations. A system trust score may be calculated for an entity by combining a variety of factors, including verification data, a network connectivity score, publicly available information, and/or ratings data. A peer trust score targeted from a first entity to a second entity may also be calculated based on the above factors. In some embodiments, the peer trust score may be derived from the system trust score for the target entity and may take into account additional factors, including social network connections, group/demographic info, and location data. Finally, a contextual trust score may be calculated between the first and second entities based on a type of transaction or activity to be performed between the two entities.

WO 2016/149806 A1

# CALCULATING A TRUST SCORE

## Cross-Reference to Related Application

This application claims priority to U.S. Utility Patent Application No. 14/664,285, filed March 20, 2015. The content of this application is hereby incorporated herein in its entirety.

## Background

Trust is an essential component to many social and business interactions, but trust can be both hard to measure and difficult to quantify. People typically looks towards a variety of different factors, experiences, and influences to determine how much to trust another party or entity in a transaction. For example, a potential customer deciding whether to dine at a particular restaurant may take into account how many times he or she has eaten at the restaurant, word of mouth from friends and family, and any ratings from online feedback sites. As another example, a bank may look up the credit score of a potential borrower as a measure of their financial responsibility when determining whether to issue a loan. Often, people can have wildly different preferences as to which factors are the most important in determining trust levels, and these preferences may change depending on the type and details of the transaction. Trust can also change over time, reflecting the cumulative experiences, transaction history, and recent trends between entities. A single negative event can destroy trust, and trust can also be rebuilt over time. All of the above considerations make "trust" an elusive measure to capture.

## Summary

Systems, devices, and methods are described herein for calculating a trust score. The trust score may be calculated between entities including, but not limited to, human users, groups of users, organizations, businesses/corporations, products/product lines, and/or locations. The trust score may reflect the trustworthiness, reputation, membership, status, and/or influence of the entity in a particular community or in relation to another entity. The trust score may take into account data from any suitable data sources, including, but not limited to, network connectivity information, social network information, credit score, available court data, transaction history, ratings/feedback data, group/demographics data, search engine data, or any publically available information. The trust score may also include

certain non-publically available information provided by the entities themselves (e.g., non-public transaction history, targeted ratings, etc.).

As used herein, a "system trust score" refers to a trust score calculated for an entity based on information available for the entity, without specific reference to another entity or activity/transaction. The system trust score may represent a base level of trustworthiness for the entity that does not take into account information about a specific activity/transaction. In some embodiments, the system trust score may be calculated based on publicly available information, such as verification data, a network connectivity score, and/or ratings data. As defined herein, a "network community" may include any collection or group of entities connected through a network, including, but not limited to a computer network or a social network. In some embodiments, a user may set an initial trust score as a minimum trust level. In these embodiments, the initial trust score may be retrieved and updated based on publicly available information in order to determine the system trust score. In some embodiments, the system trust score may be provided to an end user upon request without the end user having to identify themselves. For example, an end user may query the system trust scores of other entities, for example through a website or a mobile application, without having to sign into the website or mobile application or otherwise having to identify themselves.

As used herein, a "peer trust score" refers to a trust score calculated for a first entity in relation to a second entity. The peer trust score may take into account certain information that is specific to the first and second entity, such as specific transaction history between the first and second entity, number of common contacts/friends, etc. In some embodiments, the peer trust score may be derived from the system trust score and represent an update of the system trust score. For example, in some embodiments, the peer trust score may be calculated based on substantially the same data sources as the system trust score, where some components may be updated in order to further weight or take into account additional information that is specific to the first and second entity. In other embodiments, the peer trust score may be calculated independently from the system trust score and may be based on a different set of data sources than the system trust score.

As used herein, a "contextual trust score" refers to a trust score calculated for a first entity in relation to a specific activity or transaction. The contextual trust score may take into account certain information that is particular to the specific activity or transaction. In some embodiments, the contextual trust score may be derived from the system trust score or the peer trust score and represent an update of the system trust score or the peer trust score. For example, in some embodiments, the contextual trust score may be calculated based on

substantially the same data sources as the system trust score, where some components may be updated in order to take into account information that is particular to the activity/transaction. In other embodiments, the contextual trust score may be calculated based on a different set of data sources than the system trust score and the peer trust score. In some embodiments, the contextual trust score may be calculated by weighting data from different data sources based on the type of activity/transaction. For example, the trust score of a potential borrower who is seeking a mortgage from a bank may heavily weight the borrower's credit score and financial history rather than their level of connectivity in a social network. In this manner, the contextual trust score may be based on the same or similar data sources as the system trust score and/or the peer trust score, but with a different weighting to combine the data from the data sources. In some embodiments, specific details of the transactions may also affect the calculation of the contextual trust score. For instance, the contextual trust score for a friend borrowing $10 may focus more on social network connectivity (e.g., the number of friends they have in common, etc.), while the contextual trust score for a borrower seeking a $100K loan from the bank may focus more on financial factors. In some embodiments, the details of the transaction may affect the weighting of the combination of data from the data sources.

According to one aspect, a method for updating a trust score may comprise identifying paths from a first entity to a second entity, calculating a network connectivity score based on the identified paths, receiving data about the second entity from a remote source, and calculating a ratings score based on the received data from the remote source. A trust score for the second entity may be determined by combining the network connectivity score and the ratings score. An indication of an activity to be performed by the first entity and the second entity may be received, and the trust score may be updated based on the indication of the activity. In some embodiments, the first and second entity may be connected by a social network. In such embodiments, identifying paths from the first entity to the second entity may comprise identifying an intermediate entity in the social network that connects the first entity to the second entity. For example, the intermediate entity may be a common friend between a first user and a second user. Calculating the network connectivity score may comprise determining a number of mutual friends between the first entity and the second entity. For example, the network connectivity score may be assigned according to a graduated scale based on the number of mutual friends between the first entity and the second entity. The network connectivity score may also be calculated based on the number of

identified paths between the first and the second entity and whether the number of identified paths exceeds a certain threshold.

In some embodiments, the ratings data may be one of a credit score, criminal history data, financial transaction history data, and/or business reviews data. The ratings data may be combined with the network connectivity score according to a weighted sum in order to determine the trust score for the second entity. The weighted sum may be based on a default set of weights or based on user-assigned weights. The trust score for the second entity may then be updated based on the indication of the activity. For example, the indication of the activity may adjust the weighted sum such that a different weighted sum is used to calculate the trust score for the second entity.

In some embodiments, at least one of the first entity and the second entity is a human user. For instance, the trust score may be calculated between two users who are participating in a certain activity. In another embodiment, at least one of the first entity and the second entity may be a business. For example, the trust score between a user and a restaurant may be calculated in order to aid the user in determining whether to eat at the restaurant. In yet other embodiments, at least one of the first entity and the second entity may be a group of users or an organization. As an illustrative example, the second entity may be the Boy Scouts of America, and the trust score may be calculated between a first user and the Boy Scouts of America. In some embodiments, at least one of the first and second entity may be a product or an object. For instance, the first entity may be a first user, and the second entity may be a chainsaw, and a trust score may be calculated between the chainsaw and the first user. In this example, the trust score may take into account any user reviews of the chainsaw received from a third-party ratings source. In some embodiments, at least one of the first and second entity may be a location, city, region, nation, or any other geographic place. For instance, a trust score between a first user and a city, such as New York City, may be calculated. In this example, the trust score may take into account number of contacts that the first user has in New York City, traveler reviews received from third-party ratings sources, and/or and activities, transactions, or interactions that the first user has had with New York City.

In some embodiments, a decision related to the activity may be automatically resolved based, at least in part, on a calculated trust score. For instance, a bank may request the trust score of a potential borrower in order to evaluate the suitability of the borrower for a loan. Based on the updated trust score, the bank may automatically issue the loan, for example, if the trust score exceeds a certain threshold. In this manner, the system trust score, peer trust

score, and/or the contextual trust score can, either alone or in combination, form the basis for automatic decision making.

In some embodiments, at least one of the system, peer, and/or contextual trust score may include a confidence range. For example, each of the components from the data sources may comprise a confidence range (such as a variance or a standard deviation) indicating a level of uncertainty in the data, and the component scores may be combined to form one of the system, peer, and/or contextual trust score. Thus, the resulting trust score may be represented by a mean score and a confidence range, and in some embodiments, the confidence range may be represented by a mean and standard deviation.

Brief Description of the Drawings

The foregoing and other features and advantages will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, and in which:

FIG. 1 is a block diagram of an illustrative architecture for calculating a trust score;

FIG. 2 is another block diagram of an illustrative architecture for calculating a trust score;

FIG. 3 is a diagram of an illustrative tiered trust score system;

FIG. 4 is a block diagram of illustrative components that comprise a system trust score;

FIG. 5 is a diagram of an illustrative weighted combination of components that comprise a system trust score;

FIG. 6 is an illustrative graphical user interface displaying a trust score interface;

FIG. 7 is a graphical user interface displaying another illustrative trust score interface;

FIG. 8 is a table showing an illustrative graded scale for assigning component scores based on a metric;

FIG. 9 is an illustrative distribution for assigning component scores based on a metric;

FIG. 10 is a display of an illustrative network graph;

FIG. 11 is an illustrative data table for supporting connectivity determinations within a network community;

FIG. 12 is another illustrative data table for supporting connectivity determinations within a network community;

FIGs. 13A-E are illustrative processes for supporting connectivity determinations within a network community; and

FIG. 14 is an illustrative process for calculating a system trust score;

FIG. 15 is an illustrative process for calculating a peer trust score; and

FIG. 16 is an illustrative process for calculating a contextual trust score.

Detailed Description

To provide an overall understanding of the systems, devices, and methods described herein, certain illustrative embodiments will be described. It will be understood that the systems, devices, and methods described herein may be adapted and modified for any suitable application and that such other additions or modifications will not depart from the scope hereof.

FIG. 1 shows a block diagram of an architecture 100 for calculating a trust score in accordance with certain embodiments of the present disclosure. A user may utilize access application 102 to access application server 106 over communications network 104. For example, access application 102 may include a computer application such as a standard web browser or an app running on a mobile device. Application server 106 may comprise any suitable computer server, including a web server, and communication network 106 may comprise any suitable network, such as the Internet. Access application 102 may also include proprietary applications specifically developed for one or more platforms or devices. For example, access application 102 may include one or more instances of an Apple iOS, Android, or WebOS application or any suitable application for use in accessing application server 106 over communications network 104. Multiple users may access application server 106 via one or more instances of access application 102. For example, a plurality of mobile devices may each have an instance of access application 102 running locally on the respective devices. One or more users may use an instance of access application 102 to interact with application server 106.

Communication network 104 may include any wired or wireless network, such as the Internet, WiMax, wide area cellular, or local area wireless network. Communication network 104 may also include personal area networks, such as Bluetooth and infrared networks. Communications on communications network 104 may be encrypted or otherwise secured using any suitable security or encryption protocol.

Application server 106, which may include any network server or virtual server, such as a file or web server, may access data sources 108 locally or over any suitable network connection. Application server 106 may also include processing circuitry (e.g., one or more computer processors or microprocessors), memory (e.g., RAM, ROM, and/or hybrid types of

memory), and one or more storage devices (e.g., hard drives, optical drives, flash drives, tape drives). The processing circuitry included in application server 106 may execute a server process for calculating trust scores, while access application 102 executes a corresponding client process. The access application 102 may be executed by processing circuitry on a user's equipment, such as a computer or a mobile device (e.g., a cell phone, a wearable mobile device such as a smartwatch, etc.). The processing circuitry included in application server 106 and/or the processing circuitry that executes access application 102 may also perform any of the calculations and computations described herein in connection with calculating a trust score. In some embodiments, a computer-readable medium with computer program logic recorded thereon is included within application server 106. The computer program logic may calculate trust scores and may generate such trust scores for display on a display device. In some embodiments, application 102 and/or application server 106 may store a calculation date of a trust score and may generate for display the trust score together with a date of calculation.

Application server 106 may access data sources 108 over the Internet, a secured private LAN, or any other communications network. Data sources 108 may include one or more third-party data sources, such as data from third-party social networking services and third-party ratings bureaus. For example, data sources 108 may include user and relationship data (e.g., "friend" or "follower" data) from one or more of Facebook, MySpace, openSocial, Friendster, Bebo, hi5, Orkut, PerfSpot, Yahoo! 360, LinkedIn, Twitter, Google Buzz, Really Simple Syndication readers or any other social networking website or information service. Data sources 108 may also include data stores and databases local to application server 106 containing relationship information about users accessing application server 106 via access application 102 (e.g., databases of addresses, legal records, transportation passenger lists, gambling patterns, political and/or charity donations, political affiliations, vehicle license plate or identification numbers, universal product codes, news articles, business listings, and hospital or university affiliations).

Application server 106 may be in communication with one or more of data store 110, key-value store 112, and parallel computational framework 114. Data store 110, which may include any relational database management system (RDBMS), file server, or storage system, may store information relating to one or more network communities. For example, one or more of data tables 1100 (FIG. 11) may be stored on data store 110. Data store 110 may store identity information about users and entities in the network community, an identification of the nodes in the network community, user link and path weights, user

configuration settings, system configuration settings, and/or any other suitable information. There may be one instance of data store 110 per network community, or data store 110 may store information relating to a plural number of network communities. For example, data store 110 may include one database per network community, or one database may store information about all available network communities (e.g., information about one network community per database table).

Parallel computational framework 114, which may include any parallel or distributed computational framework or cluster, may be configured to divide computational jobs into smaller jobs to be performed simultaneously, in a distributed fashion, or both. For example, parallel computational framework 114 may support data-intensive distributed applications by implementing a map/reduce computational paradigm where the applications may be divided into a plurality of small fragments of work, each of which may be executed or re-executed on any core processor in a cluster of cores. A suitable example of parallel computational framework 114 includes an Apache Hadoop cluster.

Parallel computational framework 114 may interface with key-value store 112, which also may take the form of a cluster of cores. Key-value store 112 may hold sets of key-value pairs for use with the map/reduce computational paradigm implemented by parallel computational framework 114. For example, parallel computational framework 114 may express a large distributed computation as a sequence of distributed operations on data sets of key-value pairs. User-defined map/reduce jobs may be executed across a plurality of nodes in the cluster. The processing and computations described herein may be performed, at least in part, by any type of processor or combination of processors. For example, various types of quantum processors (e.g., solid-state quantum processors and light-based quantum processors), artificial neural networks, and the like may be used to perform massively parallel computing and processing.

In some embodiments, parallel computational framework 114 may support two distinct phases, a "map" phase and a "reduce" phase. The input to the computation may include a data set of key-value pairs stored at key-value store 112. In the map phase, parallel computational framework 114 may split, or divide, the input data set into a large number of fragments and assign each fragment to a map task. Parallel computational framework 114 may also distribute the map tasks across the cluster of nodes on which it operates. Each map task may consume key-value pairs from its assigned fragment and produce a set of intermediate key-value pairs. For each input key-value pair, the map task may invoke a user-defined map function that transmutes the input into a different key-value pair. Following the

map phase, parallel computational framework 114 may sort the intermediate data set by key and produce a collection of tuples so that all the values associated with a particular key appear together. Parallel computational framework 114 may also partition the collection of tuples into a number of fragments equal to the number of reduce tasks.

In the reduce phase, each reduce task may consume the fragment of tuples assigned to it. For each such tuple, the reduce task may invoke a user-defined reduce function that transmutes the tuple into an output key-value pair. Parallel computational framework 114 may then distribute the many reduce tasks across the cluster of nodes and provide the appropriate fragment of intermediate data to each reduce task.

Tasks in each phase may be executed in a fault-tolerant manner, so that if one or more nodes fail during a computation the tasks assigned to such failed nodes may be redistributed across the remaining nodes. This behavior may allow for load balancing and for failed tasks to be re-executed with low runtime overhead.

Key-value store 112 may implement any distributed file system capable of storing large files reliably. For example, key-value store 112 may implement Hadoop's own distributed file system (DFS) or a more scalable column-oriented distributed database, such as HBase. Such file systems or databases may include BigTable-like capabilities, such as support for an arbitrary number of table columns.

Although FIG. 1, in order to not over-complicate the drawing, only shows a single instance of access application 102, communications network 104, application server 106, data source 108, data store 110, key-value store 112, and parallel computational framework 114, in practice architecture 100 may include multiple instances of one or more of the foregoing components. In addition, key-value store 112 and parallel computational framework 114 may also be removed, in some embodiments. As shown in architecture 200 of FIG. 2, the parallel or distributed computations carried out by key-value store 112 and/or parallel computational framework 114 may be additionally or alternatively performed by a cluster of mobile devices 202 instead of stationary cores. In some embodiments, cluster of mobile devices 202, key-value store 112, and parallel computational framework 114 are all present in the network architecture. Certain application processes and computations may be performed by cluster of mobile devices 202 and certain other application processes and computations may be performed by key-value store 112 and parallel computational framework 114. In addition, in some embodiments, communication network 104 itself may perform some or all of the application processes and computations. For example, specially configured routers or

satellites may include processing circuitry adapted to carry out some or all of the application processes and computations described herein.

Cluster of mobile devices 202 may include one or more mobile devices, such as PDAs, cellular telephones, mobile computers, or any other mobile computing device. Cluster of mobile devices 202 may also include any appliance (e.g., audio/video systems, microwaves, refrigerators, food processors) containing a microprocessor (e.g., with spare processing time), storage, or both. Application server 106 may instruct devices within cluster of mobile devices 202 to perform computation, storage, or both in a similar fashion as would have been distributed to multiple fixed cores by parallel computational framework 114 and the map/reduce computational paradigm. Each device in cluster of mobile devices 202 may perform a discrete computational job, storage job, or both. Application server 106 may combine the results of each distributed job and return a final result of the computation.

FIG. 3 is a diagram 300 of a tiered trust score system in accordance with certain embodiments of the present disclosure. The system trust score 302, peer trust score 304, and contextual trust score 306 may represent a tiered trust system in which a user may inquire about the trustworthiness of a target entity either in isolation, in relation to another entity, and/or in relation to a specific activity/transaction. In some embodiments, the system trust score 302 may be calculated from a first set of data sources, (e.g., data sources 108 in FIG. 1). In some embodiments, the peer trust score 304 may be calculated as an update to system trust score 302 based on a second set of data sources, which may or may not be the same as the first set of data sources. Peer trust score 304 may or may not take into account additional data sources (e.g., data sources 108 in FIG. 1). In some embodiments, peer trust score 304 may also combine the data from the data sources according to a different weighting than the system trust score 302. In some embodiments, the contextual trust score 306 may be calculated as an update to either peer trust score 304 or system trust score 302. For example, the contextual trust score 306 may take into account different data sources (e.g., data sources 108 in FIG. 1) or may be based on the same data sources as system trust score 302 and/or peer trust score 304. In some embodiments, the contextual trust score 306 may combine data from the data sources according to a different weighting as system trust score 304 and/or peer trust score 304. Although the system trust score 302, peer trust score 304, and contextual trust score 306 are shown in FIG. 3 as a hierarchical system, each trust score may be calculated and presented either separately or together with the other trust scores.

The system trust score 302, peer trust score 304, and contextual trust score 306 may be represented in any suitable fashion. As an illustrative example, the system trust score 302,

peer trust score 304, and contextual trust score 306 may each be represented as a percentage out of 100 or as a numerical score out of 1000. In other embodiments, the system trust score 302, peer trust score 304, and contextual trust score 306 may be represented by different categories of trustworthiness (e.g., "reliable," "flaky," "honest," "fraudulent," etc.) or by a graphical scheme (e.g., a color spectrum representing level of trustworthiness). For ease of illustration, the trust score and component scores that comprise the trust scores will be discussed herein as numerical values. However, other methods of portraying a calculated trust score will be contemplated by those of ordinary skill in the art and will not depart from the scope hereof.

Each type of trust score may combine data from data sources according to a specific weighting. For instance, a weighting for a system trust score may be set as:

Data Verification – 5%
Network Connectivity – 20%
Credit Score – 15%
Court Data – 10%
Ratings/Feedback Data – 20%
Group/Demographics – 5%
Search Engine Mining – 5%
Transaction History – 20%

In some embodiments, a user may adjust these default weightings according to their preferences. For example, a user who values network analytics (e.g., how many friends we have in common) may assign a heavier weight, e.g., 25% to network connectivity, while lowering the weight of credit score to 10%. Conversely, a bank who cares very much about the credit score of its customers may assign a heavier weight to credit score and discount network connectivity.

The following is an example that illustrates one application of a system trust score 302, peer trust score 304, and contextual trust score 306. It will be understood that the following is provided for illustrative purposes only and that the systems, devices, and methods described herein may be further adapted or modified.

John sees an ad at ABC Restaurant for a short order cook and is trying to decide if he should apply. John opens an app on his mobile device and searches for ABC Restaurant. The app shows there are multiple matches to this search, but the nearest one is sorted to the top. After tapping on the correct restaurant, the app shows the ABC Restaurant profile page. The ABC Restaurant profile page includes a system trust score for ABC Restaurant, which is

calculated based in part on the ratings from three blogs. John taps to see more details and sees a list of most recent blogs from bloggers. By tapping on individual blogs, he can read the actual article. He can also tap on the bloggers to see their profile page in the app.

The system trust score for ABC Restaurant is also calculated based on previous transactions where ABC Restaurant was the employer. John taps to show a list of previous transactions, ratings of those transactions, and comments.

John taps on the social graph to see how he is connected to the restaurant through one or more networks (e.g., Facebook, MySpace, Twitter, LinkedIn, etc.). From the social graph he sees that Bob, the manager, is a friend of a friend. Based on the social graph data, the app updates the system trust score to calculate a peer trust score between John and ABC Restaurant. The peer trust score is higher than the system trust score to indicate the incremental increase in trustworthiness based on the connections between John and Bob the manager. The app also displays Bob's system trust score, calculated based on publicly available information and a default weighting, and Bob's peer trust score with respect to John, which also takes into account the social graph data.

John decides to apply for the job. After an interview, Bob the manager is deciding whether or not to hire John as a short order cook. Bob uses the app to search for John. There are multiple results for John, but Bob eventually finds him and taps on his entry. John's profile page displays his system trust score, calculated based on publicly available information (e.g., credit score, verification data, search engine mining, employment history, etc.) and a default weighting. Bob taps on the social graph to see how he is connected to John. He discovers that they are connected through a friend of a friend. The app updates John's system trust score based on the social network data to calculate a peer trust score between John and Bob, which is higher than John's system trust score to indicate the incremental increase in trustworthiness due to the connections between John and Bob. The app also shows average ratings from previous transactions where John was the employee. Bob taps to show a list of transactions, which can be ordered into chronological order and filtered by type of job. Bob also indicates to the app that he wishes to hire John as an employee. The app adjusts the weightings of the trust score to give a higher weight to the employee history rather than other components (such as credit score). The app uses the adjusted weightings to update the peer trust score to calculate the contextual trust score, which represents John's trustworthiness as a potential employee.

After reviewing the information in the app, Bob has decided to hire John. From John's profile page, he taps on the Action icon and chooses "Hire". The app prompts Bob to fill in

relevant information such as position, start date, annual salary, and vacation days per year. After confirming the data, the transaction appears in Bob's Notification list, with the status of "Waiting for John..." John receives a notification on his phone. He opens the app and sees a new transaction in his Notifications list. The app prompts John to confirm the details of his new job. John chooses to confirm, and Bob receives a notification that John has confirmed the transaction.

As illustrated in the above example, a user may request a system trust score for another entity, which may then be subsequently refined into a peer trust score based on information specific to the parties involved and into a contextual trust score based on the details of an activity/transaction to be performed by the parties.

FIG. 4 is a block diagram 400 of components 404-418 that comprise a system trust score 402 in accordance with certain embodiments of the present disclosure. The system trust score 402 may comprise a data verification component 404, a network connectivity component 406, a credit score component 408, a court data component 410, a ratings/feedback data component 412, a group/demographics component 414, a search engine mining component 416, and/or a transaction history component 418. The components 404-418 may be received either locally or through a suitable network connection from one or more data sources (e.g., data sources 108 in FIG. 1). It will be understood that components 404-418 are provided for illustrative purposes only and that the trust scores described herein may comprise more or fewer components than components 404-418 provided in FIG. 4.

Data verification component 404 may include data that verifies information associated with the target entity. In some embodiments, the data verification component 404 may include verification of contact information, including, but not limited to, email address, phone number, and/or mailing address. The data verification component may also comprise email, IM, and other messaging factors, such as frequency of messages, time of day of messages, depth of thread, or a review of threads for key transaction/activity types (e.g., loan, rent, buy, etc.). Data verification component 404 may take into account data from passport and/or other government IDs, tax return factors (e.g., a summary of a tax return to prove income), educational data (e.g., certificates of degree/diploma), group affiliation factors (e.g., invoices that prove membership to a group), achievements (e.g., proof of awards, medals, honorary citations, etc.), employment data (e.g., paystub data). The data verification component 404 may also incorporate facial recognition software to verify certain documents, such as IDs. In some embodiments, this facial recognition software may be used for subsequent verification of the user's identity. As an illustrative example, the data verification component 404 may

be used as a part of an airport scanning system to verify the user's identity. The data verification component 404 may comprise subcomponents such as data corresponding to the above illustrative examples, and as more subcomponents are verified, the higher the data verification component 404. The subcomponents may be combined to determine the data verification component 404 in any suitable manner, such as a weighted sum or the method discussed further below in relation to FIGs. 8 and 9. In some embodiments, verification of the data may be achieved by a document that proves the subject of the subcomponent (e.g., a tax return to prove income) or by peer verification. For instance, employment information may be vetted by peers connected to the target user, and as more peers positively vet the employment information, the higher the subcomponent score becomes. In some embodiments, the information may be deleted once verified. For example, images of passports/IDs may be deleted once the information contained therein is validated.

Network connectivity component 406 is discussed further below in relation to FIGs. 11-13. In some embodiments, the network connectivity component 406 may comprise data from a social network (e.g., Facebook, Twitter, Instagram, Pinterest, LinkedIn, etc.). For example, the network connectivity component 406 may take into account the number of connections, such Facebook "friends" that the target user has, those friends that comment or "like" the target user's posts, information on who the target user adds/removes as a friend, duration of the target user's friends (e.g., how long after the user adds them as a friend does the target user remove them as a friend), who the target user messages, which posts the target user shares, and length of tenure on the social network. For a peer trust score, such as peer trust score 304, the network connectivity component may take into account number of mutual friends, degree of separation, and number of paths from a first entity to the target entity.

Credit score component 408 may comprise any suitable financial information associated with the target entity, including income, checking/savings account information (number of accounts, value), and credit score information from one or more institutions. The credit score information may be received from any typical credit score agency, including, but not limited to, Transunion, Equifax, and Experian. Credit score factors may also be taken into account, such as number of credit accounts, credit utilization, length of credit history, number of late payments, etc. Other financial information taken into account may include prior loan and payment data, data on net worth or assets/liabilities, and information on any prior infractions. The various financial data may be combined using any suitable approach, including, but not limited to, the methods discussed below in relation to FIGs. 8 and 9.

Court data component 410 may include any data on activity associated with the target entity in a criminal or civil court. For example, court data component 410 may comprise data on how many cases involve the entity suing someone else and the type of suit, how many cases involve the target entity as the defendant, any criminal cases that may have a negative impact on trustworthiness, and the final holding/disposition of any concluded cases (e.g., acquitted, convicted, settled, etc.). Court data may be derived from any publicly available sources and from any available municipal, state, federal, or international court.

A ratings/feedback data component 412 may include any data that reflects a rating or feedback associated with the target entity. For instance, online rating sites such as Yelp may provide ratings information on various businesses. Any ratings of the target entity, information on volume, number of ratings, average rating, who rates the target entity, and whether the target entity responds to comments may be taken into account. In some embodiments, ratings data may be received from ratings institutions, such as the Better Business Bureau. Feedback data may include any positive or negative comments associated with the target entity. In some embodiments, feedback data may include comments made by peers in a social network. In some embodiments, the number and timing of ratings by other users or entities may be used to affect the ratings/feedback data component 412. For instance, a lack of negative feedback for a specified period of time may result in an increase (or decrease) in the ratings/feedback data component 412. Similarly, a lack of positive feedback for a specified period of time may result in a decrease (or increase) in the ratings/feedback data component 412.

Group/demographics component 414 may include information on group membership of the target entity or demographic information such as age, sex, race, location, etc. The group data may suggest an activity performed by the target entity. For instance, membership to a national sailing club may indicate an interest in sailing and boats. In some embodiments, a peer trust score may be adjusted to take into account the group/demographic component. For instance, the peer trust score for a target entity may be increased if a first entity and the target entity are both members of the same national sailing club. As another example, similarities in demographic information (age, sex, race, location, etc.) may indicate an incremental increase in trustworthiness between a first and the target entity, and the peer trust score for the target entity may be adjusted accordingly.

The search engine mining component 416 may include analytics performed on suitable search engines, such as Google or Yahoo. Websites/blogs/articles may be searched and scanned for entries about the target entry and a positive or negative sentiment may be

detected and stored for such entries. Number of articles, sentiment, timing of the articles, may indicate a positive or negative adjustment to the search engine mining component 416. In some embodiments, online shopping or auction websites such as eBay may be scanned for information associated with the target entity, such as rating and volume of transactions, feedback comments, number of bought/sold items, average value of items, and category of items (e.g., hardware, software, furniture, etc.).

Transaction history component 418 may comprise any information on past transactions associated with the target entity. Successful transactions or activities may be identified and positively impact the transaction history component score. For example, if I loan John $100 and he promptly pays me back, I may be more inclined to loan him money in the future. Transaction history data may be locally tracked and stored (e.g., by application 102 in FIG. 2) or may be received from remote sources (e.g., a bank or website). The transaction history data may factor in details of the transaction, such as amount of money, to whom, from whom, how many times, and/or success rate. Transaction/activity types may include, but are not limited to, loan/borrow funds or objects, buy from/sell to goods and services, financial transactions, dating, partner with (e.g., develop an alliance, start a new business with, invest with, etc.), becoming friends/acquaintances, rent to/from (including, e.g., renting cars, houses, hotel rooms, etc.), hire/work for (including, e.g., plumber, babysitter, etc.). The activity or transactions may include any number of parties, and each party may need to verify that they were in fact part of the activity/transaction. Each party may also rate their experience with the transaction/activity. Reminders for uncompleted activity/transactions may be automatically sent to a user or entity. For example, an email may be sent asking whether the user would like to provide feedback.

In some embodiments, the transactions history component 418 may comprise interactions between previous transactions in the transaction history between a first entity and a second entity. In this manner, processing circuitry may take into account elements of regret and forgiveness in determining a trust score. For example, a first transaction may correspond to an increase or decrease in a trust score, while a second, subsequent transaction related to the first transaction may result in an adjustment to the peer trust score in the opposite direction. The adjustment may be either a decrease in the trust score (e.g., regret or suspicion) or an increase in the trust score (e.g., forgiveness or redemption). As an illustrative example, a subject may have stolen a car in the past and be subsequently convicted of the theft and sentenced to serve 3 years in prison for the crime. The initial theft may serve to decrease the subject's trust score, reflecting the increased suspicion associated

with a known delinquent, while the subsequent conviction and sentence might serve to increase the subject's trust score, reflecting a level of redemption in the trustworthiness of the subject.

In some embodiments, the transactions that comprise the transactions history component 418 may be associated with an increase or decrease in a trust score over time. For example, a transaction may contribute to an initial increase in a trust score, and over time, the initial increase may decay until the trust score returns to an initial value. Similarly, a transaction may cause an initial decrease in a trust score, and over time, the initial decrease may decay until the trust score returns to an initial value.

In some embodiments, any one of the system, peer, or contextual trust score may also include a location component that takes into account a geographic location of an entity. For example, the location of an end user as determined by GPS coordinates or an address of a business may be incorporated into the calculation of a trust score. In some embodiments, a peer trust score may take into account the location of a first entity and a second entity and adjust the trust score accordingly. For instance, if a first user and a second user happen to be from the same hometown, then the peer trust scores may be increase to reflect this common information. In some embodiments, the location of the entity may provide an automatic increase/decrease in the trust score. For instance, a particular location may be known as a dangerous neighborhood, city, or region, and the trust scores of all entities located or associated with the dangerous location may be automatically decreased to reflect this danger. As an illustrative example, a user who travels to a country close to a known warzone may not be as comfortable trusting strangers in the country. The trust levels of others located in the same location as the user may be automatically decreased to reflect the increased suspicion. In some embodiments, the user may be traveling with his friends, as indicated by the high level of peer trust scores the user has with the plurality of people located around the user. Processing circuitry may determine that the user is surrounded by friends in any suitable manner, including explicit indications of friendship, common hometown, place of work, or any other common information. If the user is traveling to a dangerous location, but is traveling with friends, then the trust scores of other entities associated with the dangerous location may still be decreased, but they may be decreased by a smaller amount than if the user was not traveling with friends.

In some embodiments, any of the system, peer, and/or contextual trust scores may take into account biological responses of an end user. For instance, mobile devices may include cell phones, smart watches, heart rate monitors, and other wearable mobile devices

that can monitor one or more biological responses of an end user (e.g., heart rate, breathing rate, brain waves, sweat response, etc.). These detected biological responses of an end user, in conjunction with location information, may be used, in part, to determine a trust score. For example, an increase in heart rate may be an indication of anxiety, and may result in a decrease in trust score. The increase in heart rate may be caused by the user moving to a new location, in which case the trust score associated with that location may be decreased. The increase in heart rate may have been caused by a first user moving into close proximity with a second user, in which case the peer trust score with respect to the second user may be decreased, to reflect the increased anxiety that the first user feels around the second user.

FIG. 5 is a diagram 500 of a weighted combination 502 of components 504-518 that comprise a trust score in accordance with certain embodiments of the present disclosure. It will be understood that a trust score may comprise more or fewer components than components 504-518 and that components 504-518 are provided for illustrative purposes only. Weighted combination 502 comprises a data verification component 504, a network connectivity component 506, a credit score component 508, a court data component 510, a ratings/feedback data component 512, a group/demographics component 514, a search engine mining component 516, and a transaction history component 518. The components 504-518 may correspond respectively to data verification component 404, network connectivity component 406, credit score component 408, court data component 410, ratings/feedback data component 412, group/demographics component 414, search engine mining component 416, and transaction history component 418 depicted in FIG. 4. As shown in the illustrative example depicted in FIG. 5, the components 504-518 may be combined using a default weighting according to the following weights:

Data Verification – 5%
Network Connectivity – 20%
Credit Score – 15%
Court Data – 10%
Ratings/Feedback Data – 20%
Group/Demographics – 5%
Search Engine Mining – 5%
Transaction History – 20%

The components 504-518 may be combined using the above weights using a weighted sum. For example, each of the component 504-518 may be associated with a numerical component score. The weighted sum 502 may be calculated as:

$$S = \sum_{i=1}^{n} w_i c_i$$

wherein $w_i$ is the weighting as given by the default weighting above, and $c_i$ is the component score.

In some embodiments, the default weightings may be adjusted according to user-specified values. For example, as discussed above, users who care more about network connectivity may increase the weighting for the network connectivity component 506, and users who care less about financial responsibility may choose to decrease credit score component 508. In some embodiments, the default weightings above may be automatically adjusted, for example by application 102, to reflect a peer trust score or contextual trust score. For example, application 102 may detect that a first and second entity are entering into a financial transaction and may automatically adjust the weight for the credit score component 508 to reflect the importance of this component to the type of activity. Thus, the users may be provided with an contextual trust score that weights factors in a more relevant manner than the default weightings.

In some embodiments, at least one of the system trust score, peer trust score, and contextual trust score may be represented by a mean value and confidence band. The confidence band may represent a statistical variance in the calculated trust score. For example, each of the component scores may be associated with a mean score μ and a standard deviation σ based on how trustworthy the data source is. The mean and standard deviation for each of the component scores may be combined accordingly. As will be understood by those of ordinary skill in the art, the mean value of the total component scores may be represented by a sum of the mean value of each component score. The variance of two component scores together may be combined using the following equation:

$$V(A + B) = V(A) + V(B) + 2*Covar(A,B)$$

where V(A) is the variance (i.e., the square of the standard deviation) of component A, V(B) is the variance of component B, and Covar(A,B) is the covariance of components A and B.

FIG. 6 is a graphical user interface displaying a trust score interface 600 to a requesting user in accordance with certain embodiments of the present disclosure. Trust score interface 600 includes icon 602, initial score 604, transaction selector 606, transaction details field 608, additional transaction button 610, revised score icon 612, first profile score

614, second profile score 616, and calculate button 618. Although the trust score interface 600 is depicted in FIG. 6 in the context of a mobile device display screen, it will be understood that trust score interface 600 may be generated for display on any suitable display device.

Icon 602 and initial score 604 may graphically represent a first trust score of a target entity. Although icon 602 is depicted as a smiley face, it will be understood that any suitable graphical representation may be utilized to represent a relative trust level of the target entity. In some embodiments, the initial score 604 may be a system trust score for the target entity calculated using a default set of weights. In other embodiments, the initial score 604 may be a peer trust score calculated in relation to the user of the mobile app. For instance, the initial score 604 may represent a trust level that takes into account mutual friends of the requesting user and the target user.

The requesting user may use transaction selector 606 to indicate an activity/transaction to be performed with the target user. In some embodiments, transaction selector 606 may be optional, and no transaction is needed to calculate a revised score. Although transaction selector 606 is depicted as a dropdown box, any suitable input method (e.g., text input box, radio buttons, etc.) may be utilized to receive an indication of an activity/transaction from the requesting user. After an activity/transaction is selected, transaction details field 608 may provide further details or options. For example, if the requesting user indicates that the target entity wishes to request a loan, then the transaction details field 608 may include a field for indicating the amount of the loan. In this manner, a different weighting of components may be used for a $10 loan as opposed to a $100,000 loan. The requesting user may add an additional transaction using additional transaction button 610. In cases where multiple transactions are indicated, weightings for the multiple transactions may be averaged.

Revised score icon 612 may indicate a revised trust score calculated based on the information entered into transaction selector 606 and transaction details field 608. In some embodiments, the revised score icon 612 may reflect a peer trust score, for example, when a transaction is not selected in transaction selector 606. In other embodiments, the revised score icon 612 may reflect a contextual trust score calculated based on the activity/transaction and transaction details indicated in transaction selector 606 and transaction details field 608. The revised score icon 612 may include a graphical representation of the revised trust score, similar to icon 602. In the illustrative example depicted in FIG. 6, revised icon 612 includes

a smiley face to represent a relatively high revised score of 673. The requesting user may request a calculation using calculation button 618.

The first profile score 614 and the second profile score 616 may indicate one or more of a system trust score, peer trust score, and/or contextual trust score for the requesting user. As with icon 602 and icon 612, the first profile score 614 and second profile score 616 may include a graphical representation, such as a smiley face, of the respective trust score.

FIG. 7 is a graphical user interface displaying another trust score interface 700 in accordance with certain embodiments of the present disclosure. Trust score interface 700 includes weighting profile selector 702, weighting details field 704, weighting selector 706, first profile score 708, second profile score 710, and update weighting button 712.

As discussed above in relation to FIG. 5, a user may adjust weightings to user-specified value. These user-specified weightings may be saved as profiles which may be selected in weighting profile selector 702. Weighting details field 704 may reflect the details, such as weighting values of the various components, that correspond to the selected weighting profile. A user may further adjust the weightings using weighting selector 706. Although weighting profile selector 704 and weighting selector 706 are depicted in FIG. 7 as dropdown menus, any suitable selector may be utilized, including, but not limited to, text input boxes and/or radio buttons. The requesting user may update the weighting profile with the specified weights by selecting update weighting button 712.

In some embodiments, the weighting profiles may be stored, for example in data store 110 depicted in FIG. 1. These weighting profiles may form the basis for developing default weighting profiles specific to a particular transaction type. These default weighting profiles for specific transaction types may be suggested to other users, and the system, using processing circuitry, may use AI/machine learning techniques in order to monitor how users are adjusting the weighting profiles and automatically readjust the default weighting profiles for other users. By doing so, the system may improve response time and convenience for the end users, since they will not have to manually adjust their weighting profiles.

In some embodiments, the user may indicate an initial or base trust score factor that may be applied to every other user. At least one of the system trust score, peer trust score, and contextual trust score may then be calculated as updates to the initial or base trust score that the user has indicated. For example, each of the components discussed in relation with FIG. 4 may result in an increase or decrease in the indicated initial or base trust score. In some embodiments, the initial or base trust score may be determined by presenting a questionnaire or series of questions to the user to determine their general trust level towards

other entities. In some embodiments the user may specify different initial or base trust scores for different entities.

First profile score 708 and second profile score 710 may be substantially similar to first profile score 614 and second profile score 616 depicted in FIG. 6 and may indicate one or more of a system trust score, peer trust score, and/or contextual trust score for the requesting user.

FIG. 8 is a table 800 showing a graded scale for assigning component scores based on a metric in accordance with certain embodiments of the present disclosure. Table 800 depicts but one illustrative example for determining a component score or subcomponent score based on a measured metric 802. The illustrative example depicted in FIG. 8 uses number of friends in a social network as a measurable metric. Based on metric 802, component scores 804 and 806 may be assigned according to a graded scale. In the example depicted in FIG. 8, the component score 804 is depicted as a numerical score out of 1000, and the component score 806 is depicted as a percentage out of 100%. It will be understood that any suitable method for depicting the component score may be used. For example, the component score may be a represented by discrete categories (e.g., "very bad," "bad," "ok," "good," and "very good"). Furthermore, although the graded scale depicted in FIG. 8 shows only five steps, the graded scale may be divided into any suitable number of steps or categories.

According to the graded scale depicted in FIG. 8, the network component score (e.g., network connectivity score 406 in FIG. 4) may be assigned based on the number of friends the target entity has. For example, if the target entity has 306 friends, the network component score may be 600. In some embodiments, the network component score may comprise a combination of two or more subcomponent scores, wherein each subcomponent score is determined based on a grade scale similar to table 800. In some embodiments, the subcomponent scores may also be determined based on the method discussed below in relation to FIG. 9. In some embodiments, the subcomponent scores may be combined using an average or a weighted average. For example, the network component score may combine the number of friends and the number of "likes" a target user has received on their posts. The network component score may be weighted so that the number of friends accounts for 700/1000 of the potential network component score, and the number of "likes" accounts for 300/1000 of the potential network component score.

The metric 802 and the steps of the graded scale may be determined by a server, such as application server 106 depicted in FIG. 1. For example, the provider of the trust app may set the metric according to their proprietary algorithm. In some embodiments, the metric 802

may be adjusted by an entity such that the component score may be calculated according to the user's preferences. Although the metric 802 is discussed with respect to a network connectivity score, it will be understood that any of the components 404-418, or any other components, may be determined using a similar graded scale scheme.

FIG. 9 is a distribution 900 for assigning component scores based on a metric in accordance with certain embodiments of the present disclosure. Distribution 900 depicts one illustrative example for determining a component score or subcomponent score based on a measured metric 902. The illustrative example depicted in FIG. 9 uses number of friends in a social network as a measurable metric 904. An application (such as access application 102 in FIG. 1) or an application server (such as application server 106 in FIG. 1) may identify entities connected to a requesting user through a network. In some embodiments, the network may be a social network (such as Facebook) or a computer network (such as the Internet or a subset of the Internet). The application or application server may then determine or retrieve, for each identified user, information on the desired metric 904. In the illustrative example depicted in FIG. 9, the application or application server may identify all of the requesting user's friends and determine how many friends each of the user's friends has. Distribution 900 may be graphed based on the determined or retrieved information. In FIG. 9, distribution 900 is depicted as a Gaussian distribution, but it will be understood that any distribution may result from the determined or retrieved data. The distribution 900 may have a peak 912 at an average value $\mu$. For instance, most of a requesting user's friends may have an average value of $\mu = 500$ friends. The distribution 900 may be divided into regions 906, 908, 910, 914, 916, and 918 based on a standard deviation $\sigma$. For example, region 906 may represent a number of friends that is two standard deviations $\sigma$ below the average value $\mu$. Region 908 may represent a number of friends that is between two standard deviations $\sigma$ and one standard deviation $\sigma$ below the average value $\mu$. Region 910 may represent a number of friends that is less than one standard deviation $\sigma$ below the average value $\mu$. Region 914 may represent a number of friends that is between the average value $\mu$ and one standard deviation $\sigma$ above the average value $\mu$. Region 916 may represent a number of friends that is between one standard deviation $\sigma$ and two standard deviations $\sigma$ above the average value $\mu$. Finally, region 918 may represent a number of friends that is above two standard deviations $\sigma$ above the average value $\mu$.

The metric for the target user may fall into one of regions 906, 908, 910, 914, 916, and 918. As will be understood by those of ordinary skill in the art, regions 906 and 918 represent about 2.5% each of distribution 900, regions 908 and 916 represent about 13.5%

each of distribution 900, and regions 910 and 914 represent about 34% each of distribution 900. The application or application server may assign a component score depending on which of regions 906, 908, 910, 914, 916, and 918 the metric of the target user falls into. For instance, the component score for the target user may be relatively low if the metric falls within regions 906 or 918 and may be relatively high if the metric falls within regions 910 or 914. A graded scale, similar to table 800 depicted in FIG. 8, may be assigned to the regions 906, 908, 910, 914, 916, and 918.

FIG. 10 is a display of a network graph 1000 in accordance with certain embodiments of the present disclosure. Network graph 1000 includes source node 1002, target node 1004, intermediate node 1006, and paths 1008 and 1010. The network graph 1000 may be generated for display on any suitable display device and in any suitable interface, such as the interfaces 600 and 700 depicted in FIGs. 6 and 7. As defined herein, a "node" may include any user terminal, network device, computer, mobile device, access point, or any other electronic device. In some embodiments, a node may also represent an individual human being, entity (e.g., a legal entity, such as a public or private company, corporation, limited liability company (LLC), partnership, sole proprietorship, or charitable organization), concept (e.g., a social networking group), animal, or inanimate object (e.g., a car, aircraft, or tool).

The network graph 1000 may represent a visualization of a network that connects a requesting entity, depicted by source node 1002, and a target entity, depicted by target node 1004. One or more intermediate nodes, such as intermediate node 1006, may also be displayed, as well as paths 1008 that connect nodes 1002, 1004, and 1006. In some embodiments, a dominant path 1010 may be displayed and visually distinguished from other paths 1008. The dominant path 1010 may be determined using any suitable algorithm. For example, the dominant path 1010 may represent the shortest-length path from source node 1002 to source node 1004. In other embodiments, the dominant path 1010 may represent a path through specific intermediate nodes, such as nodes with relatively high trust values. For example, a longer path from node 1002 through node 1006 to node 1004 may have higher trust at each link of the path than the shorter path 1010.

In some embodiments, each of the nodes 1002, 1004, and 1006 may include images, text, or both, such as a profile picture associated with the entity depicted by the nodes. In some embodiments, the network graph 1000 may be generated for display in a scrollable display, wherein a user may scroll and zoom the network graph 1000 to see more and less nodes as desired.

FIGs. 11-13 describe illustrative methods for calculating a network component score, such as network connectivity component 406 depicted in FIG. 4. Connectivity may be determined, at least in part, using various graph traversal and normalization techniques described in more detail below.

In an embodiment, a path counting approach may be used where processing circuitry is configured to count the number of paths between a first node $n_1$ and a second node $n_2$ within a network community. A connectivity rating $R_{n1n2}$ may then be assigned to the nodes. The assigned connectivity rating may be proportional to the number of subpaths, or relationships, connecting the two nodes, among other possible measures. Using the number of subpaths as a measure, a path with one or more intermediate nodes between the first node $n_1$ and the second node $n_2$ may be scaled by an appropriate number (e.g., the number of intermediate nodes) and this scaled number may be used to calculate the connectivity rating.

In some embodiments, weighted links are used in addition to or as an alternative to the subpath counting approach. Processing circuitry may be configured to assign a relative user weight to each path connecting a first node $n_1$ and a second node $n_2$ within a network community. A user connectivity value may be assigned to each link. For example, a user or entity associated with node $n_1$ may assign user connectivity values for all outgoing paths from node $n_1$. In some embodiments, the connectivity values assigned by the user or entity may be indicative of that user or entity's trust in the user or entity associated with node $n_2$. The link values assigned by a particular user or entity may then be compared to each other to determine a relative user weight for each link.

The relative user weight for each link may be determined by first computing the average of all the user connectivity values assigned by that user (i.e., the out-link values). If $t_i$ is the user connectivity value assigned to link $i$, then the relative user weight, $w_i$, assigned to that link may be given in accordance with:

$$w_i = 1 + (t_i - \bar{t}_i)^2 \quad (1)$$

To determine the overall weight of a path, in some embodiments, the weights of all the links along the path may be multiplied together. The overall path weight may then be given in accordance with:

$$w_{path} = \Pi(w_i) \quad (2)$$

The connectivity value for the path may then be defined as the minimum user connectivity value of all the links in the path multiplied by the overall path weight in accordance with:

$$t_{path} = w_{path} \times t_{\min} \quad (3)$$

To determine path connectivity values, in some embodiments, a parallel computational framework or distributed computational framework (or both) may be used. For example, in one embodiment, a number of core processors implement an Apache Hadoop or Google MapReduce cluster. This cluster may perform some or all of the distributed computations in connection with determining new path link values and path weights.

The processing circuitry may identify a changed node within a network community. For example, a new outgoing link may be added, a link may be removed, or a user connectivity value may have been changed. In response to identifying a changed node, in some embodiments, the processing circuitry may re-compute link, path, and weight values associated with some or all nodes in the implicated network community or communities.

In some embodiments, only values associated with affected nodes in the network community are recomputed after a changed node is identified. If there exists at least one changed node in the network community, the changed node or nodes may first undergo a prepare process. The prepare process may include a "map" phase and "reduce" phase. In the map phase of the prepare process, the prepare process may be divided into smaller sub-processes which are then distributed to a core in the parallel computational framework cluster. For example, each node or link change (e.g., tail to out-link change and head to in-link change) may be mapped to a different core for parallel computation. In the reduce phase of the prepare process, each out-link's weight may be determined in accordance with equation (1). Each of the out-link weights may then be normalized by the sum of the out-link weights (or any other suitable value). The node table may then be updated for each changed node, its in-links, and its out-links.

After the changed nodes have been prepared, the paths originating from each changed node may be calculated. Once again, a "map" and "reduce" phase of this process may be defined. During this process, in some embodiments, a depth-first search may be performed of the node digraph or node tree. All affected ancestor nodes may then be identified and their paths recalculated.

In some embodiments, to improve performance, paths may be grouped by the last node in the path. For example, all paths ending with node $n_1$ may be grouped together, all paths ending with node $n_2$ may be grouped together, and so on. These path groups may then be stored separately (e.g., in different columns of a single database table). In some embodiments, the path groups may be stored in columns of a key-value store implementing an HBase cluster (or any other compressed, high performance database system, such as BigTable).

In some embodiments, one or more threshold functions may be defined. The threshold function or functions may be used to determine the maximum number of links in a path that will be analyzed in a connectivity determination or connectivity computation. Threshold factors may also be defined for minimum link weights, path weights, or both. Weights falling below a user-defined or system-defined threshold may be ignored in a connectivity determination or connectivity computation, while only weights of sufficient magnitude may be considered.

In some embodiments, a user connectivity value may represent the degree of trust between a first node and a second node. In one embodiment, node $n_1$ may assign a user connectivity value of $l_1$ to a link between it and node $n_2$. Node $n_2$ may also assign a user connectivity value of $l_2$ to a reverse link between it and node $n_1$. The values of $l_1$ and $l_2$ may be at least partially subjective indications of the trustworthiness of the individual or entity associated with the node connected by the link. A user (or other individual authorized by the node) may then assign this value to an outgoing link connecting the node to the individual or entity. Objective measures (e.g., data from third-party ratings agencies or credit bureaus) may also be used, in some embodiments, to form composite user connectivity values indicative of trust. The subjective, objective, or both types of measures may be automatically harvested or manually inputted for analysis.

FIG. 11 shows data tables 1100 used to support the connectivity determinations for calculating a network component score in accordance with certain embodiments of the present disclosure. One or more of tables 1100 may be stored in, for example, a relational database in data store 110 (FIG. 1). Table 1102 may store an identification of all the nodes registered in a network community. A unique identifier may be assigned to each node and stored in table 1102. In addition, a string name may be associated with each node and stored in table 1102. As described above, in some embodiments, nodes may represent individuals or entities, in which case the string name may include the individual or person's first and/or last name, nickname, handle, or entity name.

Table 1104 may store user connectivity values. In some embodiments, user connectivity values may be assigned automatically by the system (e.g., by application server 106 (FIG. 1)). For example, application server 106 (FIG. 1) may monitor all electronic interaction (e.g., electronic communication, electronic transactions, or both) between members of a network community. In some embodiments, a default user connectivity value (e.g., the link value 1) may be assigned initially to all links in the network community. After electronic interaction is identified between two or more nodes in the network community,

user connectivity values may be adjusted upwards or downwards depending on the type of interaction between the nodes and the result of the interaction. For example, each simple email exchange between two nodes may automatically increase or decrease the user connectivity values connecting those two nodes by a fixed amount. More complicated interactions (e.g., product or service sales or inquiries) between two nodes may increase or decrease the user connectivity values connecting those two nodes by some larger fixed amount. In some embodiments, user connectivity values between two nodes may be increased unless a user or node indicates that the interaction was unfavorable, not successfully completed, or otherwise adverse. For example, a transaction may not have been timely executed or an email exchange may have been particularly displeasing. Adverse interactions may automatically decrease user connectivity values while all other interactions may increase user connectivity values (or have no effect). In addition, user connectivity values may be automatically harvested using outside sources. For example, third-party data sources (such as ratings agencies and credit bureaus) may be automatically queried for connectivity information. This connectivity information may include completely objective information, completely subjective information, composite information that is partially objective and partially subjective, any other suitable connectivity information, or any combination of the foregoing.

In some embodiments, user connectivity values may be manually assigned by members of the network community. These values may represent, for example, the degree or level of trust between two users or nodes or one node's assessment of another node's competence in some endeavor. User connectivity values may include a subjective component and an objective component in some embodiments. The subjective component may include a trustworthiness "score" indicative of how trustworthy a first user or node finds a second user, node, community, or subcommunity. This score or value may be entirely subjective and based on interactions between the two users, nodes, or communities. This manual user connectivity score may “override” one or more of the system trust score, peer trust score, or contextual trust score. When a user “overrides” one of the above trust scores with a manual trust score, the user-specified trust score may be provided concurrently with, or instead of, the overridden trust score.

In some embodiments, a system administrator may override one or more of the system trust score, peer trust score, or contextual trust score. For example, a system administrator may override a system trust score of an entity to take into account recent trends or events. When a trust score is overridden by the system administrator, the administrator’s trust score

may be provided concurrently with, or instead of, the overridden trust score. When the overridden trust score reaches a specified range or threshold of the administrator's trust score, the system may automatically revert back to the overridden trust score. As an illustrative example, the system administrator may decrease a system trust score of an entity that has taken negative public attention in the news. The overridden trust score will continue to be calculated by the system and will gradually reflect the negative public attention of the entity. When the overridden trust score reaches within a certain range of the administrator's trust level (e.g., within 10%), then the system will automatically revert back to the calculated score. In some embodiments, the administrator's trust score will be provided to a user with a notification that the score was overridden and/or a reason why the trust score was overridden.

Table 1104 may store an identification of a link head, link tail, and user connectivity value for the link. Links may or may not be bidirectional. For example, a user connectivity value from node $n_1$ to node $n_2$ may be different (and completely separate) than a link from node $n_2$ to node $n_1$. Especially in the trust context described above, each user can assign his or her own user connectivity value to a link (i.e., two users need not trust each other an equal amount in some embodiments).

Table 1106 may store an audit log of table 1104. Table 1106 may be analyzed to determine which nodes or links have changed in the network community. In some embodiments, a database trigger is used to automatically insert an audit record into table 1106 whenever a change of the data in table 1104 is detected. For example, a new link may be created, a link may be removed, or a user connectivity value may be changed. This audit log may allow for decisions related to connectivity values to be made prospectively (i.e., before an anticipated event). Such decisions may be made at the request of a user, or as part of an automated process. This prospective analysis may allow for the initiation of a transaction (or taking of some particular action) in a fluid and/or dynamic manner. After such a change is detected, the trigger may automatically create a new row in table 1106. Table 1106 may store an identification of the changed node, and identification of the changed link head, changed link tail, and the user connectivity value to be assigned to the changed link. Table 1106 may also store a timestamp indicative of the time of the change and an operation code. In some embodiments, operation codes may include "insert," "update," or "delete" operations, corresponding to whether a link was inserted, a user connectivity value was changed, or a link was deleted, respectively. Other operation codes may be used in other embodiments.

FIG. 12 shows data structure 1210 used to support the connectivity determinations of the present disclosure. In some embodiments, data structure 1210 may be stored using key-value store 112 (FIG. 1), while tables 1200 are stored in data store 110 (FIG. 1). As described above, key-value store 112 (FIG. 1) may implement an HBase storage system and include BigTable support. Like a traditional relational database management system, the data shown in FIG. 12 may be stored in tables. However, the BigTable support may allow for an arbitrary number of columns in each table, whereas traditional relational database management systems may require a fixed number of columns.

Data structure 1210 may include node table 1212. In the example shown in FIG. 12, node table 1212 includes several columns. Node table 1212 may include row identifier column 1214, which may store 64-bit, 128-bit, 256-bit, 512-bit, or 1024-bit integers and may be used to uniquely identify each row (e.g., each node) in node table 1212. Column 1216 may include a list of all the incoming links for the current node. Column 1218 may include a list of all the outgoing links for the current node. Column 1220 may include a list of node identifiers to which the current node is connected. A first node may be connected to a second node if outgoing links may be followed to reach the second node. For example, for A -> B, A is connected to B, but B may not be connected to A. Node table 1212 may also include one or more "bucket" columns 1222. These columns may store a list of paths that connect the current node to a target node. As described above, grouping paths by the last node in the path (e.g., the target node) may facilitate connectivity computations. As shown in FIG. 12, in some embodiments, to facilitate scanning, bucket column names may include the target node identifier appended to the end of the "bucket:" column .

FIGs. 13A-13E show illustrative processes for determining the connectivity of nodes within a network community. The processes depicted in FIGs. 13A-13E may be used to determine a network component score, such as network connectivity component 406 depicted in FIG. 4. FIG. 13A shows process 1300 for updating a connectivity graph (or any other suitable data structure) associated with a network community. As described above, in some embodiments, each network community is associated with its own connectivity graph, digraph, tree, or other suitable data structure. In other embodiments, a plurality of network communities may share one or more connectivity graphs (or other data structure).

In some embodiments, the processes described with respect to FIGs. 13A-13E may be executed to make decisions prospectively (i.e., before an anticipated event). Such decisions may be made at the request of a user, or as part of an automated process. This prospective analysis may allow for the initiation of a transaction (or taking of some particular action) in a

fluid and/or dynamic manner. In some embodiments, processing circuitry may anticipate an increase or decrease in a trust score as a result of making a certain decision. The processing circuitry may provide an alert to an end user, for example through one of user interface 600 or 700, that indicates to the end user that the trust score of the end user will increase/decrease as a result of the decision. In some embodiments, the prospective decision may also be made, either manually or automatically, based on the potential increase/decrease in trust score as a result of the decision. For example, processing circuitry may automatically make a prospective decision if the decision would result in an increase/decrease in a trust score within a certain threshold. In this manner, prospective decisions, whether made automatically or manually, may take into account a risk tolerance or risk preference of an end user.

At step 1302, a determination is made whether at least one node has changed in the network community. As described above, an audit record may be inserted into table 1106 (FIG. 11) after a node has changed. By analyzing table 1106 (FIG. 11), a determination may be made (e.g., by application server 106 of FIG. 1) that a new link has been added, an existing link has been removed, or a user connectivity value has changed. If, at step 1304, it is determined that a node has changed, then process 1300 continues to step 1310 (shown in FIG. 13B) to prepare the changed nodes, step 1312 (shown in FIG. 13C) to calculate paths originating from the changed nodes, step 1314 (shown in FIG. 13D) to remove paths that go through a changed node, and step 1316 (shown in FIG. 13E) to calculate paths that go through a changed node. It should be noted that more than one step or task shown in FIGS. 13B, 13C, 13D, and 13E may be performed in parallel using, for example, a cluster of cores. For example, multiple steps or tasks shown in FIG. 13B may be executed in parallel or in a distributed fashion, then multiple steps or tasks shown in FIG. 13C may be executed in parallel or in a distributed fashion, then multiple steps or tasks shown in FIG. 13D may be executed in parallel or in a distributed fashion, and then multiple steps or tasks shown in FIG. 13E may be executed in parallel or in a distributed fashion. In this way, overall latency associated with process 1300 may be reduced.

If a node change is not detected at step 1304, then process 1300 enters a sleep mode at step 1306. For example, in some embodiments, an application thread or process may continuously check to determine if at least one node or link has changed in the network community. In other embodiments, the application thread or process may periodically check for changed links and nodes every $n$ seconds, where $n$ is any positive number. After the paths are calculated that go through a changed node at step 1316 or after a period of sleep at step

1306, process 1300 may determine whether or not to loop at step 1308. For example, if all changed nodes have been updated, then process 1300 may stop at step 1318. If, however, there are more changed nodes or links to process, then process 1300 may loop at step 1308 and return to step 1304.

In practice, one or more steps shown in process 1300 may be combined with other steps, performed in any suitable order, performed in parallel (e.g., simultaneously or substantially simultaneously), or removed.

FIGS. 13B-13E each include processes with a "map" phase and "reduce" phase. As described above, these phases may form part of a map/reduce computational paradigm carried out by parallel computational framework 114 (FIG. 1), key-value store 112 (FIG. 1), or both. As shown in FIG. 13B, in order to prepare any changed nodes, map phase 1320 may include determining if there are any more link changes at step 1322, retrieving the next link change at step 1340, mapping the tail to out-link change at step 1342, and mapping the head to in-link change at step 1344.

If there are no more link changes at step 1322, then, in reduce phase 1324, a determination may be made at step 1326 that there are more nodes and link changes to process. If so, then the next node and its link changes may be retrieved at step 1328. The most recent link changes may be preserved at step 1330 while any intermediate link changes are replaced by more recent changes. For example, the timestamp stored in table 1106 (FIG. 11) may be used to determine the time of every link or node change. At step 1332, the average out-link user connectivity value may be calculated. For example, if node $n_1$ has eight out-links with assigned user connectivity values, these eight user connectivity values may be averaged at step 1332. At step 1334, each out-link's weight may be calculated in accordance with equation (1) above. All the out-link weights may then be summed and used to normalize each out-link weight at step 1336. For example, each out-link weight may be divided by the sum of all out-link weights. This may yield a weight between 0 and 1 for each out-link. At step 1338, the existing buckets for the changed node, in-links, and out-links may be saved. For example, the buckets may be saved in key-value store 112 (FIG. 1) or data store 110 (FIG. 1). If there are no more nodes and link changes to process at step 1326, the process may stop at step 1346.

As shown in FIG. 13C, in order to calculate paths originating from changed nodes, map phase 1348 may include determining if there are any more changed nodes at step 1350, retrieving the next changed node at step 1366, marking existing buckets for deletion by mapping changed nodes to the NULL path at step 1368, recursively generating paths by

following out-links at step 1370, and if the path is a qualified path, mapping the tail to the path. Qualified paths may include paths that satisfy one or more predefined threshold functions. For example, a threshold function may specify a minimum path weight. Paths with path weights greater than the minimum path weight may be designated as qualified paths.

If there are no more changed nodes at step 1350, then, in reduce phase 1352, a determination may be made at step 1354 that there are more nodes and paths to process. If so, then the next node and its paths may be retrieved at step 1356. At step 1358, buckets may be created by grouping paths by their head. If a bucket contains only the NULL path at step 1360, then the corresponding cell in the node table may be deleted at step 1362. If the bucket contains more than the NULL path, then at step 1364 the bucket is saved to the corresponding cell in the node table. If there are no more nodes and paths to process at step 1356, the process may stop at step 1374.

As shown in FIG. 13D, in order to remove paths that go through a changed node, map phase 1376 may include determining if there are any more changed nodes at step 1378 and retrieving the next changed node at step 1388. At step 1390, the "bucket:" column in the node table (e.g., column 1222 of node table 1212 (both of FIG. 12)) corresponding to the changed node may be scanned. For example, as described above, the target node identifier may be appended to the end of the "bucket:" column name. Each bucket may include a list of paths that connect the current node to the target node (e.g., the changed node). At step 1392, for each matching node found by the scan and the changed node's old buckets, the matching node may be matched to a (changed node, old bucket) deletion pair.

If there are no more changed nodes at step 1378, then, in reduce phase 1380, a determination may be made at step 1384 that there are more node and deletion pairs to process. If so, then the next node and its deletion pairs may be retrieved at step 1384. At step 1386, for each deletion pair, any paths that go through the changed node in the old bucket may be deleted. If there are no more nodes and deletion pairs to process at step 1382, the process may stop at step 1394.

As shown in FIG. 13E, in order to calculate paths that go through a changed node, map phase 1396 may include determining if there are any more changed nodes at step 1398 and retrieving the next changed node at step 1408. At step 1410, the "bucket:" column in the node table (e.g., column 1222 of node table 1212 (both of FIG. 12)) corresponding to the changed node may be scanned. At step 1412, for each matching node found in the scan and

the changed node's paths, all paths in the scanned bucket may be joined with all paths of the changed bucket. At step 1414, each matching node may be mapped to each qualified joined

If there are no more changed nodes at step 1398, then, in reduce phase 1400, a determination may be made at step 1402 that there are more node and paths to process. If so, then the next node and its paths may be retrieved at step 1404. Each path may then be added to the appropriate node bucket at step 1406. If there are no more nodes and paths to process at step 1402, the process may stop at step 1416.

FIG. 14 shows a process 1420 for calculating a system trust score in accordance with certain embodiments of the present disclosure. Process 1420 includes verifying at least one entry in the entity's profile at step 1422, determining connectivity metrics for a social network at step 1424, performing a web search to determine publicly available information at step 1426, identifying past transactions at step 1428, receiving ratings information from a third-party source at step 1430, calculating component scores at step 1432, determining whether user weightings have been received at step 143, combining component scores using default weights at step 1436, and combining component scores using user weights at step 1438. It will be understood that process 1420 depicts illustrative steps for calculating a system trust score, and that one or more of steps 1422-1438 may be omitted and additional steps added to process 1420 as will be apparent to those of skill in the art without departing from the scope hereof.

At step 1422, processing circuitry, such as processing circuitry of access application 102 or application server 106, may verify at least one entry in an entity's profile. The entry may be one or more pieces of verification data, such as verification data described in connection with data verification component 404 depicted in FIG. 4. For example, the processing circuitry may verify one or more of a human user's email address, phone number, mailing address, education information, employment information. At step 1424, the processing circuitry may determine connectivity metrics for a social network. The connectivity metrics may comprise metrics as discussed in connection with network connectivity component 406 depicted in FIG. 4. The connectivity metrics may include, but are not limited to, number of friends, number of posts, or number of messages. At step 1426, the processing circuitry may perform a web search to determine publicly available information associated with the entity. For example, the processing circuitry may perform search engine mining as discussed above in relation to search engine mining component 416 depicted in FIG. 4. The processing circuitry may also determine information such as the entity's credit score or available court data, as discussed above in relation to credit score

component 408 and court data component 410 depicted in FIG. 4. At step 1428, the processing circuitry may identify past transactions associated with the entity. For example, the processing circuitry may identify past financial transactions that the entity has taken part in and whether the financial transactions were completed favorably (e.g., paid back a loan) or unfavorably (e.g., defaulted on a loan). At step 1430, the processing circuitry may receive ratings information from a third-party source, as discussed above in relation to ratings/feedback data component 412 depicted in FIG. 4. As an illustrative example, the processing circuitry may receive ratings from the Better Business Bureau or from an online ratings site such as Yelp about an entity. At 1432, the processing circuitry may calculate component scores based on the information received from steps 1424-1430. The processing circuitry may calculate the components scores in any suitable manner, such as the methods discussed above in FIGs. 8 and 9.

At step 1434, the processing circuitry may determine whether user-specified weightings have been received. For example, a user may have specified custom weightings through a user interface such as interface 700 depicted in FIG. 7. If user-specified weightings have been received, then the processing circuitry may combine the component scores using the user-specified weights at step 1438. If user-specified weights have not been received, then the processing circuitry may combine the component scores using default weights at step 1436, such as the default weights depicted in FIG. 5. In some embodiments, the processing circuitry may calculate the system trust score in response to a user request for the system trust score. For example, the user may press calculate button 618 depicted in FIG. 6, and in response, the processing circuitry may calculate the system trust score in substantially real-time. In other embodiments, the processing circuitry may calculate the system trust score in advance of a user request for the system trust score. In such embodiments, the processing circuitry may retrieve a pre-calculated system trust score, for example from data store 110 depicted in FIG. 1, in response to the user request for the system trust score.

FIG. 15 shows a process 1500 for calculating a peer trust score in accordance with certain embodiments of the present disclosure. Process 1500 includes receiving a system trust score at step 1502, identifying paths from a first entity to a second entity at step 1504, receiving data from a remote source associated with at least one of the first entity or the second entity at step 1506, updating component scores at step 1508, and calculating a peer trust score based on the updated component scores at step 1510. It will be understood that process 1500 depicts illustrative steps for calculating a peer trust score, and that one or more of steps 1502-1510 may be omitted and additional steps added to process 1500 as will be

apparent to those of skill in the art without departing from the scope hereof. For example, the process 1500 for calculating a peer trust score is depicted in FIG. 15 as an update to a system trust score. However, it will be understood that the peer trust score may be calculated from component scores independently from a system trust score, as discussed above.

At step 1502, processing circuitry, such as processing circuitry of access application 102 or application server 106, may receive a system trust score. The system trust score may have been calculated previously, such as by a method similar to process 1420 depicted in FIG. 14. At step 1504, the processing circuitry may identify paths from a first entity to a second entity. For example, the processing circuitry may utilize a path counting approach, as discussed above in relation to FIGs. 11-13. At step 1506, the processing circuitry my receive data from a remote source associated with at least one of the first entity or the second entity. For example, the processing circuitry may receive data regarding the second entity's social connections, credit score, court data, or previous transaction history with the first entity.

At step 1508, the processing circuitry may update component scores based on the information from steps 1502 – 1506. In some embodiments, updating component scores comprises updating less than all of the component scores that comprise the system trust score. For example, the processing circuitry may only update the network connectivity component to take into account the mutual contacts of the first entity and the second entity. Other component scores that were calculated with respect to the second entity's system trust score, such as credit score or court data, may not be affected by the additional social graph information. At step 1510, the processing circuitry may calculate the peer trust score based on the updated components by, for instance, combining the component scores using a weighted average. In some embodiments, the processing circuitry may calculate the peer trust score in response to a user request for the peer trust score. For example, the user may press calculate button 618 depicted in FIG. 6, and in response, the processing circuitry may calculate the peer trust score in substantially real-time. In other embodiments, the processing circuitry may calculate the peer trust score in advance of a user request for the peer trust score. In such embodiments, the processing circuitry may retrieve a pre-calculated peer trust score, for example from data store 110 depicted in FIG. 1, in response to the user request for the peer trust score.

FIG. 16 shows a process 1600 for calculating a contextual trust score in accordance with certain embodiments of the present disclosure. Process 1600 includes receiving a peer trust score at step 1602, receiving an indication of an activity to be performed by a first entity and a second entity at step 1604, updating component scores based on the activity at step

1606, updating weights based on the activity at step 1608, and calculating a contextual score based on the updated component scores and the updated weights at step 1610. It will be understood that process 1600 depicts illustrative steps for calculating a contextual trust score, and that one or more of steps 1602-1610 may be omitted and additional steps added to process 1600 as will be apparent to those of skill in the art without departing from the scope hereof. For example, the process 1600 for calculating a peer trust score is depicted in FIG. 16 as an update to a peer trust score. However, it will be understood that the contextual trust score may be calculated from component scores independently from a system trust score or a peer trust score, as discussed above.

At step 1602, processing circuitry, such as processing circuitry of access application 102 or application server 106, may receive a peer trust score. The system trust score may have been calculated previously, such as by a method similar to process 1500 depicted in FIG. 15. At step 1604, the processing circuitry may receive an indication of an activity to be performed by a first entity and a second entity. For example, the processing circuitry may receive the indication of the activity through transaction selector 606 depicted in FIG. 6. The processing circuitry may also receive details of the activity/transaction through transaction details field 608, as discussed above in relation to FIG. 6. At step 1606, the processing circuitry may update component scores based on the activity. For example, certain component scores may be affected by a type of transaction. As an illustrative example, the transaction history component, such as transaction history component 418 depicted in FIG. 4, may be updated to reflect only the transaction history of the particular type of transaction that is being performed by the first and second entity. At step 1608, the processing circuitry may update weights based on the activity. As discussed above in relation to FIG. 7, different transaction types may be associated with different weightings, and the components may be combined according to these different weightings. At step 1610, the processing circuitry may calculate the contextual trust score based on the updated component scores and the updated weights, for example, by taking a weighted average of the updated component scores according to the updated weights. In some embodiments, the processing circuitry may calculate the contextual trust score in response to a user request for the contextual trust score. For example, the user may press calculate button 618 depicted in FIG. 6, and in response, the processing circuitry may calculate the contextual trust score in substantially real-time. In other embodiments, the processing circuitry may calculate the contextual trust score in advance of a user request for the contextual trust score. In such embodiments, the processing

circuitry may retrieve a pre-calculated contextual trust score, for example from data store 110 depicted in FIG. 1, in response to the user request for the contextual trust score.

The foregoing is merely illustrative of the principles of the disclosure, and the systems, devices, and methods described herein are presented for purposes of illustration, and not of limitation. Variations and modifications will occur to those of skill in the art after reviewing this disclosure. The disclosed features may be implemented, in any combination and subcombination (including multiple dependent combinations and subcombinations), with one or more other features described herein. The various features described or illustrated above, including any components thereof, may be combined or integrated in other systems. Moreover, certain features may be omitted or not implemented. Examples, changes, substitutions, and alterations ascertainable by one skilled in the art can be made without departing from the scope of the information disclosed herein.

What is Claimed is:

1. A method for updating a trust score, the method comprising:

identifying paths from a first entity to a second entity;

calculating a network connectivity score based on the identified paths;

receiving data about the second entity from a remote source;

calculating a ratings score based on the received data from the remote source;

determining a trust score for the second entity by combining the network connectivity score and the ratings score;

receiving an indication of an activity to be performed together by the first entity and the second entity; and

updating the trust score based on the indication of the activity.

2. The method of claim 1, wherein the first entity and the second entity are connected through a social network.

3. The method of claim 2, wherein identifying paths from the first entity to the second entity comprises identifying one or more intermediate entities in the social network that connect the first entity to the second entity.

4. The method of claim 2, wherein calculating the network connectivity score comprises determining a number of mutual friends between the first entity and the second entity.

5. The method of claim 4, wherein calculating the network connectivity score comprises assigning the network connectivity score according to a graduated scale based on the number of mutual friends between the first entity and the second entity.

6. The method of claim 1, wherein calculating a network connectivity score based on the identified paths comprises determining whether the identified paths exceeds a threshold number of paths.

7. The method of claim 1, wherein the received data is one of: a credit score, criminal history data, financial transaction history data, and/or business reviews data.

8. The method of claim 1, wherein determining the trust score for the second entity by combining the network connectivity score and the ratings score comprises combining the network connectivity score and the ratings score according to a weighted sum.

9. The method of claim 8, wherein the weighted sum is based on user-assigned weights.

10. The method of claim 8, wherein the weighted sum is a first weighted sum, and wherein updating the trust score based on the indication of the activity comprises combining the network connectivity score and the ratings score according to a second weighted sum, wherein the second weighted sum is different than the first weighted sum.

11. The method of claim 1, wherein at least one of the first entity and the second entity is a human user.

12. The method of claim 1, wherein at least one of the first entity and the second entity is a business.

13. The method of claim 1, further comprising resolving a decision related to the activity based, at least in part, on the updated trust score.

14. The method of claim 1, wherein the trust score for the second entity comprises a confidence range determined based on the network connectivity score and the ratings score.

15. The method of claim 1, wherein determining the trust score for the second entity comprises:

retrieving an initial trust score associated with the second entity; and

updating the initial trust score associated with the second entity based on the network connectivity score and the ratings score.

16. A system for updating a trust score, the system comprising:

processing circuitry configured to:

identify paths from a first entity to a second entity;

calculate a network connectivity score based on the identified paths;

receive data about the second entity from a remote source;

calculate a ratings score based on the received data from the remote source;

determine a trust score for the second entity by combining the network connectivity score and the ratings score;

receive an indication of an activity to be performed together by the first entity and the second entity; and

update the trust score based on the indication of the activity.

17. The system of claim 16, wherein the first entity and the second entity are connected through a social network.

18. The system of claim 17, wherein the processing circuitry is configured to identify paths from the first entity to the second entity by identifying one or more intermediate entities in the social network that connect the first entity to the second entity.

19. The system of claim 16, wherein the processing circuitry is configured to calculate the network connectivity score by determining a number of mutual friends between the first entity and the second entity.

20. The system of claim 19, wherein the processing circuitry is configured to calculate the network connectivity score by assigning the network connectivity score according to a graduated scale based on the number of mutual friends between the first entity and the second entity.

21. The system of claim 16, wherein the processing circuitry is configured to calculate a network connectivity score based on the identified paths by determining whether the identified paths exceeds a threshold number of paths.

22. The system of claim 16, wherein the received data is one of: a credit score, criminal history data, financial transaction history data, and/or business reviews data.

23. The system of claim 16, wherein the processing circuitry is configured to determine the trust score for the second entity by combining the network connectivity score and the ratings score by combining the network connectivity score and the ratings score according to a weighted sum.

24. The system of claim 23, wherein the weighted sum is based on user-assigned weights.

25. The system of claim 23, wherein the weighted sum is a first weighted sum, and wherein the processing circuitry is configured to update the trust score based on the indication of the activity by combining the network connectivity score and the ratings score according to a second weighted sum, wherein the second weighted sum is different than the first weighted sum.

26. The system of claim 16, wherein at least one of the first entity and the second entity is a human user.

27. The system of claim 16, wherein the processing circuitry is further configured to resolve a decision related to the activity based, at least in part, on the updated trust score.

28. The system of claim 16, wherein the trust score for the second entity comprises a confidence range determined based on the network connectivity score and the ratings score.

29. The system of claim 16, wherein the processing circuitry is configured to determine the trust score for the second entity by:

retrieving an initial trust score associated with the second entity; and

updating the initial trust score associated with the second entity based on the network connectivity score and the ratings score.

30. A system for updating a trust score, the system comprising:

means for identifying paths from a first entity to a second entity;

means for calculating a network connectivity score based on the identified paths;

means for receiving data about the second entity from a remote source;

means for calculating a ratings score based on the received data from the remote source;

means for determining a trust score for the second entity by combining the network connectivity score and the ratings score;

means for receiving an indication of an activity to be performed together by the first entity and the second entity; and

means for updating the trust score based on the indication of the activity.

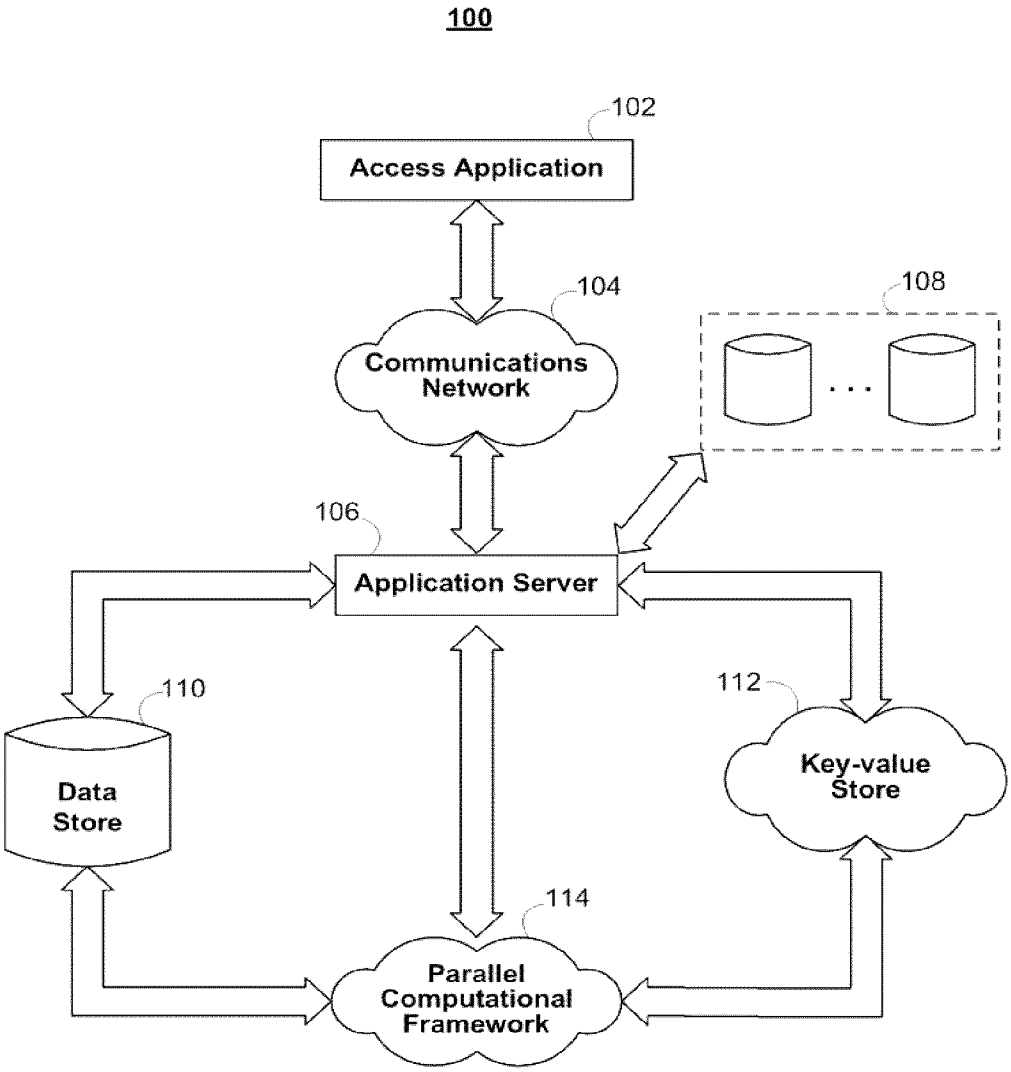

100
102
Access Application
104
Communications Network
108
106
Application Server
110
Data Store
112
Key-value Store
114
Parallel Computational Framework

FIG. 1

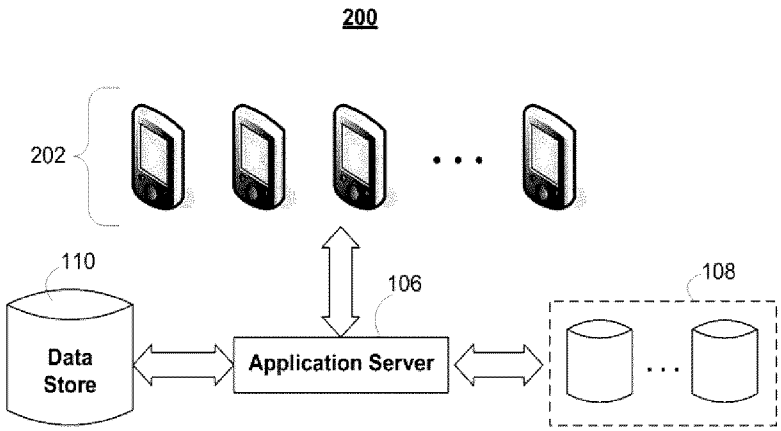



FIG. 2

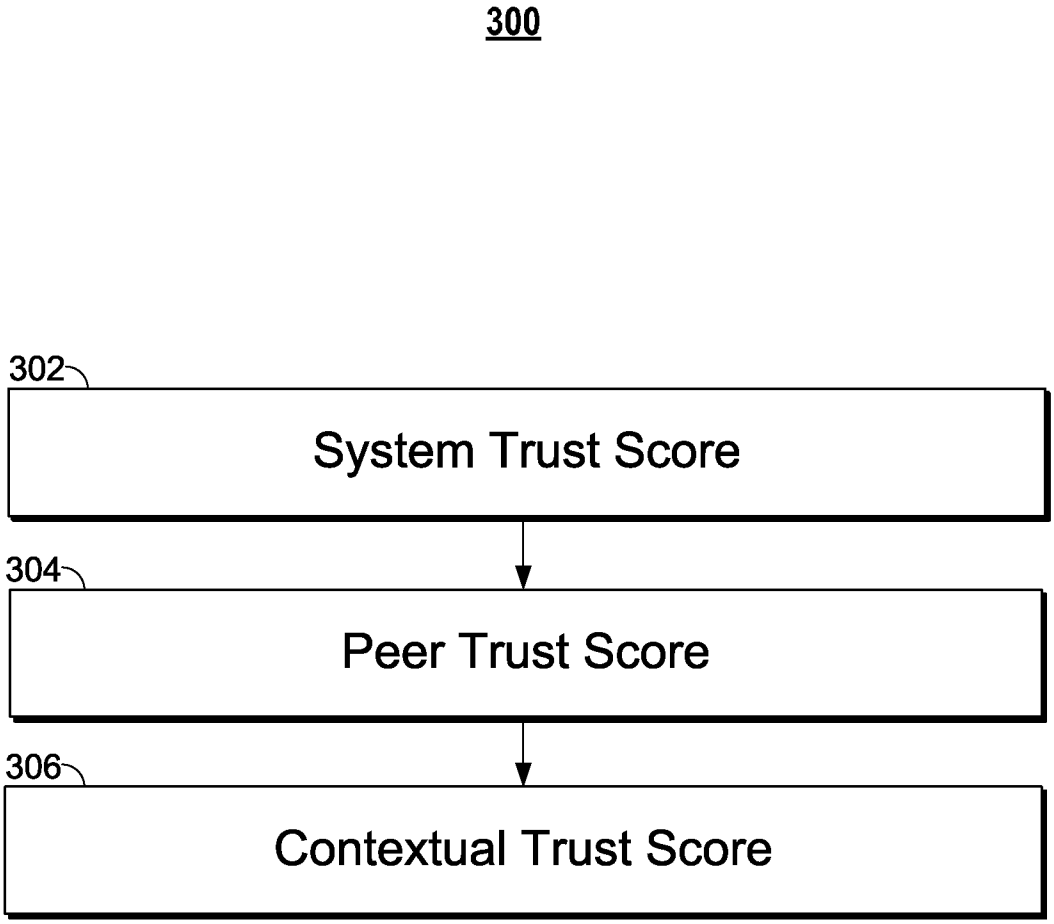

300
302
System Trust Score
304
Peer Trust Score
306
Contextual Trust Score

FIG. 3

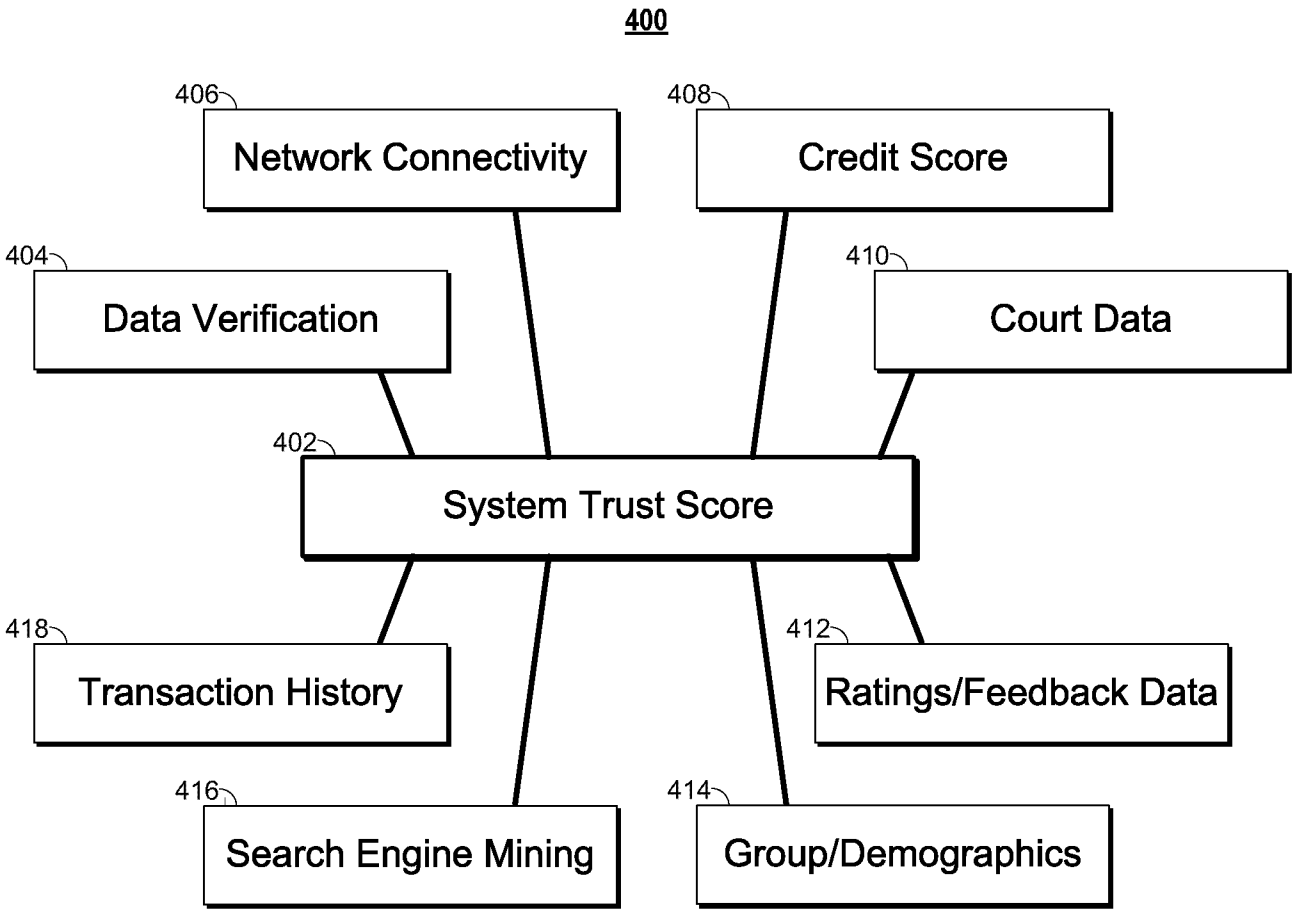

400
406
Network Connectivity
408
Credit Score
404
Data Verification
410
Court Data
402
System Trust Score
418
Transaction History
412
Ratings/Feedback Data
416
Search Engine Mining
414
Group/Demographics

FIG. 4

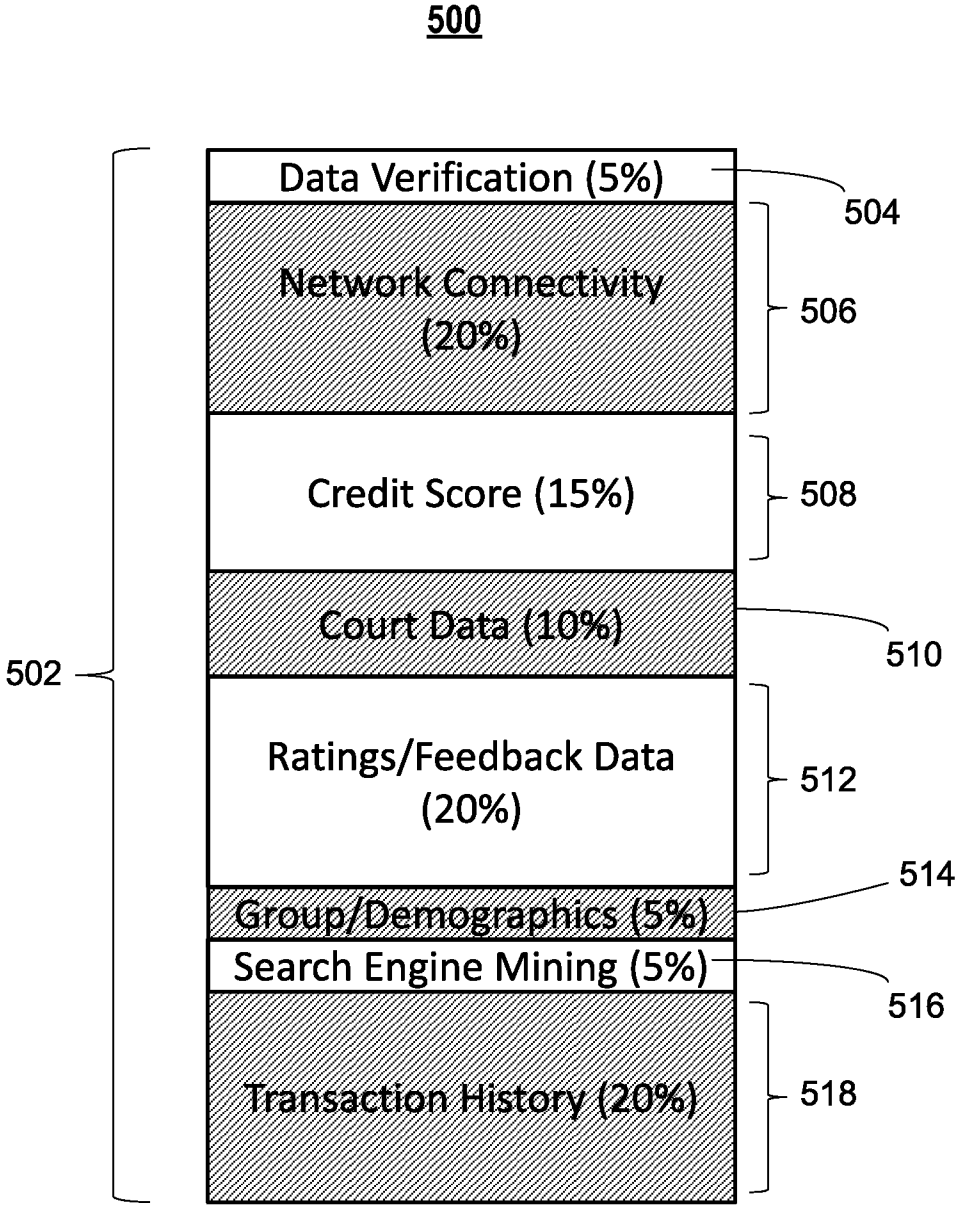

500
Data Verification (5%)
504
Network Connectivity (20%)
506
Credit Score (15%)
508
Court Data (10%)
510
502
Ratings/Feedback Data (20%)
512
514
Group/Demographics (5%)
Search Engine Mining (5%)
516
Transaction History (20%)
518

FIG. 5

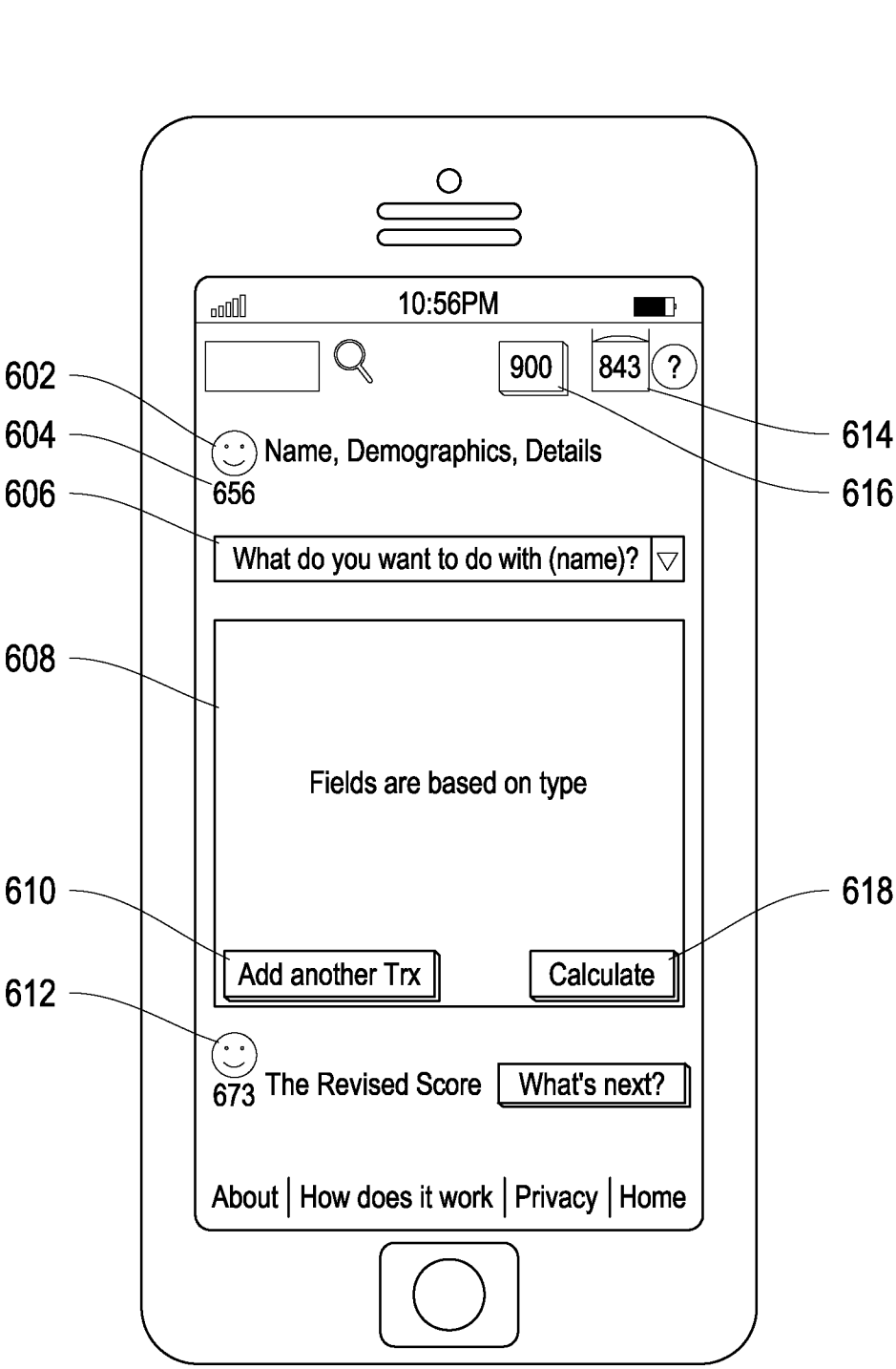

600
10:56PM
900
843
?
602
604
606
608
610
612
614
616
618
Name, Demographics, Details
656
What do you want to do with (name)?
Fields are based on type
Add another Trx
Calculate
673
The Revised Score
What's next?
About | How does it work | Privacy | Home

FIG. 6

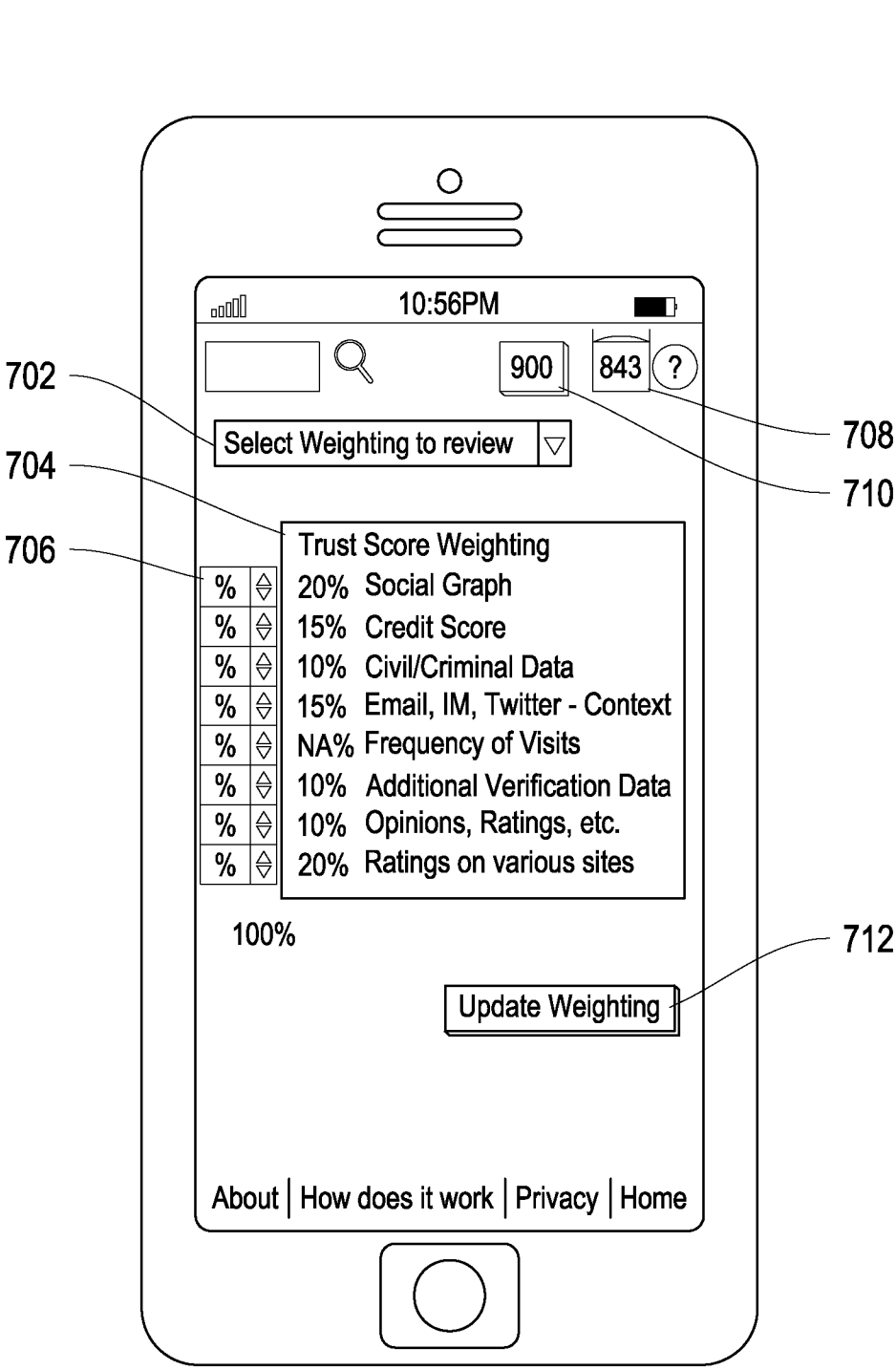

700
10:56PM
900
843
?
702
Select Weighting to review
708
704
710
706
Trust Score Weighting
% 20% Social Graph
% 15% Credit Score
% 10% Civil/Criminal Data
% 15% Email, IM, Twitter - Context
% NA% Frequency of Visits
% 10% Additional Verification Data
% 10% Opinions, Ratings, etc.
% 20% Ratings on various sites
100%
712
Update Weighting
About | How does it work | Privacy | Home

FIG. 7

800

802 804 806

| Metric | Component Score (out of 1000) | Component Score (%) |
|---|---|---|
| 100 or less friends | 200 | 20% |
| 100 – 250 friends | 400 | 40% |
| 250-400 friends | 600 | 60% |
| 400-500 friends | 800 | 80% |
| 500+ friends | 1000 | 100% |

FIG. 8

900

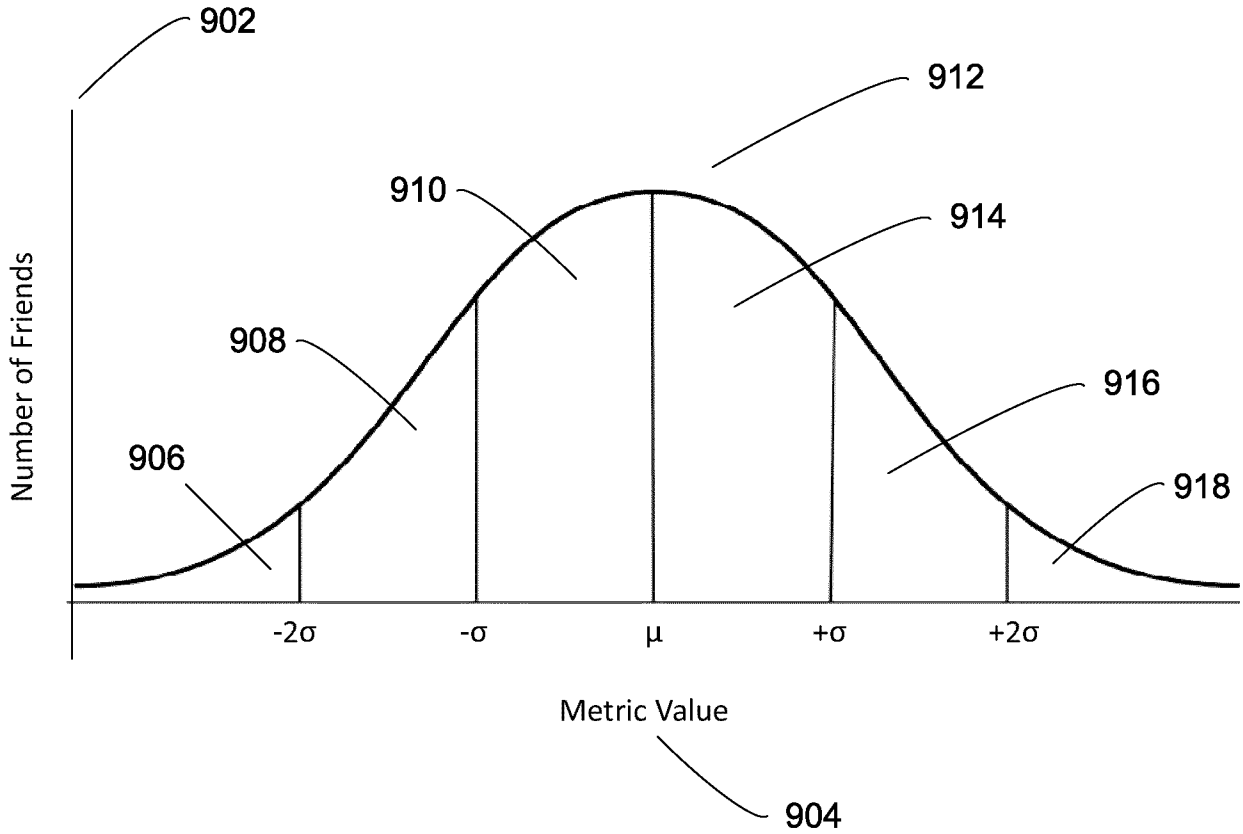

902
912
910
914
908
916
906
918
Number of Friends
-2σ
-σ
μ
+σ
+2σ
Metric Value
904

FIG. 9

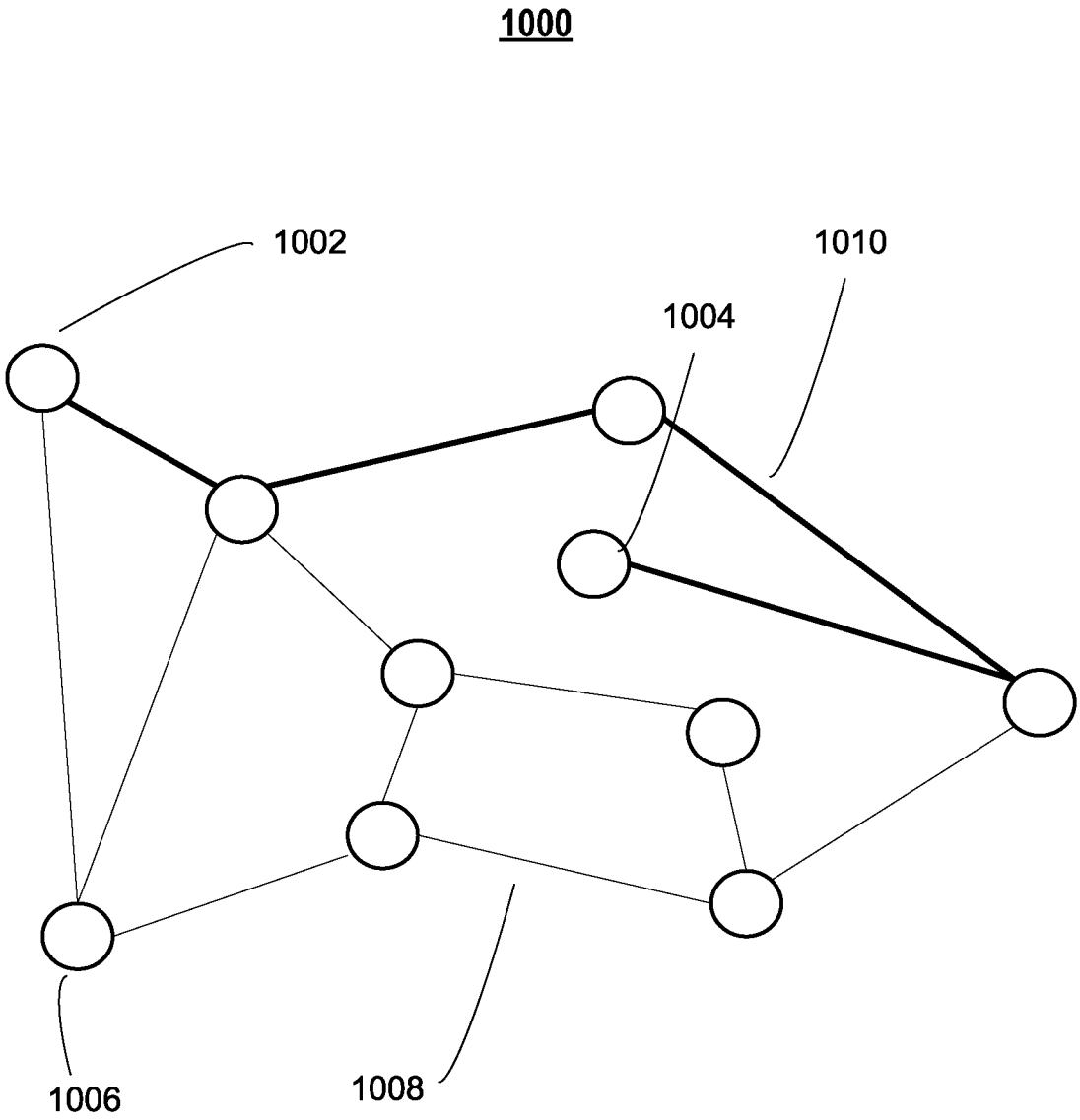



FIG. 10

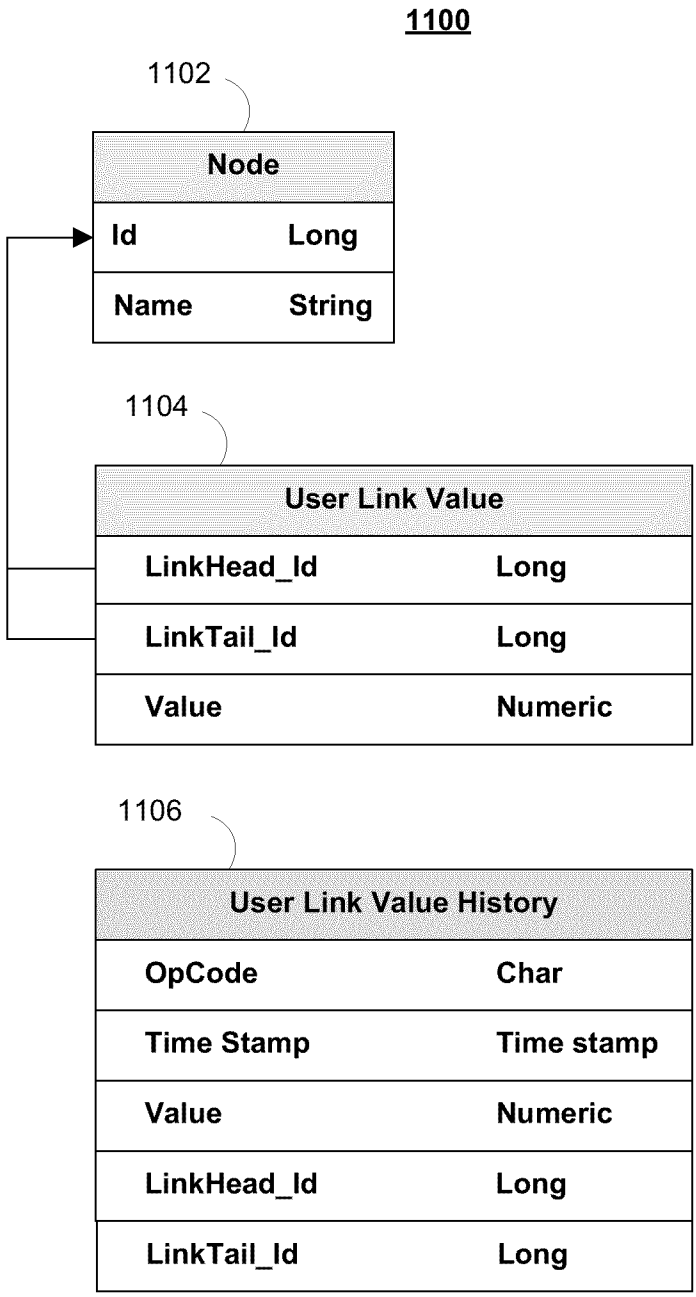

1100
1102
Node
Id Long
Name String
1104
User Link Value
LinkHead_Id Long
LinkTail_Id Long
Value Numeric
1106
User Link Value History
OpCode Char
Time Stamp Time stamp
Value Numeric
LinkHead_Id Long
LinkTail_Id Long

FIG. 11

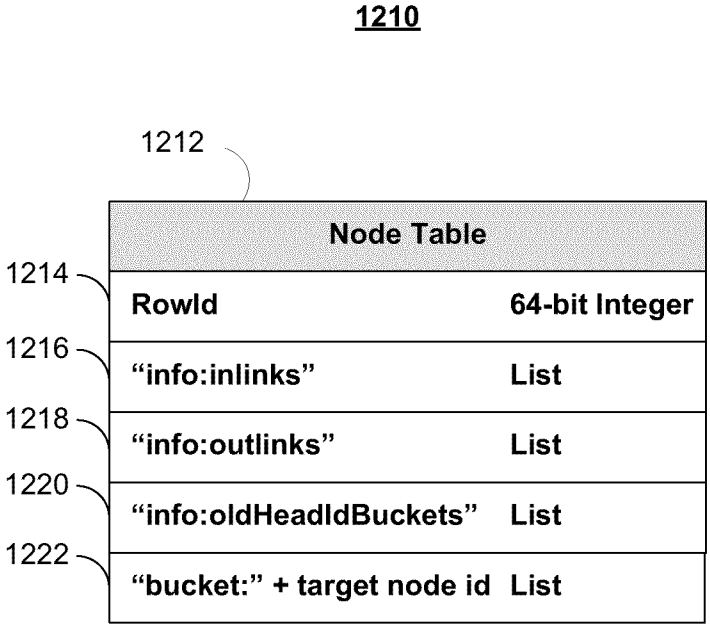

1210
1212
Node Table
1214
RowId
64-bit Integer
1216
“info:inlinks”
List
1218
“info:outlinks”
List
1220
“info:oldHeadIdBuckets”
List
1222
“bucket:” + target node id
List

FIG. 12

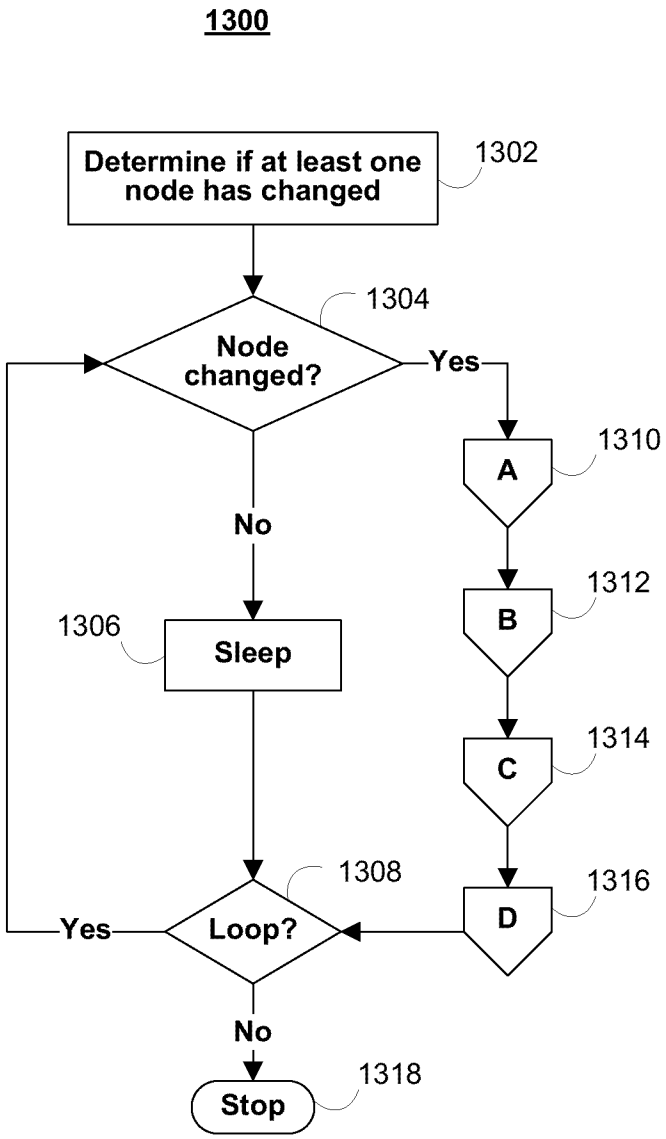

1300
Determine if at least one node has changed
1302
1304
Node changed?
Yes
No
1306
Sleep
1310
A
1312
B
1314
C
1316
D
1308
Loop?
Yes
No
1318
Stop

FIG. 13A

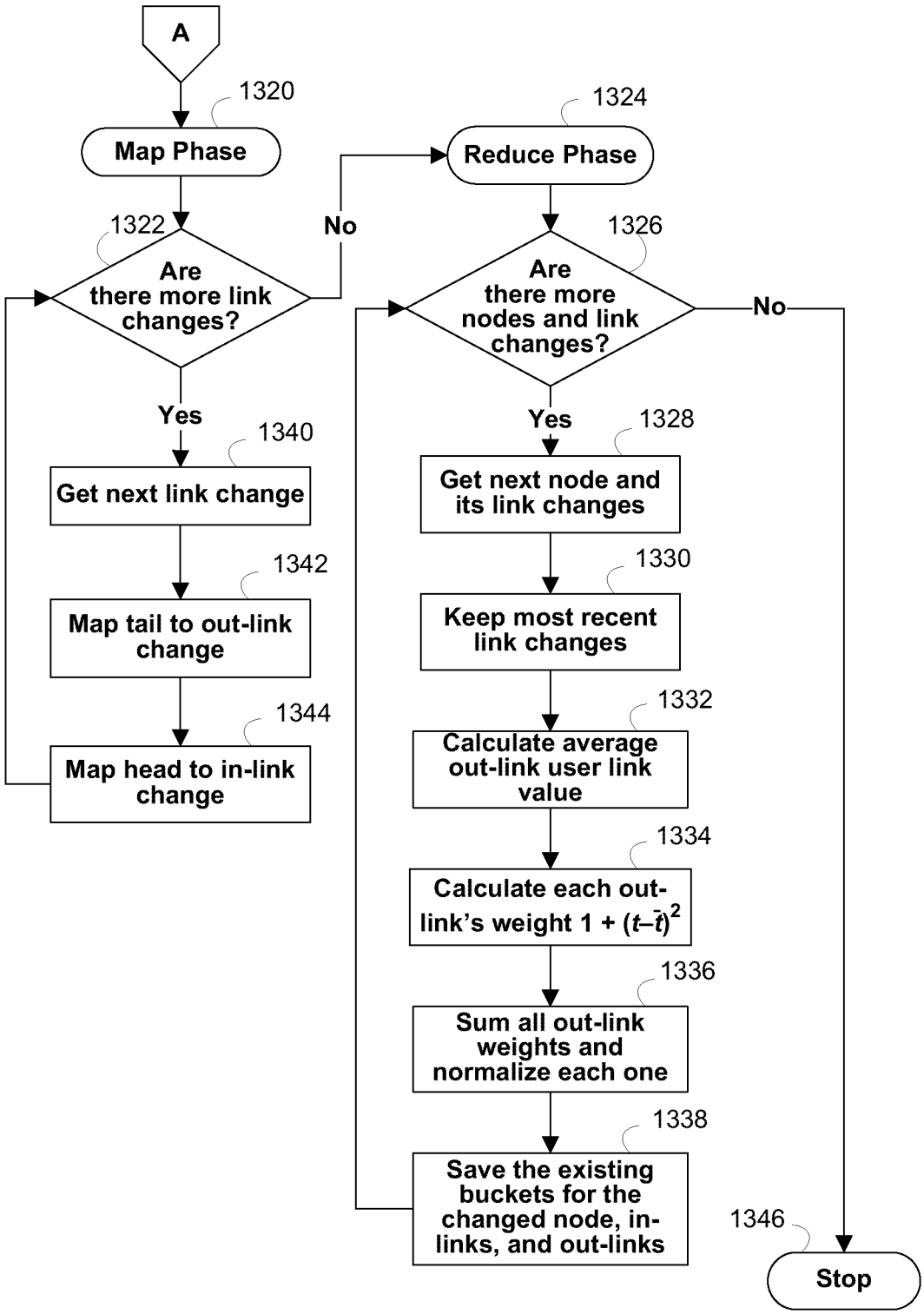

A
1320
Map Phase
1322
Are there more link changes?
No
Yes
1340
Get next link change
1342
Map tail to out-link change
1344
Map head to in-link change
1324
Reduce Phase
1326
Are there more nodes and link changes?
No
Yes
1328
Get next node and its link changes
1330
Keep most recent link changes
1332
Calculate average out-link user link value
1334
Calculate each out-link's weight 1 + $(t-\bar{t})^2$
1336
Sum all out-link weights and normalize each one
1338
Save the existing buckets for the changed node, in-links, and out-links
1346
Stop

FIG. 13B

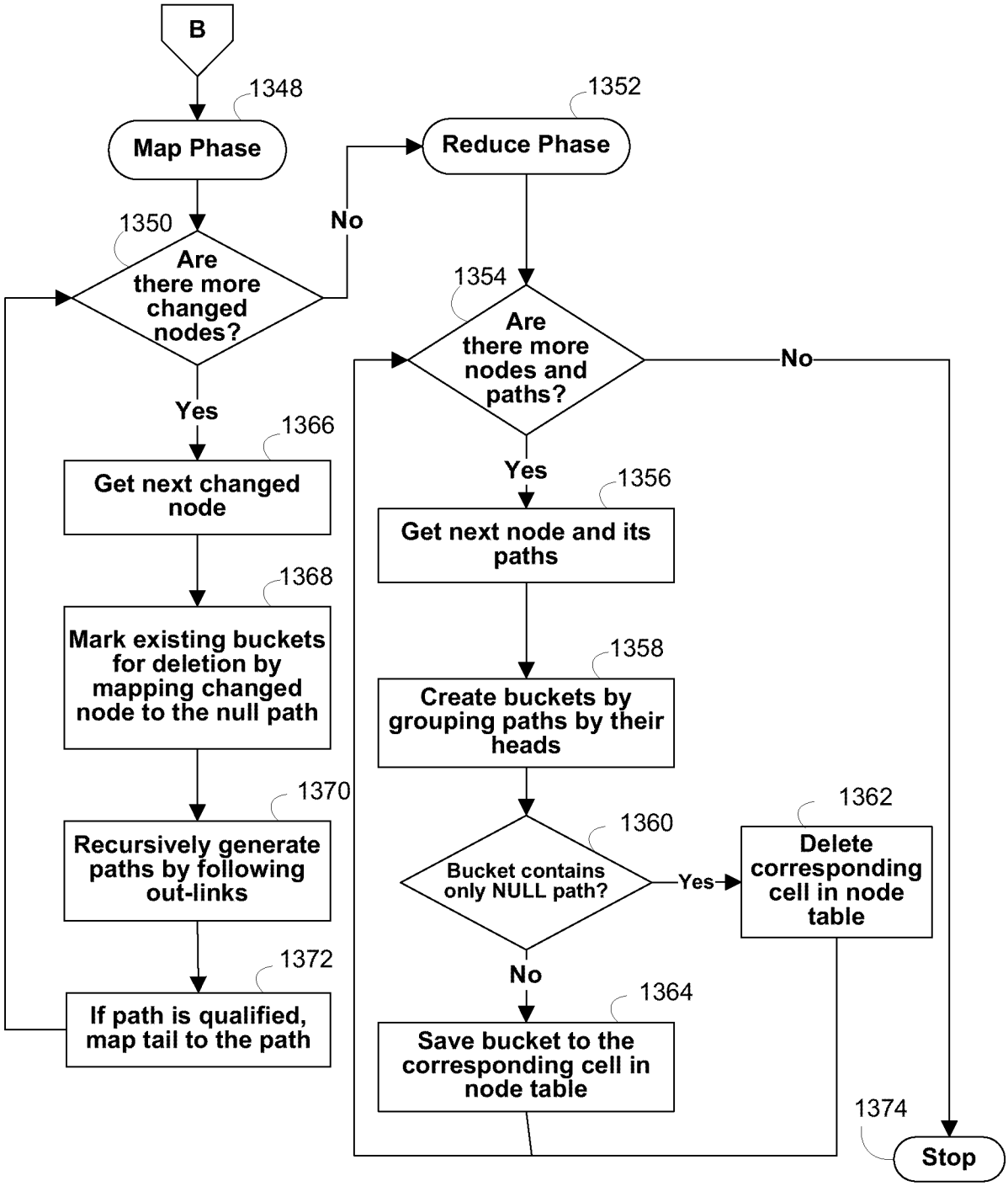

B
1348
Map Phase
1350
Are there more changed nodes?
No
Yes
1366
Get next changed node
1368
Mark existing buckets for deletion by mapping changed node to the null path
1370
Recursively generate paths by following out-links
1372
If path is qualified, map tail to the path
1352
Reduce Phase
1354
Are there more nodes and paths?
No
Yes
1356
Get next node and its paths
1358
Create buckets by grouping paths by their heads
1360
Bucket contains only NULL path?
Yes
1362
Delete corresponding cell in node table
No
1364
Save bucket to the corresponding cell in node table
1374
Stop

FIG. 13C

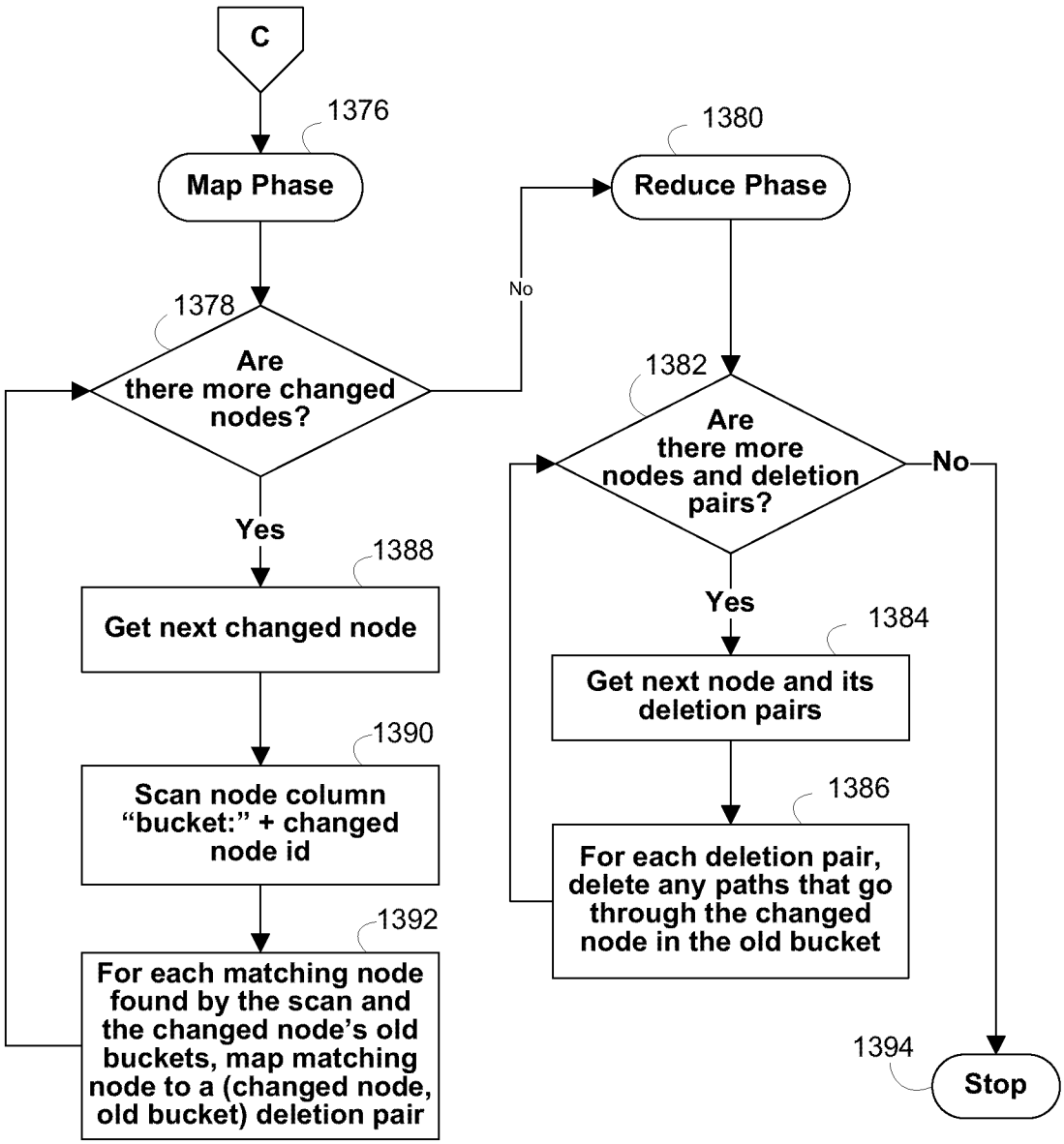

C
1376
Map Phase
1378
Are there more changed nodes?
No
Yes
1388
Get next changed node
1390
Scan node column "bucket:" + changed node id
1392
For each matching node found by the scan and the changed node's old buckets, map matching node to a (changed node, old bucket) deletion pair
1380
Reduce Phase
1382
Are there more nodes and deletion pairs?
No
Yes
1384
Get next node and its deletion pairs
1386
For each deletion pair, delete any paths that go through the changed node in the old bucket
1394
Stop

FIG. 13D

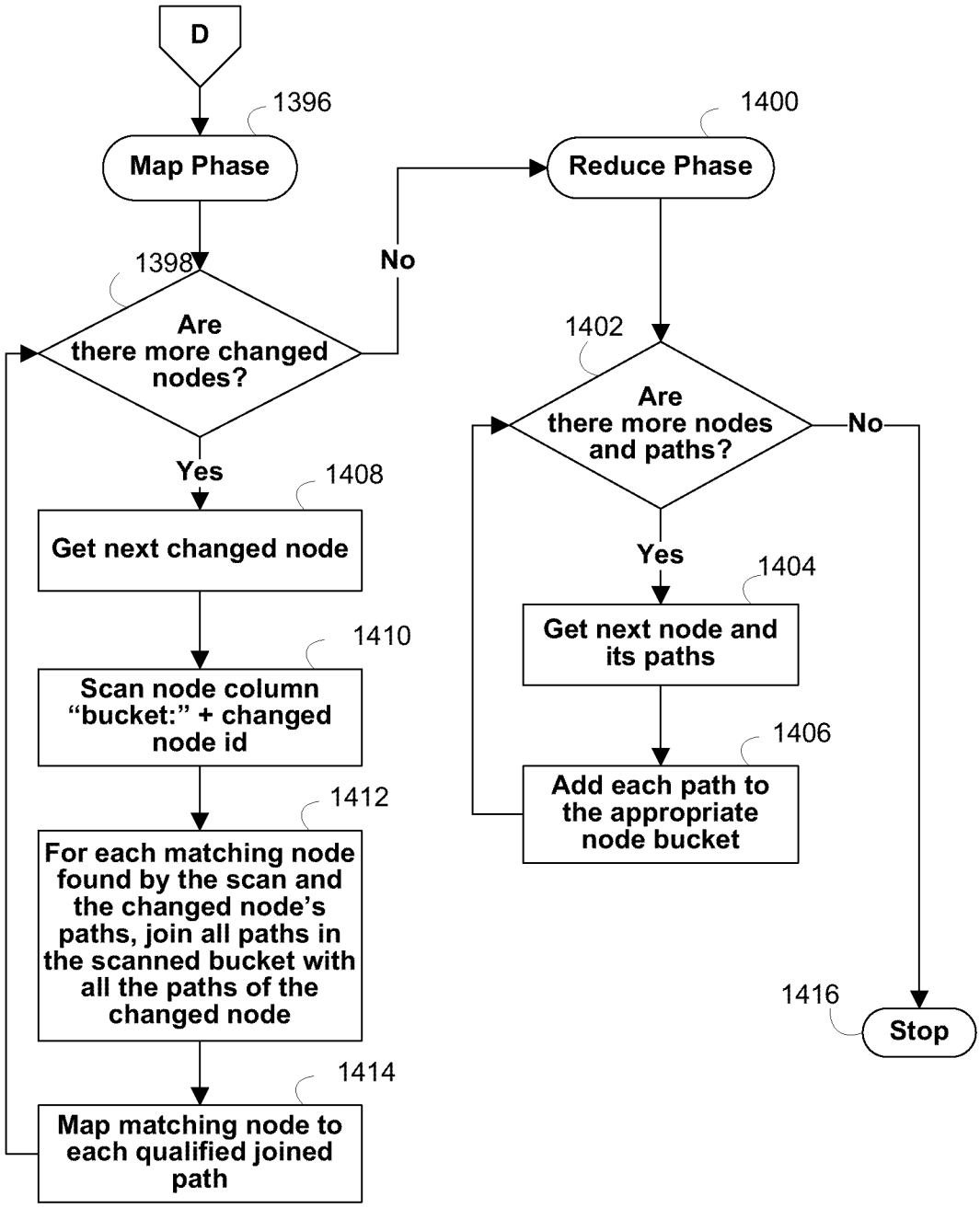

D
1396
Map Phase
1398
Are there more changed nodes?
No
Yes
1408
Get next changed node
1410
Scan node column "bucket:" + changed node id
1412
For each matching node found by the scan and the changed node's paths, join all paths in the scanned bucket with all the paths of the changed node
1414
Map matching node to each qualified joined path
1400
Reduce Phase
1402
Are there more nodes and paths?
No
Yes
1404
Get next node and its paths
1406
Add each path to the appropriate node bucket
1416
Stop

FIG. 13E

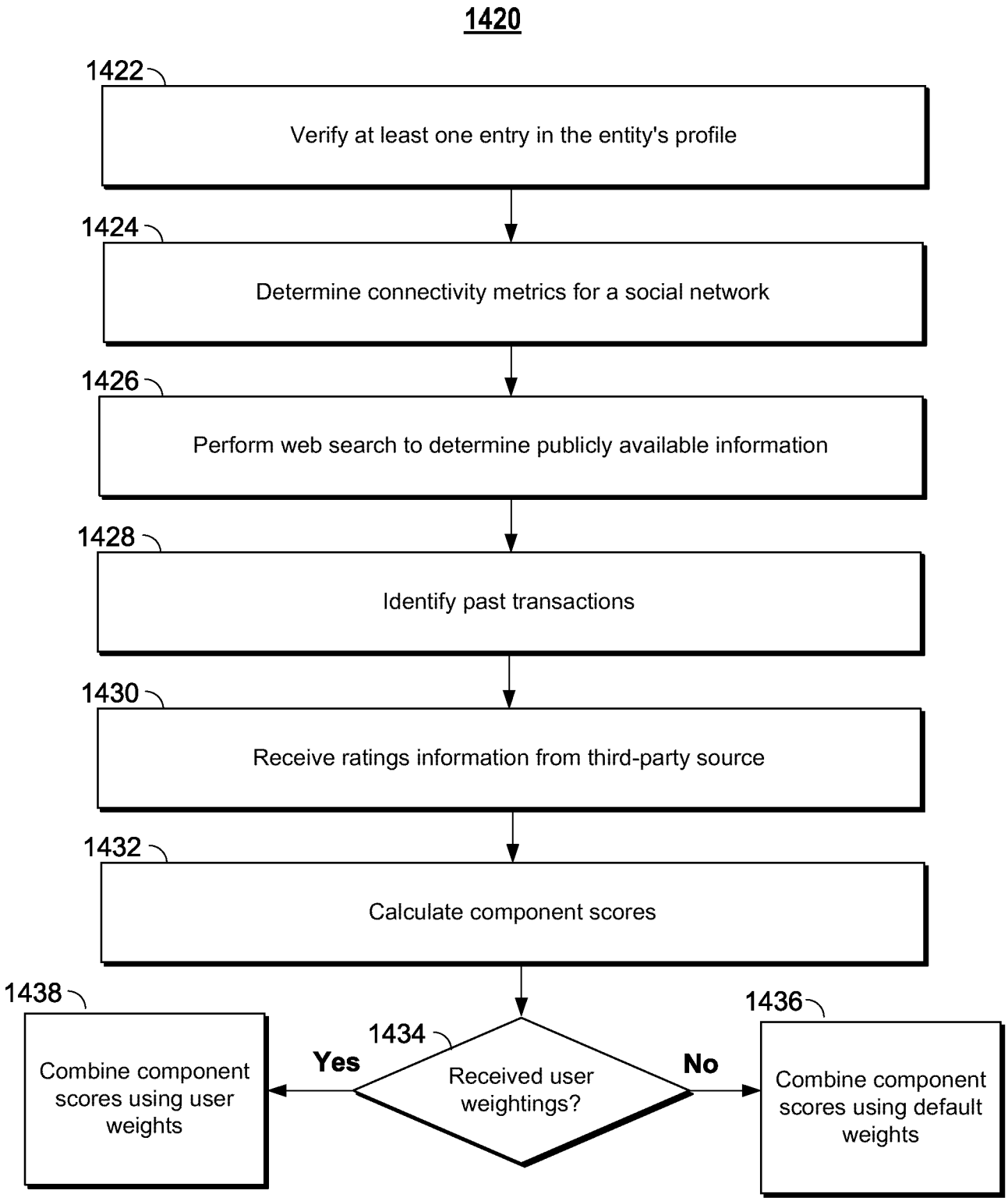

1420
1422
Verify at least one entry in the entity's profile
1424
Determine connectivity metrics for a social network
1426
Perform web search to determine publicly available information
1428
Identify past transactions
1430
Receive ratings information from third-party source
1432
Calculate component scores
1434
Received user weightings?
Yes
No
1438
Combine component scores using user weights
1436
Combine component scores using default weights

FIG. 14

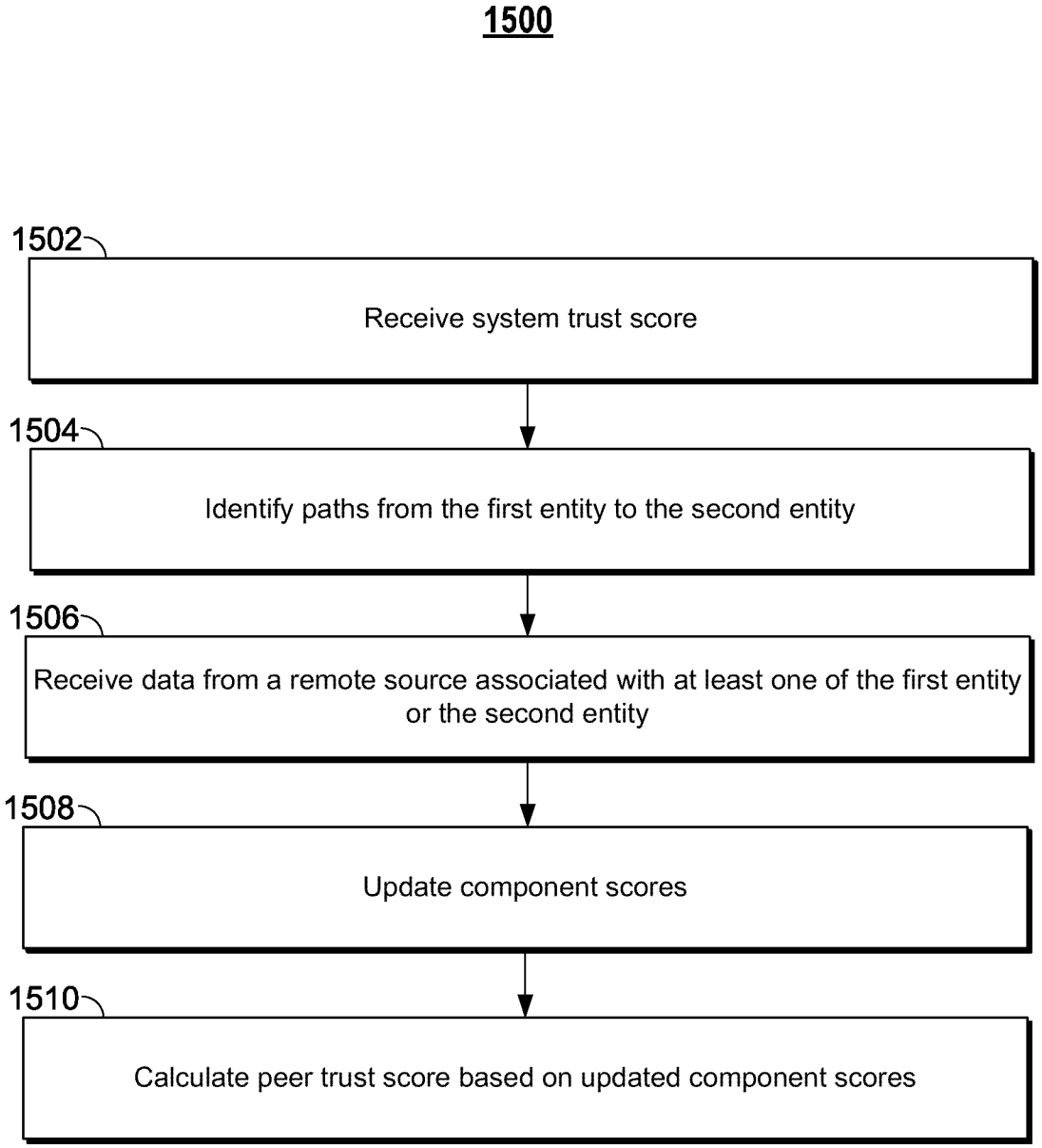

1500
1502
Receive system trust score
1504
Identify paths from the first entity to the second entity
1506
Receive data from a remote source associated with at least one of the first entity or the second entity
1508
Update component scores
1510
Calculate peer trust score based on updated component scores

FIG. 15

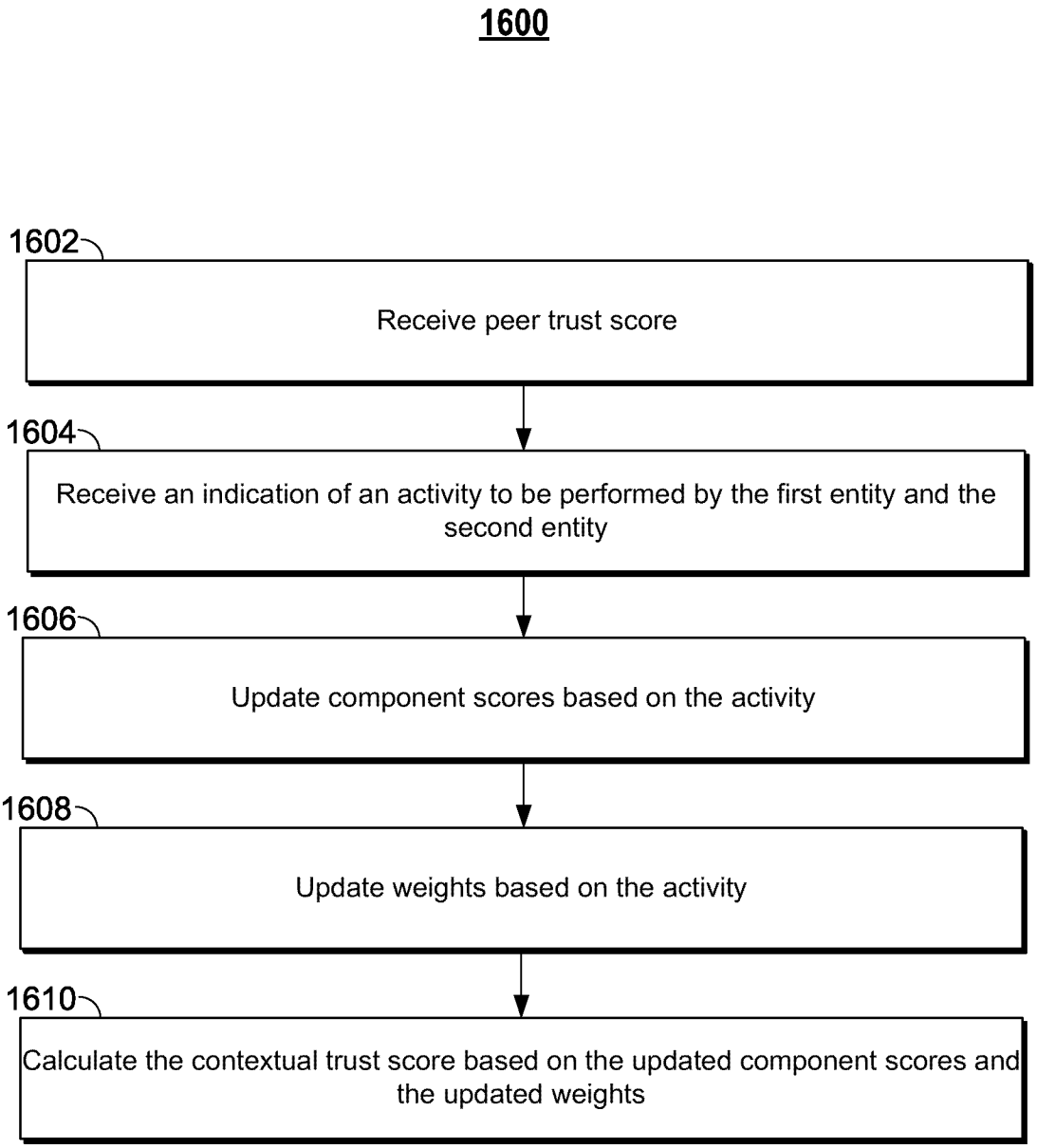

1600
1602
Receive peer trust score
1604
Receive an indication of an activity to be performed by the first entity and the second entity
1606
Update component scores based on the activity
1608
Update weights based on the activity
1610
Calculate the contextual trust score based on the updated component scores and the updated weights

FIG. 16

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/CA2016/050305**

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G06Q 30/00*** (2012.01), ***G06Q 40/02*** (2012.01), ***H04L 12/16*** (2006.01)

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
***G06Q 30/00*** (2012.01), ***G06Q 40/02*** (2012.01), ***H04L 12/16*** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
None.

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Canadian Patent Database (Intellect): trust, score, rating, mutual, intermedi*, contact, colleague
Questel Orbit: trust, score, value, metric, path, degree, separate*

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | WO2013026095A1, "(Matsumoto, Y. et al.)", 28 February 2013 (28-02-2013)<br>* entire document * | 1-30 |
| A | CA2600344A1, "(Shull, M. et al.)", 08 September 2006 (08-09-2006)<br>* para 0070-0073 * | 1-30 |
| A | CA2775899A1, "(Chrapko E. et al.)", 07 April 2011 (07-04-2011)<br>* p. 12 L21-24 * | 1-30 |

☐ Further documents are listed in the continuation of Box C.

☑ See patent family annex.

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

| | |
|---|---|
| Date of the actual completion of the international search<br>24 May 2016 (24.05.2016) | Date of mailing of the international search report<br>03 June 2016 (03-06-2016) |
| Name and mailing address of the ISA/CA<br>Canadian Intellectual Property Office<br>Place du Portage I, C114 - 1st Floor, Box PCT<br>50 Victoria Street<br>Gatineau, Quebec K1A 0C9<br>Facsimile No.: 819-953-2476 | Authorized officer<br>Zarrar Riaz (819) 635-3855 |

**INTERNATIONAL SEARCH REPORT**
Information on patent family members

International application No.
**PCT/CA2016/050305**

| Patent Document Cited in Search Report | Publication Date | Patent Family Member(s) | Publication Date |
|---|---|---|---|
| WO2013026095A1 | 28 February 2013 (28-02-2013) | None | |
| CA2600344A1 | 08 September 2006 (08-09-2006) | | |
| | | CA2600344A1 | 08 September 2006 (08-09-2006) |
| | | CA2600373A1 | 08 September 2006 (08-09-2006) |
| | | EP1849058A2 | 31 October 2007 (31-10-2007) |
| | | EP1856639A2 | 21 November 2007 (21-11-2007) |
| | | EP1856640A2 | 21 November 2007 (21-11-2007) |
| | | US2006212925A1 | 21 September 2006 (21-09-2006) |
| | | US2006212930A1 | 21 September 2006 (21-09-2006) |
| | | US2006212931A1 | 21 September 2006 (21-09-2006) |
| | | US2006230039A1 | 12 October 2006 (12-10-2006) |
| | | WO2006081328A2 | 03 August 2006 (03-08-2006) |
| | | WO2006081328A3 | 25 January 2007 (25-01-2007) |
| | | WO2006094228A2 | 08 September 2006 (08-09-2006) |
| | | WO2006094228A3 | 02 April 2009 (02-04-2009) |
| | | WO2006094271A2 | 08 September 2006 (08-09-2006) |
| | | WO2006094271A3 | 19 April 2007 (19-04-2007) |
| | | WO2006094275A2 | 08 September 2006 (08-09-2006) |
| | | WO2006094275A3 | 16 April 2009 (16-04-2009) |
| CA2775899A1 | 07 April 2011 (07-04-2011) | | |
| | | CA2775899A1 | 07 April 2011 (07-04-2011) |
| | | BR112012007316A2 | 19 April 2016 (19-04-2016) |
| | | CN102668457A | 12 September 2012 (12-09-2012) |
| | | EP2484054A1 | 08 August 2012 (08-08-2012) |
| | | EP2484054A4 | 05 November 2014 (05-11-2014) |
| | | IL218813D0 | 28 June 2012 (28-06-2012) |
| | | JP2013506204A | 21 February 2013 (21-02-2013) |
| | | JP5735969B2 | 17 June 2015 (17-06-2015) |
| | | JP2015164055A | 10 September 2015 (10-09-2015) |
| | | MX2012003721A | 28 June 2012 (28-06-2012) |
| | | US2012182882A1 | 19 July 2012 (19-07-2012) |
| | | US9171338B2 | 27 October 2015 (27-10-2015) |
| | | US2014258160A1 | 11 September 2014 (11-09-2014) |
| | | WO2011038491A1 | 07 April 2011 (07-04-2011) |