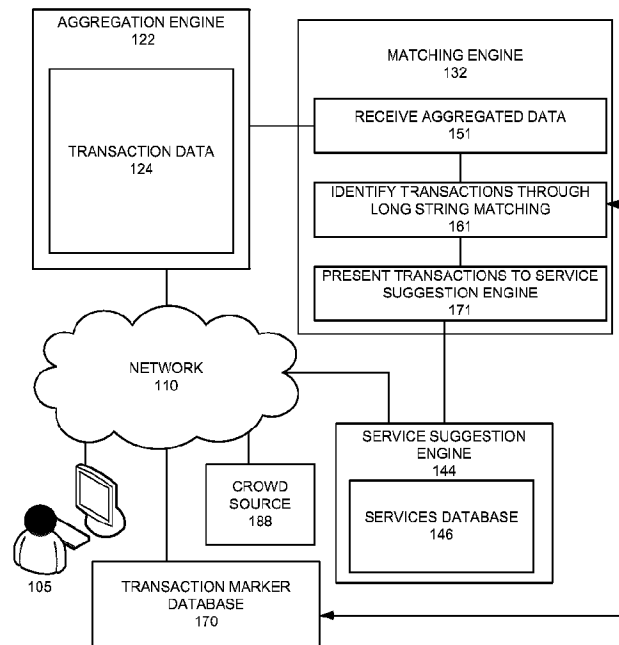




(86) **Date de dépôt PCT/PCT Filing Date:** 2014/12/16
 (87) **Date publication PCT/PCT Publication Date:** 2015/06/25
 (45) **Date de délivrance/Issue Date:** 2020/08/25
 (85) **Entrée phase nationale/National Entry:** 2016/06/16
 (86) **N° demande PCT/PCT Application No.:** US 2014/070685
 (87) **N° publication PCT/PCT Publication No.:** 2015/095242
 (30) **Priorité/Priority:** 2013/12/16 (US61/916,703)

(51) **Cl.Int./Int.Cl. G06F 16/901** (2019.01),
G06F 16/907 (2019.01), **G06F 7/00** (2006.01),
G06Q 30/00 (2012.01)
 (72) **Inventeurs/Inventors:**
 CALDWELL, JOHN RYAN, US;
 DEWITT, BRANDON, US;
 STIEN, BRIAN, US
 (73) **Propriétaire/Owner:**
 MX TECHNOLOGIES, INC., US
 (74) **Agent:** BENNETT JONES LLP

(54) **Titre : FILTRAGE DE LONGUES CHAINES DE CARACTERES SUR DES DONNEES DE COMPTES AGREGUEES**
 (54) **Title: LONG STRING PATTERN MATCHING OF AGGREGATED ACCOUNT DATA**



(57) **Abrégé/Abstract:**

The disclosure extends to methods, systems, and computer program products for identifying transaction data from within aggregated data and suggesting additional services based on the transaction data. The disclosure also extends to methods, systems, and computer based products for discovering transaction attributes within a database of aggregated account data including, aggregating and matching financial account attributes between a plurality of accounts in order to provide additional services to account holders.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 June 2015 (25.06.2015)



(10) International Publication Number
WO 2015/095242 A1

- (51) International Patent Classification:
G06Q 30/00 (2012.01)
- (21) International Application Number:
PCT/US2014/070685
- (22) International Filing Date:
16 December 2014 (16.12.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/916,703 16 December 2013 (16.12.2013) US
- (71) Applicant: MONEYDESKTOP, INC. [US/US]; 251 West Riverpark Drive, Suite 200, Provo, UT 84604 (US).
- (72) Inventors: CALDWELL, John, Ryan; 3214 North University Avenue, #405, Provo, UT 84604 (US). DEWITT, Brandon; 120 Village Lane, Provo, UT 84604 (US). STIEN, Brian; 120 Village Lane, Provo, UT 84604 (US).
- (74) Agent: EDWARDS, Terrence, J.; TechLaw Ventures, PLLC, 3290 West Mayflower Way, Lehi, UT 84043 (US).

- (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))

[Continued on next page]

(54) Title: LONG STRING PATTERN MATCHING OF AGGREGATED ACCOUNT DATA

(57) Abstract: The disclosure extends to methods, systems, and computer program products for identifying transaction data from within aggregated data and suggesting additional services based on the transaction data. The disclosure also extends to methods, systems, and computer based products for discovering transaction attributes within a database of aggregated account data including, aggregating and matching financial account attributes between a plurality of accounts in order to provide additional services to account holders.

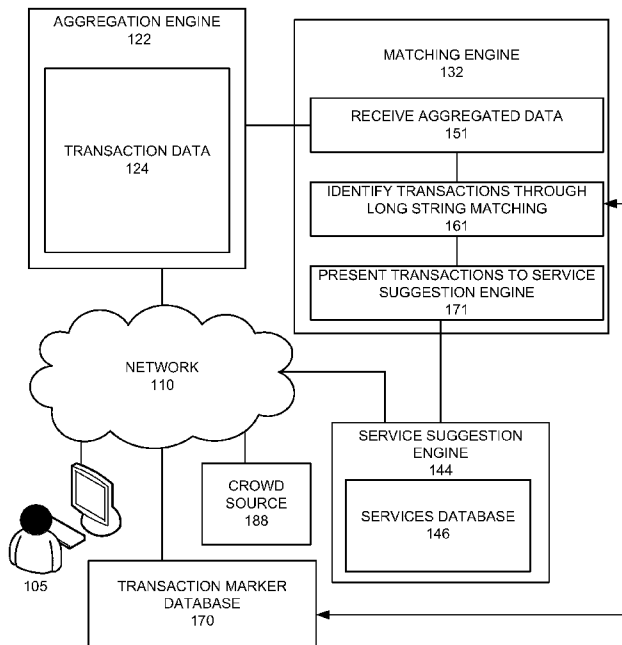


FIG. 1

WO 2015/095242 A1

WO 2015/095242 A1 

- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

LONG STRING PATTERN MATCHING OF AGGREGATED ACCOUNT DATA

BACKGROUND

[0001] In modern financial institutions, data management and utilization is paramount for providing the best possible service to clients or customers while increasing the profitability of the financial institution. Such things as purchases, automatic payments and deposits, transfers and other repeated transactions have been facilitated by financial entities, payees, and employers for the convenience of all involved.

[0002] A benefit for financial service providers is that by facilitating such transactions they increase the number of services that they can provide to their customers. The increased interaction is created in part because setting up automatic transactions and discovering new transactions that can be done automatically is time consuming and difficult. Accordingly, many customers and patrons are deterred from change because of the difficulty in changing automatic services. Additionally, better financial institutions are deterred from attracting new customers, or providing new services, because many of the customers have existing automatic services with lesser service providers.

SUMMARY OF THE DISCLOSURE

[0003] The disclosure relates to a method and system to use indexable data strings that match on patterns and allow lookup of matches to take significantly less time and even avoid disk access with the addition of presence (bloom) filters to determine presence of a match in the dataset before referencing the index at all. This reduces the seek time for a unique pattern in a large dataset to at best $O(1)$ and worst $O(\log(n))$. This allows the matching of large datasets against other large datasets based on patterns and this can save a significant amount of processing time and needed hardware/software for discovering transaction data.

[0004] The disclosure also relates generally to automatic financial services and relates more particularly, but not necessarily entirely, to a computer system for gleaning transaction attributes from data for suggesting additional services with minimal user input.

[0005] The features and advantages of the disclosure will be set forth in the description which follows, and in part will be apparent from the description, or may be learned by the practice of the disclosure without undue experimentation. The features and advantages of the disclosure may be realized and obtained by means of computing systems, networks, and combinations of firmware, software and hardware, particularly disclosed and enumerated in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Non-limiting and non-exhaustive implementations of the disclosure are described with reference to the following figures, wherein like reference numerals refer to like parts throughout the various views unless otherwise specified. Advantages of the disclosure will become better understood with regard to the following description and accompanying drawings where:

[0007] FIG. 1 illustrates a computer environment and network for identifying transaction within aggregated data in accordance with the principles and teachings of the disclosure;

[0008] FIG.2 illustrates a schematic representation of computer hardware and protocols that enable the various embodiments discussed herein in accordance with the principles and teachings of the disclosure;

[0009] FIG. 3 illustrates a method for identifying transaction within aggregated data in accordance with the principles and teachings of the disclosure;

[0010] FIG. 4 illustrates a method for identifying transaction within aggregated data in accordance with the principles and teachings of the disclosure; and

[0011] FIG. 5 illustrates a method for identifying transaction within aggregated data in accordance with the principles and teachings of the disclosure.

DETAILED DESCRIPTION

[0012] The disclosure extends to methods, systems, and computer based products for discovering transaction attributes within a database of aggregated account data including, aggregating and matching financial account attributes between a plurality of accounts in order to provide additional services to account holders. In the following description of the disclosure, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific implementations in which the disclosure may be practiced. It is understood that other implementations may be utilized and structural changes may be made without departing from the scope of the disclosure.

[0013] Aggregated account data generally can provide historical transaction information once it is parsed through by a knowledgeable entity. Everyday transactions that are repeated provide opportunity and information for financial service providers, retailers and customers alike. Such transactions as regular payments (bill-pay) and deposits (direct deposit) are important services that financial institutions provide to their customers and may be automated, or suggested for automation, for the convenience of customers once discovered in the account data. For simplicity and clarity, the service of making bill payments will be primarily used in the embodiments of the disclosure, however, the advantages and features disclosed herein also apply equally to any repeatable transaction or identifiable transaction from within aggregated data. Such services are so important in fact that customers will often select a financial institution and stay with a financial institution due to the automated services like bill-pay over many other factors. A common problem related to suggesting and providing additional services, such as to customers, is that the current transactions of interest may be with data that is stored by third parties and must be aggregated before it can be used. Additionally once the data has been aggregated, this large amount of data must be made usable. Accordingly, systems and methods will be disclosed herein for automating the process of identifying transaction from within large aggregation data stores.

[0014] In describing and claiming the subject matter of the disclosure, the following terminology will be used in accordance with the definitions set out below.

[0015] It must be noted that, as used in this specification and the appended claims, the singular forms “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise.

[0016] As used herein, “user” is intended to denote a person or entity that is utilizing the methods and the systems disclosed herein.

[0017] As used herein, the terms “comprising,” “including,” “containing,” “characterized by,” and grammatical equivalents thereof are inclusive or open-ended terms that do not exclude additional, unrecited elements or method steps.

[0018] As used herein, the phrase “consisting of” and grammatical equivalents thereof exclude any element or step not specified in the claim.

[0019] As used herein, the phrase “consisting essentially of” and grammatical equivalents thereof limit the scope of a claim to the specified materials or steps and those that do not materially affect the basic and novel characteristic or characteristics of the claimed disclosure.

[0020] An implementation of methods for providing additional services may comprise processes that: determine signature for transactions types and generate unique identifier for signatures; create index of unique identifiers; apply bloom filter; substitute signatures for transaction data; compare to index of unique identifiers; generate recognized patterns for transactions; present patterns and transactions to crowd sourcing; receive corrections from crowd sourcing; implement corrections into index of unique identifiers; suggest services based on patterns of transactions; and recording the above in computer memory used in the above processes.

[0021] An embodiment of a system for suggesting additional financial services to an account holder based on the account holder’s previous transactions may comprise:

[0022] an aggregation database within computer storage wherein the aggregation database is accessible over a computer network and wherein the aggregation database comprises aggregated account data comprising transactional data;

[0023] an aggregation engine for providing instruction to computer hardware for performing processes that aggregate data to the aggregation database;

[0024] a matching engine for providing operational instructions to computer hardware for matching strings of data representing transactions from within the aggregated data;

[0025] a transaction marker database comprising strings of data that are markers that have been determined to represent transactions, wherein the markers are stored in the transaction marker database for use by the matching engine; and

[0026] a service suggestion engine for providing suggested services based on transactions identified using the transaction marker database and the matching engine.

[0027] An implementation of a method for suggesting service based on previous transactions to a customer may comprise:

[0028] aggregating account data using an aggregation engine that instructs computer hardware in performing processes that aggregate data to an aggregation database within computer storage wherein the aggregation database is accessible over a computer network and wherein the aggregation database comprises aggregated account data comprising transactional data;

[0029] matching strings of data using a matching engine that provides operational instructions to computer hardware for matching strings of data representing transactions from within the aggregated data;

[0030] determining transaction markers within the aggregated data and storing the markers in a transaction marker database comprising strings of data that are markers that have been determined to represent transactions, wherein the markers are stored in the transaction marker database for use by the matching engine; and

[0031] suggesting services based on the strings of data representing transactions using a service suggestion engine that provides suggested services based on transactions identified using the transaction marker database and the matching engine.

[0032] FIG. 1 illustrates a computer environment and network for providing transaction identification from aggregated account data and services suggestions to a client. Illustrated in the figure is a user 105 that is electronically connected to financial institutions through a network 110. The financial institutions may comprise an aggregation engine 122 utilizing a server comprising transaction data 124. The system may further comprise a matching engine 132 utilizing computer hardware comprising received aggregated data, identified transaction data 161, and may further transmit the transactions for service suggestions.

[0033] As illustrated in the figure, the matching engine may perform the methods of receiving aggregation data at 151 and identifying transactions through string matching at 161, and presenting the identified transactions to a service suggestion engine at 171. The methods may be enabled by computer readable instructions stored in the memory of the computing environment.

[0034] As further illustrated in the FIG. 1, the system may comprise a service suggestion engine 144 running on computer hardware over a network, wherein the service suggestion engine utilizes a services database 146 comprising services available from financial institutions. The system may further comprise a transaction marker database 170 wherein unique identifiers/keys are stored representing various known or expected transaction types for use in optimizing the system as it performs additional iterations during use. Additionally, as can be seen in the figure, crowd sourcing 188 may be used to verify the nature of the transactions identified by the method.

[0035] Illustrated in FIG.2 is a schematic representation of computer hardware and protocols that enable the various embodiments discussed herein. Implementations of the present disclosure may comprise or utilize a special purpose or general-purpose computer, including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Implementations within the scope of the present disclosure may also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are computer storage media (devices). Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, implementations of the disclosure can comprise at least two distinctly different kinds of computer-readable media: computer storage media (devices) and transmission media.

[0036] Computer storage media (devices) includes RAM, ROM, EEPROM, CD-ROM, solid state drives (“SSDs”) (e.g., based on RAM), Flash memory, phase-change memory (“PCM”), other types of memory, other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0037] A “network” is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. In an implementation, a sensor and camera control unit may be networked in order to communicate with each other, and other components, connected over the network to which they are connected. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links, which can be used to carry desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

[0038] Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures that can be transferred automatically from transmission media to computer storage media (devices) (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile computer storage media (devices) at a computer system. RAM can also include solid state drives (SSDs or PCIx based real time memory tiered storage, such as FusionIO). Thus, it should be understood that computer storage media (devices) can be included in computer system components that also (or even primarily) utilize transmission media.

[0039] Computer-executable instructions comprise, for example, instructions and data which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0040] Those skilled in the art will appreciate that the disclosure may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, control units, camera control units, hand-held devices, hand pieces, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, tablets, pagers, routers, switches, various storage devices, and the like. It should be noted that any of the above mentioned computing devices may be provided by or located within a brick and mortar location. The disclosure may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0041] Further, where appropriate, functions described herein can be performed in one or more of: hardware, software, firmware, digital components, or analog components. For example, one or more application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) can be programmed to carry out one or more of the systems and procedures described herein. Certain terms are used throughout the following description and Claims to refer to particular system components. As one skilled in the art will appreciate, components may be referred to by different names. This document does not intend to distinguish between components that differ in name, but not function.

[0042] Illustrated in the figure is a block diagram illustrating an example computing device 200. Computing device 200 may be used to perform various procedures, such as those discussed herein. Computing device 200 can function as a server, a client, or any other computing entity. Computing device 200 can perform various monitoring functions as discussed herein, and can execute one or more application programs, such as the application programs described herein. Computing device 200 can be any of a wide variety of computing devices, such as a desktop computer, a notebook computer, a server computer, a handheld computer, camera control unit, tablet computer and the like.

[0043] Computing device 200 includes one or more processor(s) 202, one or more memory device(s) 204, one or more interface(s) 206, one or more mass storage device(s) 208, one or more Input/Output (I/O) device(s) 210, and a display device 230 all of which are coupled to a bus 212. Processor(s) 202 include one or more processors or controllers that execute instructions stored in memory device(s) 204 and/or mass storage device(s) 208. Processor(s) 202 may also include various types of computer-readable media, such as cache memory.

[0044] Memory device(s) 204 include various computer-readable media, such as volatile memory (e.g., random access memory (RAM) 214) and/or nonvolatile memory (e.g., read-only memory (ROM) 216). Memory device(s) 204 may also include rewritable ROM, such as Flash memory.

[0045] Mass storage device(s) 208 include various computer readable media, such as magnetic tapes, magnetic disks, optical disks, solid-state memory (e.g., Flash memory), and so forth. As shown in the figure, a particular mass storage device is a hard disk drive 224. Various drives may also be included in mass storage device(s) 208 to enable reading from and/or writing to the various computer readable media. Mass storage device(s) 208 include removable media 226 and/or non-removable media.

[0046] I/O device(s) 210 include various devices that allow data and/or other information to be input to or retrieved from computing device 200. Example I/O device(s) 210 include digital imaging devices, electromagnetic sensors and emitters, cursor control devices, keyboards, keypads, microphones, monitors or other display devices, speakers, printers, network interface cards, modems, lenses, CCDs or other image capture devices, and the like.

[0047] Display device 230 includes any type of device capable of displaying information to one or more users of computing device 200. Examples of display device 580 include a monitor, display terminal, video projection device, and the like.

[0048] Interface(s) 206 include various interfaces that allow computing device 200 to interact with other systems, devices, or computing environments. Example interface(s) 206 may include any number of different network interfaces 220, such as interfaces to local area networks (LANs), wide area networks (WANs), wireless networks, and the Internet. Other interface(s) include user interface 218 and peripheral device interface 222.

The interface(s) 206 may also include one or more user interface elements 218. The interface(s) 206 may also include one or more peripheral interfaces such as interfaces for printers, pointing devices (mice, track pad, etc.), keyboards, and the like.

[0049] Bus 212 allows processor(s) 202, memory device(s) 204, interface(s) 206, mass storage device(s) 208, and I/O device(s) 210 to communicate with one another, as well as other devices or components coupled to bus 212. Bus 212 represents one or more of several types of bus structures, such as a system bus, PCI bus, IEEE 1394 bus, USB bus, and so forth.

[0050] For purposes of illustration, programs and other executable program components are shown herein as discrete blocks, although it is understood that such programs and components may reside at various times in different storage components of computing device 200, and are executed by processor(s) 202. Alternatively, the systems and procedures described herein can be implemented in hardware, or a combination of hardware, software, and/or firmware. For example, one or more application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) can be programmed to carry out one or more of the systems and procedures described herein.

[0051] Illustrated in FIG. 3 is an implementation of a method for identifying transactions in aggregated account data. As can be seen in the figure, an implementation of methods for identifying transactions and providing additional services suggestions may comprise processes that: determine signatures for transaction types at 310 wherein strings of data may be identified and transformed into unique signatures that can then be used for categorization. It will be appreciated that a signature may be used to describe a category of the transaction or the transaction type. In an implementation, a signature may be used to recognize transactions types from within aggregated data comprising data relating to a plurality of transactions. In another implementation, a signature may be used for place holding when categorizing various transactions.

[0052] The method may then generate a unique identifier for signatures at 320 wherein the unique identifier may be a unique number or key that may be generated to create a finite deterministic qualifier for the majority of the signatures.

[0053] Additionally, an index of the unique identifiers may be created at 330. An index may comprise header metadata that may comprise the range of values contained within the index, and may additionally contain other identifying information that may be used within the system for optimization in parsing aggregated data.

[0054] An implementation may apply a bloom filter at 340 in order to preserve the efficiency of the system and prevent problematic queries from bogging down the system. Such queries may include transaction types that are not repetitious or may not fall within the intended index values, such that the bloom filter may read the index metadata and dismiss the query before searching and matching within the index.

[0055] Additionally, the system may comprise substituting the signatures for transaction data at 350 in order to reduce the amount of data processed and transmitted by the system. Such substitution may comprise hash functions for stream lining the data for mass processing in large aggregated databases. In an embodiment, the hash function may be predetermined based on collision expectations from hashing the transactions in the aggregated data.

[0056] The implementation may further comprise comparing transactions against the index of unique identifiers 355 and generating recognized patterns for transactions at 360. A simplified example of string matching and substitution is as follows:

Input string to match against dataset:

"APL*APPLE ITUNES STOR xxxxxxxx7753 CA 12/30"

[0057] The significant elements of the input string are the white space, number patterns, non-alphabetic characters (symbols), and alphabetic text. One derivation of the transformation of this string into a pattern is as follows:

replace all non-alpha/non-space/non-numeric characters with a space:

"APL APPLE iTUNES STOR xxxxxxxx7753 CA 12 30"

replace all repeating white-space with a single space:

"APL APPLE iTUNES STOR xxxxxxxx7753 CA 12 30"

replace all numerics with #:

"APL APPLE iTUNES STOR xxxxxxxx##### CA ## ##"

replace all repeating # which a single #:

"APL APPLE iTUNES STOR xxxxxxxx# CA # #"

standardize the capitalization (uppercase or lowercase):

"APL APPLE ITUNES STOR xxxxxxxx# CA # #"

[0058] This pattern is now able to be used in a unique index using known patterns for unique index matching and sorting/searching. The standardized transformation into a pattern based string allows a unique index to match a "fuzzy" pattern without having to calculate the "distance" or "fuzzy match" quotient as the pattern derived from a new input string must match exactly to be considered a match.

[0059] This method may be used in combination with presence (bloom) filters to allow a pattern's presence to be determined before referencing the index thus avoiding having to read the index off of disc.

[0060] The system may next present the patterns and transactions to crowd sourcing 365 for review and correction. In an implementation, a threshold may be predetermined for sending data for crowd sourcing. For example, a threshold of 88% accuracy may be determined to be acceptable for recognizing and categorizing, such that the system would not require the use of crowd sourcing.

[0061] In the implementation, the system may receive corrections from crowd sourcing at 367 and implement the correction within the transaction data. Additionally, the crowd source correction may be implemented into index of unique identifiers at 369 for greater optimization. Once the transactions have been identified, the system may suggest services based on the patterns of transactions at 370 and may present the suggestions to a user. It should be noted that the system may record the generated and newly discovered signatures as well as the resulting or processed data mentioned in each of the processes above in computer memory at 315.

[0062] Illustrated in FIG. 4 is an implementation of a method for identifying transactions in aggregated account data. As can be seen in the figure, an implementation of methods for identifying transactions and providing additional service suggestions using hash and string matching algorithms is illustrated and may comprise processes to: determine signatures for transaction types at 410 wherein strings of data may be

identified into unique signatures that can then be used for categorization. The method may then generate a unique identifier for signatures at 420, wherein the unique identifier may be a unique number or key that may be generated to create a finite deterministic qualifier for the majority of the signatures. Such signatures may be generated by hash functions at 422 for streamlining the data for mass processing in large aggregated databases. In an embodiment, the hash function may be predetermined based on collision estimations at 424 from hashing the transactions in the aggregated data.

[0063] Additionally, an index of the unique identifiers may be created at 430. An index may comprise header metadata that may comprise the range of values contained within the index, and may additionally contain other identifying information that may be used within the system for optimization in parsing aggregated data.

[0064] An implementation may apply a bloom filter 440 in order to preserve the efficiency of the system and prevent problematic queries from bogging down the system. Such queries may include transaction types that are not repetitious or may not fall within the intended index values, such that the bloom filter may read the index metadata and dismiss the query before searching and matching within the index.

[0065] Additionally, the system may comprise substituting the signatures for transaction data at 450 in order to reduce the amount of data processed and transmitted by the system. The substitution method may be predetermined at 452. Various substitution methods may be used that currently exist at the time of the disclosure, while future substitution methods yet to be developed are intended to fall within the scope of this disclosure.

[0066] The implementation may further comprise comparing transactions against the index of unique identifiers 455 and generate recognized patterns for transactions at 460. The comparisons may be facilitated by various string matching algorithms, such as the Levenshtein method and the like. The string matching algorithm may be determined at the outset or may be derived from an estimated accuracy at 456. Examples of known string matching processes are: Aho–Corasick string matching algorithm, Apostolico–Giancarlo algorithm, Bitap algorithm, Boyer–Moore string search algorithm, Boyer–Moore–Horspool algorithm, Commentz-Walter algorithm, Knuth–Morris–Pratt algorithm, Levenshtein automaton, Rabin–Karp algorithm, Raita Algorithm, and Zhu–Takaoka string matching algorithm.

[0067] Significant string fuzzy pattern matching methods may be used, such as for example, regular expressions and dynamic programming (Levenshtein), each of these can be optimized to be very fast, but still may require too much time and computational complexity overhead. This is because in the case of regular expressions that run fastest may still require that all words in a dataset to be compared against the regular expression pattern, and at the same time require tracking of the current state of the regular expression while testing each input character. With dynamic string matching, each potential distance measure is determined at runtime and the entire dataset must be compared against the input string. The previously mentioned pattern matching mechanisms can be very effective, but the buildup in computational complexity (time Big O) makes the comparison of large dataset impractical on a system, which should have guaranteed response time for matching long strings against large datasets.

[0068] The system may then present the patterns and transactions to crowd sourcing 465 for review and correction. In an implementation, a threshold may be predetermined for sending data for crowd sourcing. For

example, a threshold of 88% accuracy may be determined to be acceptable for recognizing and categorizing such that the system would not require the use of crowd sourcing.

[0069] In the implementation, the system may receive corrections from crowd sourcing at 467 and implement the correction within the transaction data. Additionally, the crowd source correction may be implemented into index of unique identifiers at 469 for greater optimization. Once the transactions have been identified, the system may suggest services based on the patterns of transactions at 470 and may present the suggestions to a user. It should be noted that the system may record the generated and newly discovered signatures as well as the resulting or processed data mentioned in each of the processes above in computer memory.

[0070] Illustrated in FIG. 5 is an implementation of a method for identifying transactions in aggregated account data wherein a bloom filter is applied during use. As can be seen in the figure, an implementation of methods for identifying transactions and providing additional services suggestions using hash and string matching algorithms is illustrated and may comprise processes to: determine signatures for transaction types at 510 wherein strings of data may be identified into unique signatures that can then be used for categorization. The method may then generate a unique identifier for signatures at 520 wherein the unique identifier may be a unique number or key that may be generated to create a finite deterministic qualifier for the majority of the signatures. Such signatures may be generated by hash functions for streamlining the data for mass processing in large aggregated databases. In an embodiment, the hash function may be predetermined based on collision estimations from hashing the transactions in the aggregated data.

[0071] Additionally, an index of the unique identifiers may be created at 530. An index may comprise header metadata that may comprise the range of values contained within the index, and may additionally contain other identifying information that may be used within the system for optimization in parsing aggregated data. In an implementation the unique identifier may be a hash code of the signature wherein the hash characters are place holders for specific transaction string data.

[0072] Additionally, in an implementation thresholds for matching may be determined by available memory for use by the matching engine.

[0073] An implementation may apply a bloom filter 540 in order to preserve the efficiency of the system and prevent problematic queries from bogging down the system. Such queries may include transaction types that are not repetitious or may not fall within the intended index values, such that the bloom filter may read the index metadata at 542 and dismiss the query before searching and matching within the index if it is not in range at 545.

[0074] Additionally, the system may comprise substituting the signatures for transaction data at 550 in order to reduce the amount of data processed and transmitted by the system. The implementation may further comprise comparing transactions against the index of unique identifiers 555 and generate recognized patterns for transactions at 560.

[0075] The system may then present the patterns and transactions to crowd sourcing 565 for review and correction. In an implementation, a threshold may be predetermined for sending data for crowd sourcing. For example, a threshold of 88% accuracy may be determined to be acceptable for recognizing and categorizing such that the system would not require the use of crowd sourcing.

[0076] In the implementation, the system may receive corrections from crowd sourcing at 567 and implement the correction within the transaction data. Additionally, the crowd source correction may be implemented into index of unique identifiers at 569 for greater optimization. Once the transactions have been identified, the system may suggest services based on the patterns of transactions at 570 and may present the suggestions to a user. It should be noted that the system may record the generated and newly discovered signatures as well as the resulting or processed data mentioned in each of the processes above in computer memory.

[0077] The foregoing description has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the disclosure to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. Further, it should be noted that any or all of the aforementioned alternate implementations may be used in any combination desired to form additional hybrid implementations of the disclosure.

[0078] Further, although specific implementations of the disclosure have been described and illustrated, the disclosure is not to be limited to the specific forms or arrangements of parts so described and illustrated. The scope of the disclosure is to be defined by the claims appended hereto, any future claims submitted here and in different applications, and their equivalents.

CLAIMS

1. A method for long string pattern matching of account data, the method comprising:

determining transaction signatures for each of multiple first strings of transaction data stored in a transaction database;

substituting the transaction signatures for each of the multiple first strings of transaction data;

generating unique identifiers for each of the transaction signatures;

storing the unique identifiers in an index on a nonvolatile computer-readable media;

determining metadata corresponding to the index of unique identifiers, the metadata comprising a range of values contained within the index;

identifying a second string of transaction data received from a third-party system;

determining whether the second string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata to avoid accessing the non-volatile computer-readable media;

in response to determining that the second string of transaction data cannot be dismissed as not matching any of the first strings of transaction data using only the metadata, performing long string pattern matching to determine whether the second string of transaction data matches any of the unique identifiers for the first strings of transaction data in the index stored on the nonvolatile computer-readable media;

in response to determining that the second string of transaction data matches a particular one of the unique identifiers for the first strings of transaction data in the index stored on the nonvolatile computer-readable media, assigning a transaction signature of the particular one of the first strings of transaction data to the second string of transaction data;

identifying a third string of transaction data received from a third-party system;

determining whether the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata to avoid accessing the non-volatile computer-readable media; and

in response to determining that the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata, dismissing the third string of transaction data as not matching any of the unique identifiers for the first strings of transaction data using only the metadata and without accessing the index stored on the nonvolatile computer-readable media.

2. The method of claim 1, wherein the identifying of the second string of transaction data and the identifying of the third string of transaction data are performed by aggregating the second string of transaction data and the third string of transaction data.

3. The method of claim 1, further comprising:

suggesting services to a user corresponding to the second string of transaction data based on the transaction signature assigned to the second string of transaction data.

4. The method of claim 1, further comprising generating a pattern for the third string of transaction data.

5. The method of claim 4, wherein the generating of the pattern for the third string of transaction data includes replacing all non-alpha, non-space, and non-numeric characters in the third string of transaction data with a space character.

6. The method of claim 5, wherein the generating of the pattern for the third string of transaction data further includes replacing all sequential repeats of space characters in the third string of transaction data with a single space character.

7. The method of claim 6, wherein the generating of the pattern for the third string of transaction data further includes replacing numeric characters in the third string of transaction data with a particular character that is a non-alpha, non-space, and non-numeric character.

8. The method of claim 7, wherein the generating of the pattern for the third string of transaction data further includes replacing all sequential repeats of the particular character in the third string of transaction data with a single one of the particular character.

9. The method of claim 8, wherein the generating of the pattern for the third string of transaction data further includes standardizing capitalization of alpha characters in the third string of transaction data to either all uppercase or all lowercase characters.

10. The method of claim 4, further comprising:

in response to determining that the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata, presenting the third string of transaction data and the generated pattern for the third string of transaction data to crowd sourcing in order to assign a transaction type to the third string of transaction data.

11. One or more non-transitory computer-readable media storing one or more programs that are configured, when executed, to cause one or more processors to perform the method of claim 1.

12. A method for long string pattern matching of account data, the method comprising:

determining transaction signatures for each of multiple first strings of transaction data stored in a transaction database;

substituting the transaction signatures for each of the multiple first strings of transaction data;

generating unique identifiers for each of the transaction signatures;

storing the unique identifiers in an index on a nonvolatile computer-readable media;

determining metadata corresponding to the index of unique identifiers, the metadata comprising a range of values contained within the index;

identifying a second string of transaction data received from a third-party system;

employing a bloom filter to determine whether the second string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata to avoid accessing the non-volatile computer-readable media;

in response to determining that the second string of transaction data cannot be dismissed as not matching any of the first strings of transaction data using only the metadata, performing long string pattern matching to determine whether the second string of transaction data matches any of the unique identifiers for the first strings of transaction data in the index stored on the nonvolatile computer-readable media;

in response to determining that the second string of transaction data matches a particular one of the unique identifiers for the first strings of transaction data in the index stored on the nonvolatile computer-readable media, assigning a transaction signature of the particular one of the first strings of transaction data to the second string of transaction data;

identifying a third string of transaction data received from a third-party system;

employing the bloom filter to determine whether the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata to avoid accessing the non-volatile computer-readable media; and

in response to determining that the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata, dismissing the third string of transaction data as not matching any of the unique identifiers for the first strings of transaction data using only the metadata and without accessing the index stored on the nonvolatile computer-readable media.

13. The method of claim 12, wherein:

the metadata includes a range of values of the first strings of transaction data contained within the index; and

the bloom filter is employed to dismiss the third string of transaction data where the second string of transaction data has a value that is not in the range of values.

14. The method of claim 12, wherein the identifying of the second string of transaction data and the identifying of the third string of transaction data are performed by aggregating the second string of transaction data and the third string of transaction data from data stored by a third party.

15. The method of claim 12, further comprising generating a pattern for the third string of transaction data.

16. The method of claim 15, wherein the generating of the pattern for the third string of transaction data includes replacing all non-alpha, non-space, and non-numeric characters in the third string of transaction data with a space character.

17. The method of claim 16, wherein the generating of the pattern for the third string of transaction data further includes replacing all sequential repeats of space characters in the third string of transaction data with a single space character.

18. The method of claim 17, wherein the generating of the pattern for the third string of transaction data further includes replacing numeric characters in the third string of transaction data with a particular character that is a non-alpha, non-space, and non-numeric character.

19. The method of claim 16, wherein the generating of the pattern for the third string of transaction data further includes replacing all sequential repeats of the particular character in the third string of transaction data with a single one of the particular character.

20. The method of claim 19, wherein the generating of the pattern for the third string of transaction data further includes standardizing capitalization of alpha characters in the third string of transaction data to either all uppercase or all lowercase characters.

21. The method of claim 15, further comprising:

in response to determining that the third string of transaction data can be dismissed as not matching any of the first strings of transaction data using only the metadata, presenting the third string of transaction data and the generated pattern for the third string of transaction data to crowd sourcing in order to assign a transaction signature to the third string of transaction data.

22. The method of claim 21, further comprising:

suggesting services to a user corresponding to the second string of transaction data based on the transaction signature assigned to the second string of transaction data and based on the transaction signature assigned to the third string of transaction data.

23. One or more non-transitory computer-readable media storing one or more programs that are configured, when executed, to cause one or more processors to perform the method of claim 12.

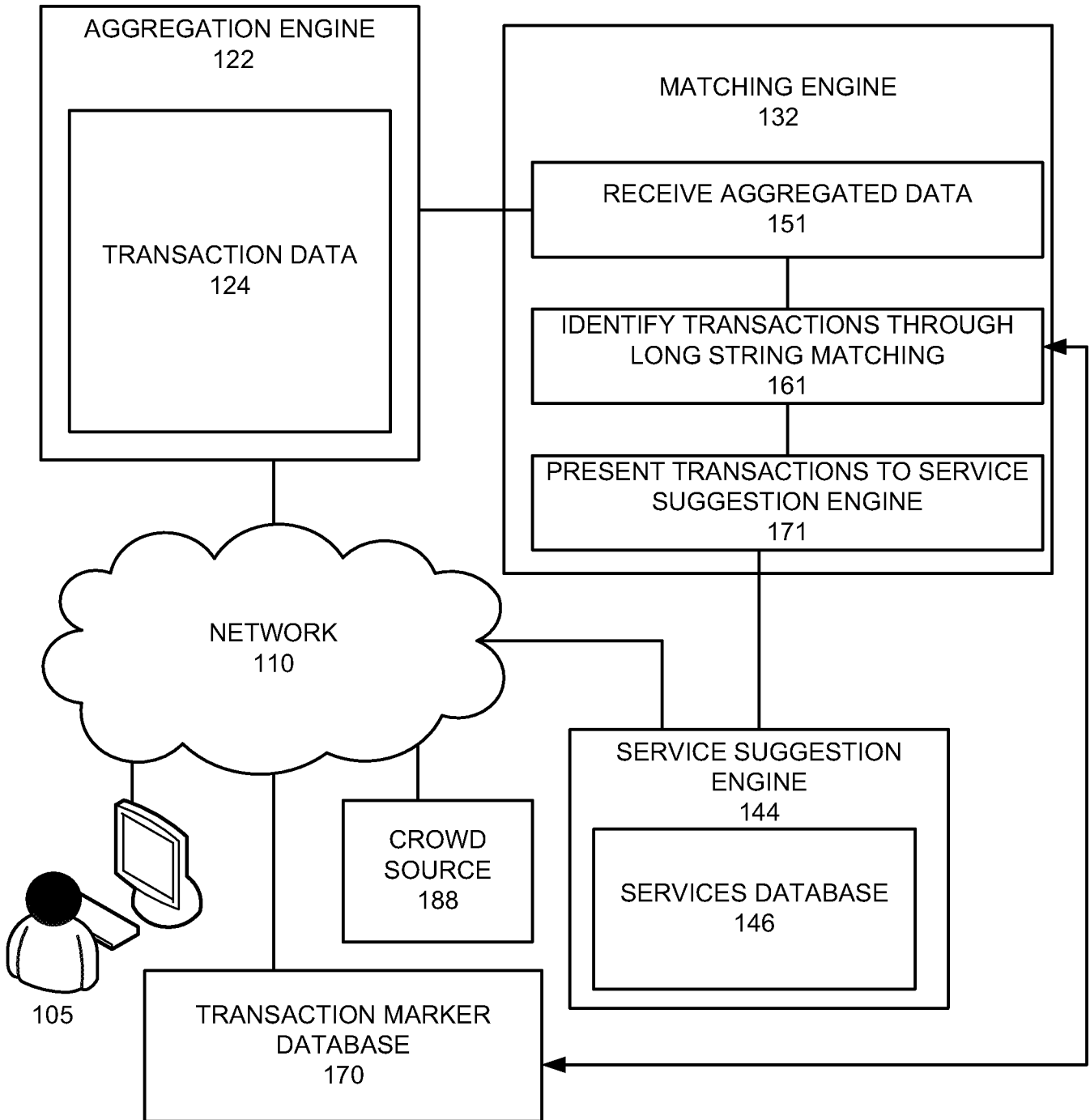


FIG. 1

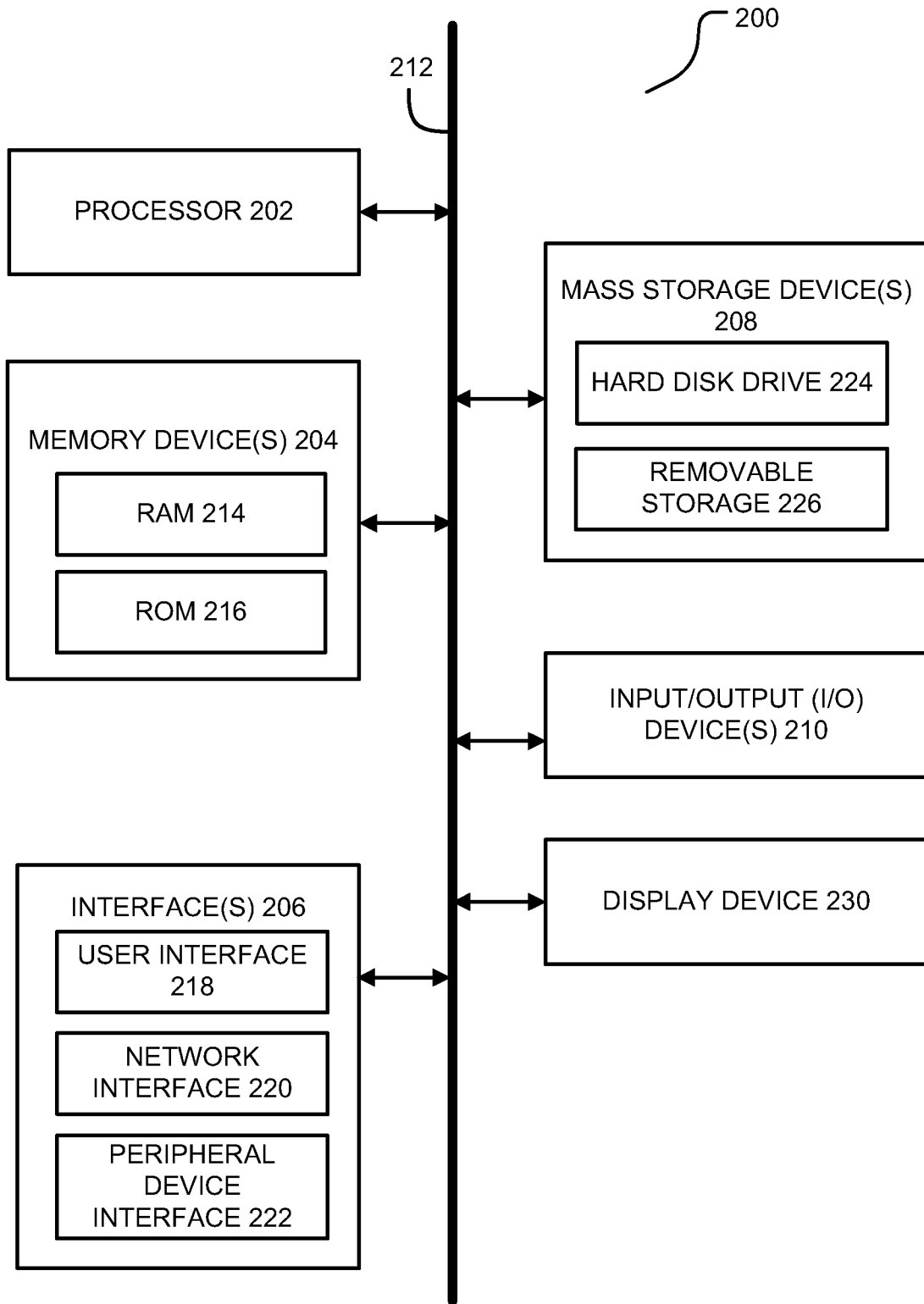
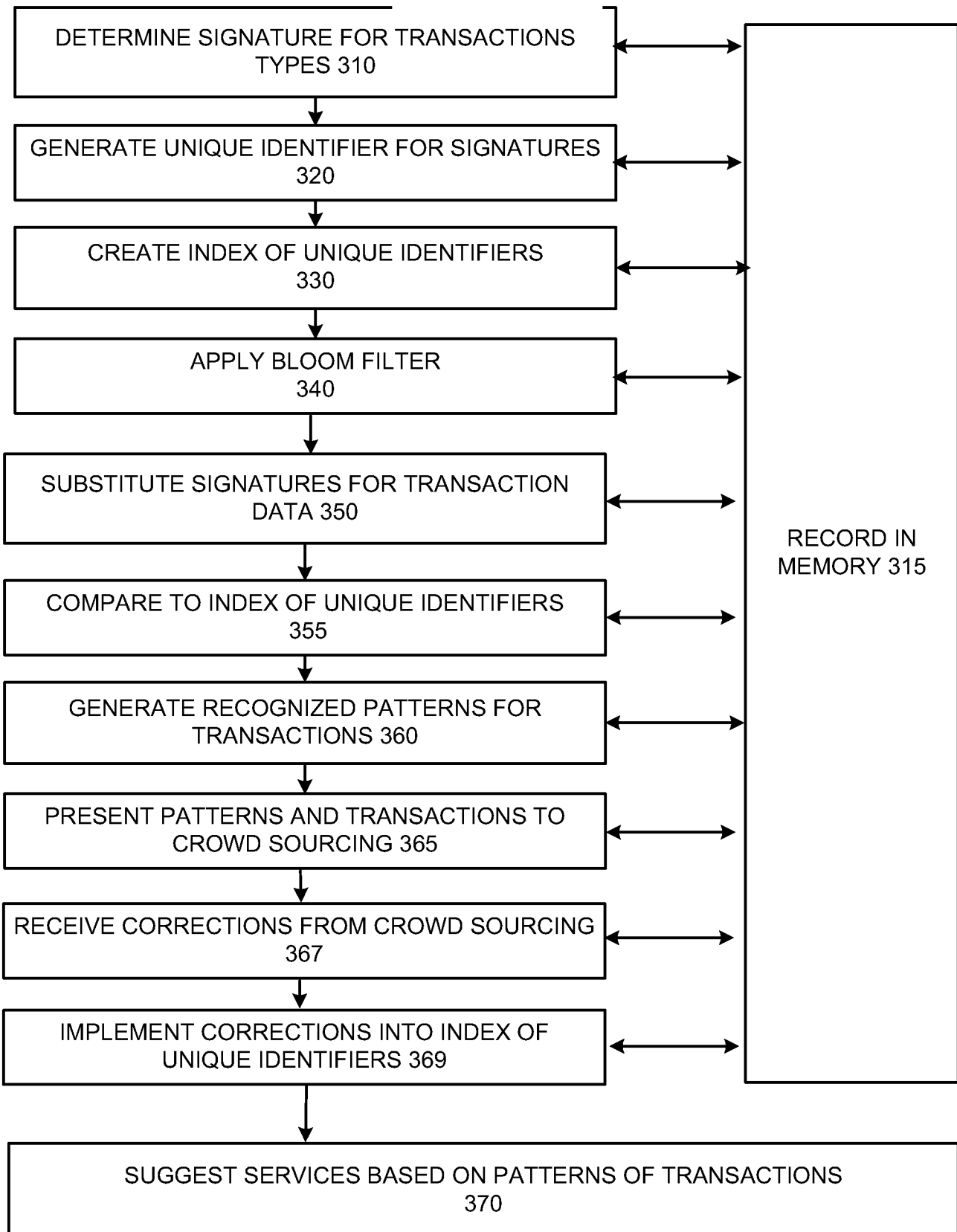
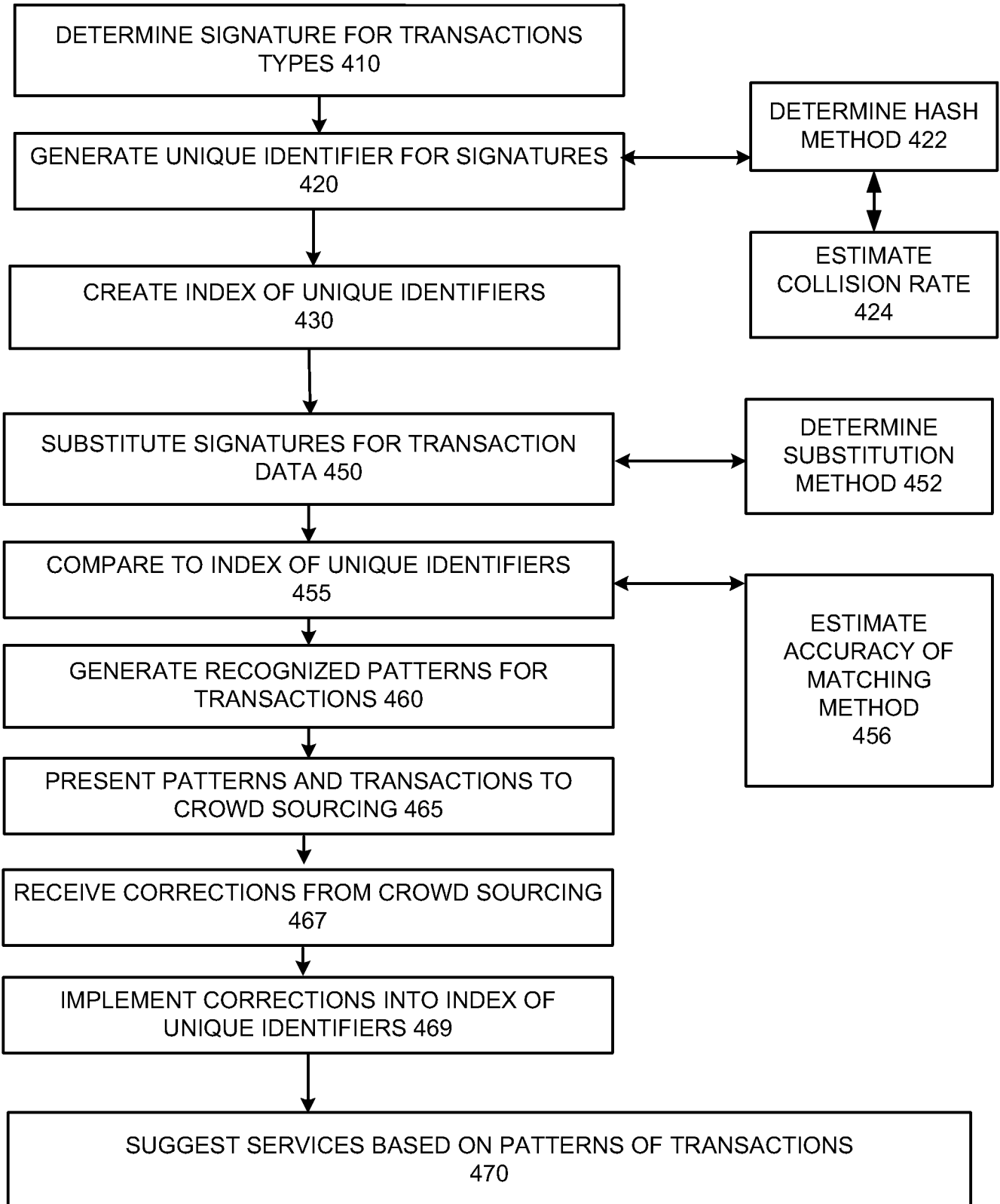
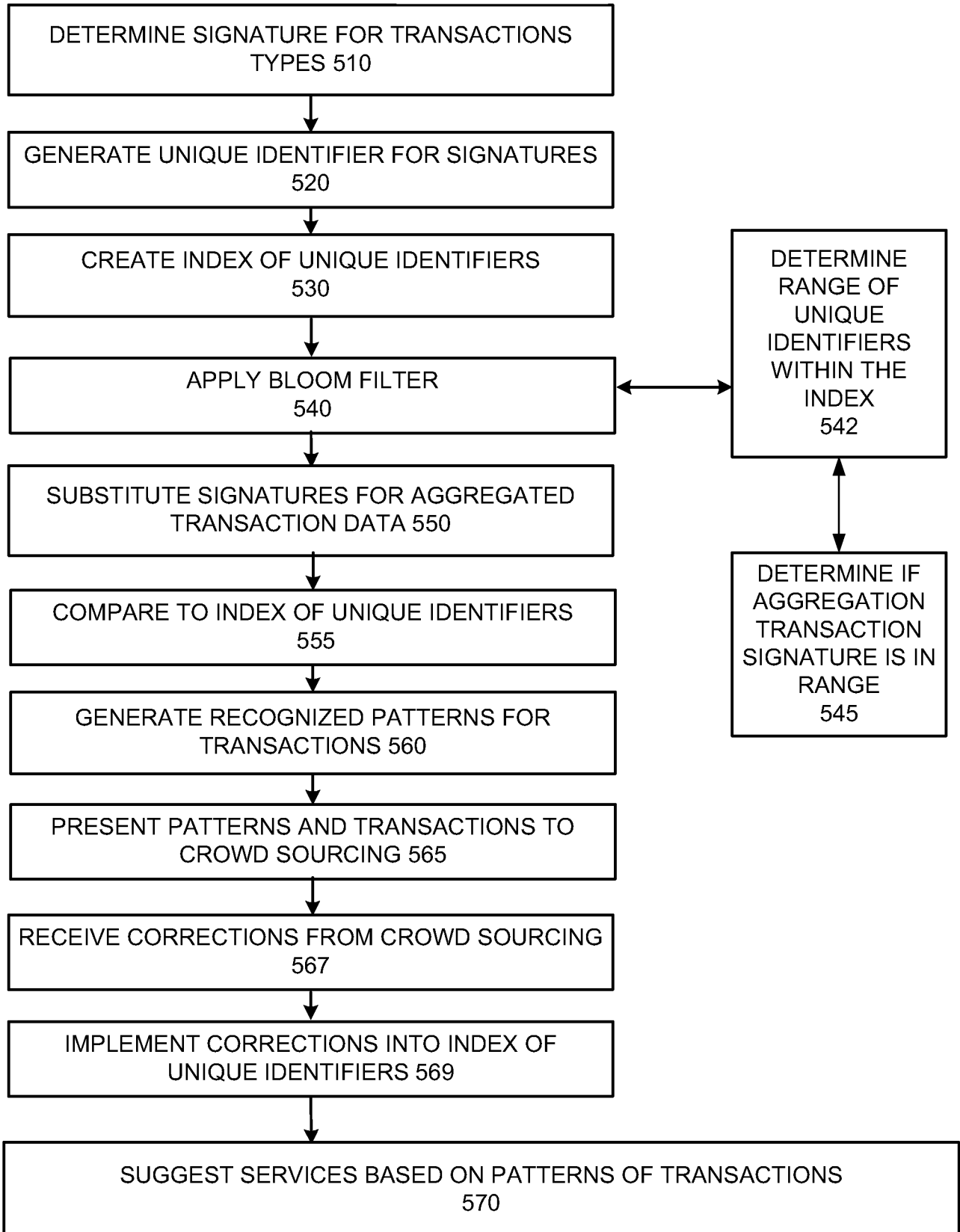


FIG. 2

**FIG. 3**

**FIG. 4**

**FIG. 5**

