

(51) International Patent Classification:  
G06F 17/30 (2006.01)(21) International Application Number:  
PCT/US2015/052228(22) International Filing Date:  
25 September 2015 (25.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
62/056,471 26 September 2014 (26.09.2014) US  
14/864,496 24 September 2015 (24.09.2015) US

(71) Applicant: ORACLE INTERNATIONAL CORPORATION [US/US]; 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US).

(72) Inventors: STOJANOVIC, Alexander Sasha; 14 W. Central Ave., Los Gatos, California 95030 (US). RIVAS, Luis E.; 8717 E. Frontier Pl., Denver, Colorado 80237 (US). OGREN, Philip; 3785 Smuggler Pl., Boulder, Colorado 80305 (US). MURRAY, Glenn Allen; 4365 Grinnell Ave., Boulder, Colorado 80305 (US).

(74) Agents: SHAH, Mehul M. et al.; Kilpatrick Townsend &amp; Stockton Llp, Two Embarcadero Center, Eighth Floor, San Francisco, California 94111 (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))

(54) Title: DECLARATIVE LANGUAGE AND VISUALIZATION SYSTEM FOR RECOMMENDED DATA TRANSFORMATIONS AND REPAIRS

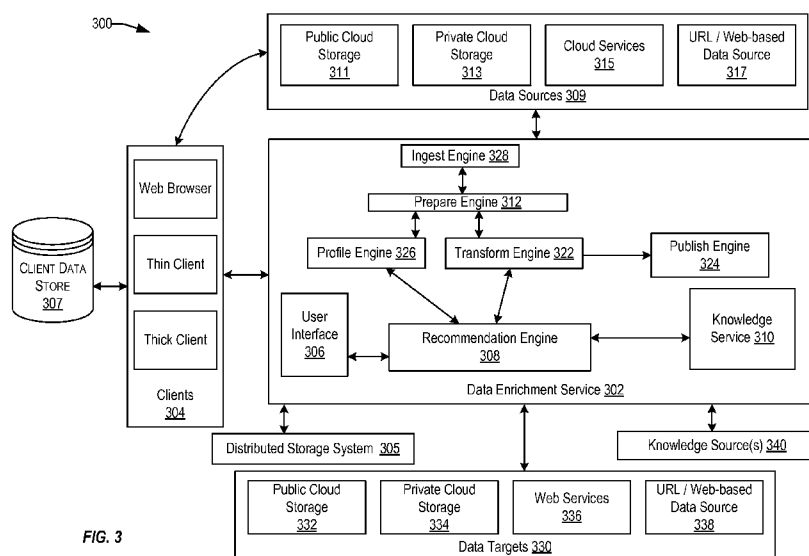


FIG. 3

(57) Abstract: The present disclosure relates generally to a data enrichment service that extracts, repairs, and enriches datasets, resulting in more precise entity resolution and correlation for purposes of subsequent indexing and clustering. As the data enrichment service can include a visual recommendation engine and language for performing large-scale data preparation, repair, and enrichment of heterogeneous datasets. This enables the user to select and see how the recommended enrichments (e.g., transformations and repairs) will affect the user's data and make adjustments as needed. The data enrichment service can receive feedback from users through a user interface and can filter recommendations based on the user feedback.

**DECLARATIVE LANGUAGE AND VISUALIZATION SYSTEM FOR  
RECOMMENDED DATA TRANSFORMATIONS AND REPAIRS**

**CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] The present application is claims the benefit and priority of U.S. Non-Provisional Patent Application No. 14/864,496, filed September 24, 2015, entitled “DECLARATIVE LANGUAGE AND VISUALIZATION SYSTEM FOR RECOMMENDED DATA TRANSFORMATIONS AND REPAIRS,” which claims the benefit and priority of U.S. Provisional Application No. 62/056,471, filed September 26, 2014, entitled “DECLARATIVE LANGUAGE AND VISUALIZATION SYSTEM FOR RECOMMENDED DATA TRANSFORMATIONS AND REPAIRS.”

[0002] The present application is related to the following applications:

- 1) U.S. Provisional Application No. 62/056,468, filed September 26, 2014 and titled “METHOD FOR SEMANTIC ENTITY EXTRACTION BASED ON GRAPH MATCHING WITH AN EXTERNAL KNOWLEDGEBASE AND SIMILARITY RANKING OF DATASET METADATA FOR SEMANTIC INDEXING, SEARCH, AND RETRIEVAL”;
- 2) U.S. Provisional Application No. 62/056,474, filed on September 26, 2014 and titled “DYNAMIC VISUAL PROFILING AND VISUALIZATION OF HIGH VOLUME DATASETS AND REAL-TIME SMART SAMPLING AND STATISTICAL PROFILING OF EXTREMELY LARGE DATASETS”;
- 3) U.S. Provisional Application No. 62/056,475, filed September 26, 2014 and titled “AUTOMATED ENTITY CORRELATION AND CLASSIFICATION ACROSS HETEROGENEOUS DATASETS”;
- 4) U.S. Provisional Application No. 62/056,476, filed September 26, 2014 and titled “DECLARATIVE EXTERNAL DATA SOURCE IMPORTATION, EXPORTATION, AND METADATA REFLECTION UTILIZING HTTP AND HDFS PROTOCOLS”;

5) U.S. Provisional Application No. 62/163,296, filed May 18, 2015 and titled “CATEGORY LABELING”; and

6) U.S. Provisional Application No. 62/203,806, filed August 11, 2015 and titled “SIMILARITY METRIC ANALYSIS AND KNOWLEDGE SCORING SYSTEM”.

5 [0003] The entire contents of the above-identified patent applications are incorporated herein by reference for all purposes.

#### BRIEF SUMMARY

[0004] The present disclosure relates generally to data preparation and analysis. More particularly, techniques are disclosed for generating and displaying interactive visualizations  
10 for recommended data transformations and repairs.

[0005] The present disclosure relates generally to a data enrichment service that extracts, repairs, and enriches datasets, resulting in more precise entity resolution and correlation for purposes of subsequent indexing and clustering. The data enrichment service can include a visual recommendation engine and language for performing large-scale data preparation,  
15 repair, and enrichment of heterogeneous datasets. This enables the user to select and see how the recommended enrichments (e.g., transformations and repairs) will affect the user’s data and make adjustments as needed. The data enrichment service can receive feedback from users through a user interface and can filter recommendations based on the user feedback.

[0006] In some embodiments, a computing system may be implemented for generating and  
20 displaying interactive visualizations for recommended data transformations and repairs. The computing system may implement a data enrichment service. The computing system may be configured to implement methods and operations described herein. In some embodiments, a system is disclosed for generating and displaying interactive visualizations for recommended data transformations and repairs. The system may include a plurality of data sources and a  
25 plurality of data targets. The system may include a cloud computing infrastructure system comprising one or more processors communicatively coupled to the plurality of data sources and communicatively coupled to the plurality of data targets, over at least one communication network. The cloud computing infrastructure system may include a memory coupled to the one or more processors, the memory storing instructions to provide a data enrichment service,  
30 where the instructions, when executed by the one or more processors, cause the one or more processors to perform one or more methods or operations described herein. Yet other

embodiments relate to systems and machine-readable tangible storage media, which employ or store instructions for methods and operations described herein.

5 [0007] In at least one embodiment, a method may include receiving a data enrichment request from a client device. The method may include identifying, by a computer system, patterns in data from one or more data sources. The method may include matching the patterns to entity information from a knowledge service. The method may include generating one or more transformation scripts for the data based on the entity information. The method may include generating one or more recommendations corresponding to the one or more transformation scripts. The method may include causing the one or more recommendations to be displayed  
10 in a user interface at the client device. The method may include receiving transformation instructions based on the one or more recommendations. The method may include transforming the data based on the transformation instructions. The method may include publishing the transformed data to one or more data targets based on the transformation instructions.

15 [0008] In some embodiments, the method may include receiving a selection of at least one column of data from at least one of the one or more data sources and causing a data profile of the at least one column of data to be displayed. The data profile may indicate a plurality of patterns identified in the at least one column of data and at least one data visualization associated with the at least one column of data.

20 [0009] In at least one embodiment, transformation instructions include a transformation instruction to rename at least one column of data based on the entity information. The method may include receiving a further transformation instruction to rename the at least one column of data to a default name. In some embodiments, transformation instructions include a transformation instruction to reformat at least one column of data based on the entity  
25 information. In some embodiments, transformation instructions include a further transformation instruction to obfuscate at least one column of data based on the entity information. In some embodiments, transformation instructions include an enrichment instruction to add one or more columns of data obtained from the knowledge service based on the entity information.

30 [0010] In one embodiment of the invention, when a selection of at least one column of data is received from at least one of the data sources, a data profile of the at least one column of data

is caused to be displayed in the user interface on the client device, where the data profile indicates a plurality of patterns identified in the at least one column of data and at least one data visualization associated with the at least one column of data.

[0011] In one embodiment of the invention, the transformation instructions include a transformation instruction to rename at least one column of data based on the entity information.

[0012] In one embodiment of the invention, a further transformation instruction is received to rename the at least one column of data to a default name.

[0013] In one embodiment of the invention, the transformation instructions include a transformation instruction to reformat at least one column of data based on the entity information.

[0014] In one embodiment of the invention, the transformation instructions include a further transformation instruction to obfuscate at least one column of data based on the entity information.

[0015] In one embodiment of the invention, the transformation instructions include an enrichment instruction to add one or more columns of data obtained from the knowledge service based on the entity information.

[0016] The foregoing, together with other features and embodiments will become more apparent upon referring to the following specification, claims, and accompanying drawings.

20

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 depicts a simplified high level diagram of a data enrichment service, in accordance with an embodiment of the present invention.

[0018] FIG. 2 depicts a simplified block diagram of a technology stack, in accordance with an embodiment of the present invention.

[0019] FIG. 3 depicts a simplified block diagram of an interactive visualization system, in accordance with an embodiment of the present invention.

[0020] FIGS. 4A-4D depict examples of a user interface that provides interactive data enrichment, in accordance with an embodiment of the present invention.

[0021] FIGS. 5A-5D depict examples of the results of interactive data enrichment, in accordance with an embodiment of the present invention.

[0022] FIG. 6 depicts a flowchart of a method of data enrichment, in accordance with an embodiment of the present invention.

5 [0023] FIG. 7 depicts a simplified diagram of a distributed system for implementing an embodiment.

[0024] FIG. 8 is a simplified block diagram of one or more components of a system environment in which services may be offered as cloud services, in accordance with an embodiment of the present disclosure.

10 [0025] FIG. 9 illustrates an exemplary computer system that may be used to implement an embodiment of the present invention.

#### DETAILED DESCRIPTION

[0026] In the following description, for the purposes of explanation, specific details are set forth in order to provide a thorough understanding of embodiments of the invention.

15 However, it will be apparent that various embodiments may be practiced without these specific details. The figures and description are not intended to be restrictive.

[0027] The present disclosure relates generally to a data enrichment service that extracts, repairs, and enriches datasets, resulting in more precise entity resolution and correlation for purposes of subsequent indexing and clustering. In some embodiments, the data enrichment  
20 service includes an extensible semantic pipeline, which processes the data in a number of stages, from ingestion of the data to analysis of the data, to publishing of the data-to-data targets.

[0028] In certain embodiments of the present invention, prior to loading data into a data warehouse (or other data target) the data is processed through a pipeline (also referred to  
25 herein as a semantic pipeline) which includes various processing stages. In some embodiments, the pipeline can include an ingest stage, prepare stage, profile stage, transform stage, and publish stage. During processing, the data can be analyzed, prepared, and enriched. The resulting data can then be published (e.g., provided to a downstream process) into one or more data targets (such as local storage systems, cloud-based storage services, web services,  
30 data warehouses, etc.) where various data analytics can be performed on the data. Because of

the repairs and enrichments made to the data, the resulting analyses produce useful results. Additionally, because the data onboarding process is automated, it can be scaled to process very large data sets that cannot be manually processed due to volume.

[0029] In some embodiments, data can be analyzed to extract entities from the data, and  
5 based on the extracted entities, the data can be repaired. For example, misspellings, address errors, and other common mistakes present a complex problem to big data systems. For small quantities of data, such errors can be manually identified and corrected. However, in very large data sets (e.g., billions of nodes or records) such manual processing is not possible. In certain embodiments of the present invention, the data enrichment service can analyze data  
10 using a knowledge service. Based on the contents of the knowledge service, entities in the data can be identified. For example, an entity can be an address, a business name, a location, a person name, an identification number, etc.

[0030] FIG. 1 depicts a simplified high-level diagram 100 of a data enrichment service, in accordance with an embodiment of the present invention. As shown in FIG. 1, a cloud-based  
15 data enrichment service 102 can receive data from various data sources 104. In some embodiments, a client can submit a data enrichment request to data enrichment service 102 which identifies one or more of the data sources 104 (or portions thereof, e.g., particular tables, datasets, etc.). The data enrichment service 102 may then request data to be processed from the identified data sources 104. In some embodiments, the data sources may be  
20 sampled, and the sampled data analyzed for enrichment, making large data sets more manageable. The identified data can be received and added to a distributed storage system (such as a Hadoop Distributed Storage (HDFS) system) accessible to the data enrichment service. The data may be processed semantically by a number of processing stages (described herein as a pipeline or semantic pipeline). These processing stages can include preparation  
25 stages 108, enrichment stages 110, and publishing stages 112. In some embodiments, data can be processed in one or more batches by the data enrichment services. In some embodiments, a streaming pipeline can be provided that processes data as it is received.

[0031] In some embodiments, a prepare stage 108 can include various processing sub-stages. This may include automatically detecting a data source format and performing content  
30 extraction and/or repair. Once the data source format is identified, the data source can be automatically normalized into a format that can be processed by the data enrichment service. In some embodiments, once a data source has been prepared, it can be processed by an enrich

stage 110. In some embodiments, inbound data sources can be loaded into a distributed storage system 105 accessible to the data enrichment service (such as an HDFS system communicatively coupled to the data enrichment service). The distributed storage system 105 provides a temporary storage space for ingested data files, which can then also provide  
5 storage of intermediate processing files, and for temporary storage of results prior to publication. In some embodiments, enhanced or enriched results can also be stored in the distributed storage system. In some embodiments, metadata captured during enrichment associated with the ingested data source can be stored in the distributed storage system 105. System level metadata (e.g., that indicates the location of data sources, results, processing  
10 history, user sessions, execution history, and configurations, etc.) can be stored in the distributed storage system or in a separate repository accessible to the data enrichment service.

[0032] In certain embodiments, the enrichment process 110 can analyze the data using a semantic bus (also referred to herein as a pipeline or semantic pipeline) and one or more  
15 natural language (NL) processors that plug into the bus. The NL processors can automatically identify data source columns, determine the type of data in a particular column, name the column if no schema exists on input, and/or provide metadata describing the columns and/or data source. In some embodiments, the NL processors can identify and extract entities (e.g., people, places, things, etc.) from column text. NL processors can also  
20 identify and/or establish relationships within data sources and between data sources. As described further below, based on the extracted entities, the data can be repaired (e.g., to correct typographical or formatting errors) and/or enriched (e.g., to include additional related information to the extracted entities).

[0033] In some embodiments, a publish stage 112 can provide data source metadata captured  
25 during enrichment and any data source enrichments or repairs to one or more visualization systems for analysis (e.g., display recommended data transformations, enrichments, and/or other modifications to a user). The publishing sub-system can deliver the processed data to one or more data targets. A data target may correspond to a place where the processed data can be sent. The place may be, for example, a location in memory, a computing system, a  
30 database, or a system that provides a service. For example, a data target may include Oracle Storage Cloud Service (OSCS), URLs, third party storage services, web services, and other cloud services such as Oracle Business Intelligence (BI), Database as a Service, and Database



Schema as a Service. In some embodiments, a syndication engine provides customers with a set of APIs to browse, select, and subscribe to results. Once subscribed and when new results are produced, the results data can be provided as a direct feed either to external web service endpoints or as bulk file downloads.

5 [0034] FIG. 2 depicts a simplified block diagram 200 of a technology stack, in accordance with an embodiment of the present invention. In some embodiments, the data enrichment service can be implemented using the logical technology stack shown in FIG. 2. The technology stack can include a user interface/experience (UX) layer 202 that provides access to the data enrichment service through one or more client devices (e.g., using a thin client,  
10 thick client, web browser, or other application executing on the client devices). A scheduler service 204 can manage requests/responses received through the UX layer and can manage the underlying infrastructure on which the data enrichment service executes.

[0035] In some embodiments, the processing stages described above with respect to FIG. 1, can include a number of processing engines. For example, the prepare processing stage 108  
15 can include ingest/prepare engines, a profiling engine and a recommendation engine. As data is ingested during prepare processing, the data (or samples thereof) can be stored in a distributed data storage system 210 (such as a “big data” cluster). The enrich processing stage 110 can include semantic/statistical engines, an entity extraction engine, and repair/transform engines. As described further below, the enrich processing stage 110 can  
20 utilize information obtained from knowledge service 206 during the enrichment process. Enrichment actions (e.g., the addition and/or transformation of data) can be performed on the data stored in the distributed storage system 210. Transformation of data may include modification to add missing data or data to enrich the data. Transformation of data may include modifying errors in the data or repairing the data. The publish processing stage 112  
25 can include a publish engine, a syndication engine, and a metadata results manager. In some embodiments, various open source technologies can be used to implement some functionality within the various processing stages and/or processing engines. For example, file format detection can use Apache Tika.

[0036] In some embodiments, a management service 208 can monitor changes made to the  
30 data during enrichment processing 110. The monitored changes can include tracking which users accessed the data, which data transformations were performed, and other data. This can enable the data enrichment service to roll back enrichment actions.

[0037] Technology stack 200 can be implemented in an environment such as a cluster 210 for big data operations ("Big Data Cluster"). Cluster 210 can be implemented using Apache Spark, which provides a set of libraries for implementing a distributed computing framework compatible with a distributed file system (DFS) such as HDFS. Apache Spark can send requests for map, reduce, filter, sort, or Sample cluster processing jobs to effective resource managers like YARN. In some embodiments, cluster 210 can be implemented using a distributed file system offering such as one offered by Cloudera®. The DFS, such as one offered by Cloudera®, may include HDFS and Yarn.

[0038] FIG. 3 depicts a simplified block diagram of data enrichment system 300, in accordance with an embodiment of the present invention. Data enrichment system 300 may implement a data enrichment service 302. Data enrichment service 302 can receive data enrichment requests from one or more clients 304. Data enrichment service 302 may comprise one or more computers and/or servers. Data enrichment service 302 may be a module that is comprised of several subsystems and/or modules, including some, which may not be shown. Data enrichment service 302 may have more or fewer subsystems and/or modules than shown in the figure, may combine two or more subsystems and/or modules, or may have a different configuration or arrangement of subsystems and/or modules. In some embodiments, data enrichment service 302 may include user interface 306, ingest engine 328, recommendation engine 308, knowledge service 310, profile engine 326, transform engine 322, a prepare engine 312, and publish engine 324. The elements implementing data enrichment service 302 may operate to implement a semantic processing pipeline as described above.

[0039] Data enrichment system 300 may include a semantic processing pipeline, in accordance with an embodiment of the present invention. All or part of the semantic processing pipeline may be implemented by data enrichment service 102. When a data source is added, the data source and/or the data stored thereon can be processed through a pipeline prior to loading the data source. The pipeline can include one or more processing engines that are configured to process the data and/or data source before publishing the processed data to one or more data targets. The processing engines can include an ingest engine that extracts raw data from the new data source and provides the raw data to a prepare engine. The prepare engine can identify a format associated with the raw data and can convert the raw data into a format (e.g., normalize the raw data) that can be processed by the

data enrichment service 302. A profile engine can extract and/or generate metadata associated with the normalized data and a transform engine can transform (e.g., repair and/or enrich) the normalized data based on the metadata. The resulting enriched data can be provided to the publish engine to be sent to one or more data targets. Each processing engine  
5 is described further below.

[0040] In some embodiments, data enrichment service 302 may be provided by a computing infrastructure system (e.g., a cloud computing infrastructure system). The computing infrastructure system may be implemented in a cloud computing environment having one or more computing systems. The computing infrastructure system may be communicatively  
10 coupled, over one or more communication networks, to one or more data sources or one or more data targets such as those described herein.

[0041] The clients 304 can include various client devices (such as desktop computers, laptop computers, tablet computers, mobile devices, etc.). Each client device can include one or more client applications 304 through which the data enrichment service 302 can be accessed.  
15 For example, a browser application, a thin client (e.g., a mobile app), and/or a thick client can execute on the client device and enable the user to interact with the data enrichment service 302. The embodiment depicted in FIG. 3 is merely an example and is not intended to unduly limit the claimed embodiments of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. For example, there may be  
20 more or fewer client devices than those shown in the figures.

[0042] The client devices 304 may be of various different types, including, but not limited to personal computers, desktops, mobile or handheld devices such as a laptop, a mobile phone, a tablet, etc., and other types of devices. A communication network facilitates communications between client devices 304 and data enrichment service 302. The communication network  
25 can be of various types and can include one or more communication networks. Examples of communication network 106 include, without restriction, the Internet, a wide area network (WAN), a local area network (LAN), an Ethernet network, a public or private network, a wired network, a wireless network, and the like, and combinations thereof. Different communication protocols may be used to facilitate the communications including both wired  
30 and wireless protocols such as IEEE 802.XX suite of protocols, TCP/IP, IPX, SAN, AppleTalk, Bluetooth, and other protocols. In general, the communication network may

include any communication network or infrastructure that facilitates communications between clients and data enrichment service 302.

[0043] A user can interact with the data enrichment service 302 through user interface 306. Clients 304 can render a graphical user interface to display the user's data, recommendations  
5 for transforming the user's data, and to send and/or receive instructions ("transformation instructions") to the data enrichment service 302 through user interface 306. The user interfaces disclosed herein, such as those references in FIGs. 4A-4D and 5A-5D may be rendered by data enrichment service 302 or via clients 304. For example, a user interface may be generated by user interface 306, and rendered by data enrichment service 302 at any  
10 one of clients 304. A user interface may be provided by data enrichment system 302 via network as part of a service (e.g., a cloud service) or a network-accessible application. In at least one example, an operator of a data enrichment service 302 may operate one of clients 304 to access and interact with any user interfaces disclosed herein. The user can send instructions to user interface 306 to add data sources (e.g., provide data source access and/or  
15 location information, etc.).

[0044] Data enrichment service 302 may ingest data using ingest engine 328. Ingest engine 328 can serve as an initial processing engine when a data source is added. The ingest engine 328 can facilitate safe, secure, and reliable uploading of user data from one or more data sources 309 into data enrichment service 302. In some embodiments, ingestion engine 328  
20 can extract data from the one or more data sources 309 and store it in a distributed storage system 305 in data enrichment service 302. Data ingested from one or more data sources 309 and/or one or more clients 304 can be processed as described above with respect to FIGS. 1-2 and stored in a distributed storage system 305. Data enrichment service 302 can receive data from a client data store 307 and/or from one or more data sources 309. The distributed  
25 storage system 305 can serve as temporary storage for the uploaded data during the remaining processing stages of the pipeline, prior to the data being published to one or more data targets 330. Once an upload is complete, the prepare engine 312 can be invoked to normalize the uploaded data set.

[0045] The received data may include structured data, unstructured data, or a combination  
30 thereof. Structure data may be based on data structures including, without limitation, an array, a record, a relational database table, a hash table, a linked list, or other types of data structures. As described above, the data sources can include a public cloud storage service

311, a private cloud storage service 313, various other cloud services 315, a URL or web-based data source 317, or any other accessible data source. A data enrichment request from the client 304 can identify a data source and/or particular data (tables, columns, files, or any other structured or unstructured data available through data sources 309 or client data store 5 307). Data enrichment service 302 may then access the identified data source to obtain the particular data specified in the data enrichment request. Data sources can be identified by address (e.g., URL), by storage provider name, or other identifier. In some embodiments, access to a data source may be controlled by an access management service. The client 304 may display a request to the user to input a credential (e.g., username and password) and/or to 10 authorize the data enrichment service 302 to access the data source.

[0046] In some embodiments, data uploaded from the one or more data sources 309 can be modified into various different formats. The prepare engine 312 can convert the uploaded data into a common, normalized format, for processing by data enrichment service 302. Normalizing may be performed by routines and/or techniques implemented using instructions 15 or code, such as Apache Tika distributed by Apache®. The normalized format provides a normalized view of data obtained from the data source. In some embodiments, the prepare engine 312 can read a number of different file types. Prepare engine 312 can normalize the data into a character separated form (e.g., tab separated values, comma separated values, etc.) or as a JavaScript Object Notation (JSON) document for hierarchical data. In some 20 embodiments, various file formats can be recognized and normalized. For example, standard file formats such as Microsoft Excel® formats (e.g., XLS or XLSX), Microsoft Word® formats (e.g., DOC or DOCX), and portable document format (PDF), and hierarchical formats like JSON and extended markup language (XML), can be supported. In some embodiments, various binary encoded file formats and serialized object data can also be read 25 and decoded. In some embodiments, data can be provided to the pipeline in Unicode format (UTF-8) encoding. Prepare engine 312 can perform context extraction and conversion to the file types expected by data enrichment service 302, and can extract document level metadata from the data source.

[0047] Normalizing a data set may include converting raw data in a data set into a format that 30 is processable by the data enrichment service 302, in particular profile engine 326. In one example, normalizing the data set to create a normalized data set includes modifying the data set having one format to an adjusted format as a normalized data set, the adjusted format

being different from the format. A data set may be normalized by identifying one or more columns of data in the data set, and modifying a format of the data corresponding to the columns to the same format. For example, data having different formatted dates in a data set may be normalized by changing the formats to a common format for the dates that can be  
5 processed by profile engine 326. Data may be normalized by being modified or converted from a non-tabular format to a tabular format, having one or more columns of data.

[0048] Once the data has been normalized, the normalized data can be passed to profile engine 326. The profile engine 326 can perform a column by column analysis of normalized data to identify the types of data stored in the columns and information about how the data is  
10 stored in the columns. In this disclosure, although profile engine 326 is described in many instances as performing operations on data, the data processed by profile engine 326 has been normalized by prepare engine 312. In some embodiments, the data processed by profile engine 326 may include data that is not normalized for being in a format (e.g., a normalized format) processable by profile engine 326. The output, or results, of profile engine 326 may  
15 be metadata (e.g., source profile) indicating profile information about the data from a source. The metadata may indicate one or more patterns about the data and/or a classification of the data. As further described below, the metadata may include statistical information based on analysis of the data. For example, profile engine 326 can output a number of metrics and pattern information about each identified column, and can identify schema information in the  
20 form of names and types of the columns to match the data.

[0049] The metadata generated by profile engine 326 may be used by other elements of data enrichment service, e.g., recommendation engine 308 and transformation engine 322, to perform operations as described herein for data enrichment service 302. In some embodiments, the profile engine 326 can provide metadata to a recommendation engine 308.

25 [0050] Recommendation engine 308 can identify repair, transform, and data enrichment recommendations for the data processed by profile engine 326. The metadata generated by profile engine 326 can be used to determine recommendations for data based on the statistical analysis and/or classifications indicated by the metadata. In some embodiments, recommendations can be provided to the user through a user interface or other web service.  
30 Recommendations can be tailored for business users, such that the recommendations describe at a high level what data repairs or enrichments are available, how those recommendations compare to past user activity, and/or how unknown items can be classified based on existing

knowledge or patterns. Knowledge service 310 can access one or more knowledge graphs or other knowledge sources 340. The knowledge sources can include publicly available information published by web sites, web services, curated knowledge stores, and other sources. Recommendation engine 308 can request (e.g., query) knowledge service 310 for  
5 data that can be recommended to a user for the data obtained for a source.

[0051] In some embodiments, transform engine 322 can present the user with the sampled data for each column, or sample rows from the input dataset through user interface 306. Through user interface 306, data enrichment service 302 may present a user with recommended transformations. The transformations may be associated with transformation  
10 instructions, which may include code and/or function calls to perform transformation actions. The transformation instructions may be invoked by a user based on selection at user interface 306, such as by selecting a recommendation for transformation or by receiving input indicating an operation (e.g., an operator command). In one example, transformation instructions include a transformation instruction to rename at least one column of data based  
15 on the entity information. A further transformation instruction can be received to rename the at least one column of data to a default name. A default name may include a name that is pre-determined. A default name may be any name that is pre-defined when a name for a column of data cannot be determined or is not defined. The transformation instructions can include a transformation instruction to reformat at least one column of data based on the entity  
20 information, and a transformation instruction to obfuscate at least one column of data based on the entity information. In some embodiments, the transformation instructions can include an enrichment instruction to add one or more columns of data obtained from the knowledge service based on the entity information.

[0052] Through user interface 306, a user can perform transform actions, and the transform  
25 engine 322 can apply them to the data obtained from a data source and display the results. This gives the user immediate feedback that can be used to visualize and verify the effects of the transform engine 322 configuration. In some embodiments, the transform engine 322 can receive pattern and/or metadata information (e.g., column names and types) from profile engine 326 and recommendation engine 308, which provides recommended transform  
30 actions. In some embodiments, transform engine 322 can provide a user event model that orchestrates and tracks changes to the data to facilitate undo, redo, delete, and edit events. The model can capture dependencies between actions so that the current configuration is kept

consistent. For example, if a column is removed, then recommended transform actions provided by the recommendation engine 308 for that column can also be removed. Similarly, if a transform action results in inserting new columns and that action is deleted, then any actions performed on the new columns are also deleted.

5 [0053] As described above, during processing the received data can be analyzed and a recommendation engine 308 can present one or more recommended transforms to be made to the data, including enrichment, repair, and other transforms. A recommended transform for enriching data may be comprised of a set of transforms, each transform of which is a single transform action, or an atomic transformation, performed on the data. A transform may be performed on data that was previously transformed by another transform in the set. The set of transforms may be performed in parallel or in a particular order, such that the data resulting after performing the set of transforms is enriched. The set of transforms may be performed according to a transform specification. The transform specification may include transformation instructions that indicate how and when to perform each of the set of transforms on the data produced by profile engine 326 and the recommendation for enriching the data determined by recommendation engine 308. Examples of the atomic transformation may include, without limitation, transforms to headers, conversions, deletions, splits, joins, and repairs. The data that is transformed according to the set of transforms may undergo a series of changes, each of which results in intermediate data the data is enriched. The data generated for intermediate steps for the set of transforms may be stored in a format such as an Resilient Distributed Dataset (RDD), text, a data record format, a file format, any other format, or a combination thereof.

[0054] In some embodiments, the data generated as a result of the operations performed by any elements of data enrichment service 302 may be stored in an intermediary data format including, but not limited to, RDD, text, a document format, any other type of format, or a combination thereof. The data stored in the intermediary format may be used to further perform operations for data enrichment service 302.

[0055] The following tables illustrate examples of transformations. Table 1 shows an outline of types of transforms actions.

30



Transform Types	Function Parameter(s)	Description	Examples
Update	String => String	Update column values	Obfuscate, date format,
Split	String => Array[String]	Split a column's values into new columns	Regex split, delimiter split
Filter	String => Boolean	Filter rows based on a single column's values	White list filtering, date range filtering
Multi-column Filter	Array[String] => Boolean	Filter rows based on multiple column values	NER false positives filtering
Edit Columns	Array[String] => Array[String]	Edit the existing columns	Reorder, remove, swap columns
Extract	(String, String) => Array[Array[String]]	Extract values from a column into a new RDD	NER with results extracted to a new table
Insert	String => Array[String]	Insert new columns	Insert timestamp
Insert 1:M	String => Array[Array[String]]	Insert new columns in a one-to-many way	Insert NER results

Table 1

[0056] Table 2 shows transform actions that do not fit within the category types shown with reference to Table 1.

Transform Actions	Description
Rename column	Rename a column
Sample	Replace the current RDD with a sample of it
Join	Performs a left-outer-join between two RDDs
Export	Export an RDD as columnar data to e.g. HDFS

Table 2

- [0057] Table 3 below shows examples of types of transform examples. Specifically Table 3 shows examples of transform actions and describes the type of transformations corresponding to those actions. For example, a transform action may include filtering data based on detecting the presence of words from a white list in data. If a user wants to track communications (e.g., tweets) containing “Android” or “iPhone”, a transform action could be added with those two words comprising the provided white list. This is just one example of the way by which data could be enriched for a user.

Transform Actions	Description	Input	Output	R1
Obfuscate	Obfuscate sensitive information such as e.g. credit card numbers, ID's, or birth dates	123-45-6789	###-##-####	Y
Date Reformat	Reformat a column containing a date	1330978536 2012-03-12 14:13:49	March 05, 2012 03/12/12 02:13:49 PM	Y
Rename Column	Rename a column	tagged_0001 text_label_0005	user_agent call_letters	Y
NER	Perform named entity	PopBooth turns your iPhone or	Type: Product Text: PopBooth,	Y

	recognition and insert values (see next section)	iPad into a photo booth, prints and all	iPhone, iPad	
Search / Replace	Perform search and replace on a column's values	Search: Mozilla Replace: Godzilla Value: Mozilla 5.0	Value: Godzilla 5.0	Y
Change case	Change the case to lower, upper, or proper	Case: Proper Value: eden prairie	Value: Eden Prairie	Y
White list filter	Filter rows based on the presence of words from a white list in a text-valued column	List: Android, iPhone Value: I heart my iPhone	Keep all rows whose values contain "Android" or "iPhone"	Y

Table 3

[0058] The recommendation engine 308 can use information from a knowledge service 310, knowledge source 340 to generate recommendations for transform engine 322 and to instruct transform engine 322 to generate transform scripts that will transform the data. Transform scripts may include programs, code, or instructions that may be executable by one or more processing units to transform received data. As such, the recommendation engine 308 can serve as an intermediary between the user interface 306 and the knowledge service 310.

[0059] As discussed above, profile engine 326 can analyze data from a data source to determine whether any patterns exist, and if so, whether a pattern can be classified. Once data obtained from a data source is normalized, the data may be parsed to identify one or more attributes or fields in the structure of the data. Patterns may be identified using a collection of regular expressions, each having a label ("tag") and being defined by a category.

The data may be compared to different types of patterns to identify a pattern. Examples of pattern types that can be identified include, without limitation, integers, decimals, dates or date/time strings, URLs, domain addresses, IP addresses, email addresses, version numbers, locale identifiers, UUIDs and other hexadecimal identifiers, social security numbers, US box  
5 numbers, typical US street address patterns, zipcodes, US phone numbers, suite numbers, credit card numbers, proper names, personal information, and credit card vendors.

[0060] In some embodiments, profile engine 326 may identify patterns in data based on a set of regular expressions defined by semantic constraints or syntax constraints constraints. A regular expression may be used to determine the shape and/or structure of data. Profile  
10 engine 326 may implement operations or routines (e.g., invoke an API for routines that perform processing for regular expressions) to determine patterns in data based on one or more regular expressions. For example, a regular expression for a pattern may be applied to data based on syntax constraints to determine whether the pattern is identifiable in the data.

[0061] Profile engine 326 may perform parsing operations using one or more regular  
15 expressions to identify patterns in data processed by profile engine 326. Regular expressions may be ordered according to a hierarchy. Patterns may be identified based on order of complexity of the regular expressions. Multiple patterns may match data that is being analyzed; the patterns having the greater complexity will be selected. As described further below, profile engine 326 may perform statistical analysis to disambiguate between patterns  
20 based on the application of regular expressions that are applied to determine those patterns.

[0062] In some embodiments, data that is unstructured may be processed to analyze metadata-describing attributes in the data. The metadata itself may indicate information about the data. The metadata may be compared to identify similarities and/or to determine a type of the information. The information identified based on the data may be compared to  
25 know types of data (e.g., business information, personal identification information, or address information) to identify the data that corresponds to a pattern.

[0063] In accordance with an embodiment, the profile engine 326 may perform statistical analysis to disambiguate the patterns and/or the text in data. Profile engine 326 may generate metadata including statistical information based on the statistical analysis. When patterns are  
30 identified, profile engine 326 may determine statistical information (e.g., a pattern metric) about each different pattern to disambiguate between the patterns. The statistical information

may include a standard deviation for different patterns that are recognized. The metadata including the statistical information can be provided to other components of data enrichment service 302, such as recommendation engine 308. For example, the metadata may be provided to recommendation engine 308 to enable recommendation engine 308 to determine  
5 recommendations for enrichment of the data based on the identified the pattern(s). Recommendation engine 308 can use the patterns to query a knowledge service 310 to obtain additional information about the patterns. Knowledge service 310 can include or have access to one or more knowledge sources 340. A knowledge sources can include publicly available information published by web sites, web services, curated knowledge stores, and other  
10 sources.

**[0064]** Profile engine 326 may perform the statistical analysis to disambiguate between patterns identified in the data. For example, data analyzed by profile engine 326, may be evaluated to compute a pattern metric (e.g., a statistical frequency of different patterns in the data) for each of the different patterns identified in the data. Each of the set of pattern  
15 metrics is computed for a different pattern of the patterns that are identified. Profile engine 326 may determine a difference amongst the pattern metrics computed for the different patterns. One of the identified patterns may be selected based on the difference. For example, one pattern may be disambiguated from another pattern based on a frequency of the patterns in the data. In another example, where the data consists of dates having multiple  
20 different formats, each corresponding to a different pattern, profile engine 326 may convert the dates to a standard format in addition to normalization and may then determine a standard deviation for each format from different patterns. In this example, profile engine 326 may statistically disambiguate between the formats in the data as having the format with the lowest standard deviation. The pattern corresponding to the format of the data having the  
25 lowest standard deviation may be selected as the best pattern for the data.

**[0065]** Profile engine 326 may determine a classification of a pattern that it identifies. Profile engine 326 may communicate with knowledge service 310 to determine whether the identified pattern can be classified within a knowledge domain. Knowledge service 310 may determine one or more possible domains associated with the data based on techniques  
30 described herein such as matching techniques and similarity analysis. Knowledge service 310 may provide profile engine 326 with a classification of one or more domains possibly similar to data identified with a pattern. Knowledge service 310 may provide, for each of the

domains identified by knowledge service 310, a similarity metric indicating a degree of similarity to the domain. The techniques disclosed herein for similarity metric analysis and scoring can be applied by recommendation engine 308 to determine a classification of data processed by profile engine 326. The metadata generated by profile engine 326 may include  
5 information about the knowledge domain, if any are applicable, and a metric indicating a degree of similarity with the data analyzed by profile engine 326.

[0066] Profile engine 326 may perform statistical analysis to disambiguate text identified in data, regardless of whether patterns are identified in the data. The text may be part of a pattern, and the analysis of the text may be used to further identify a pattern, if any can be  
10 identified. Profile engine 326 may request knowledge service 310 to perform domain analysis on text to determine whether the text can be classified into one or more domains. Knowledge service 310 may operate to provide information about one or more domains that are applicable to the text being analyzed. Analysis performed by knowledge service 310 to determine a domain may be performed using techniques described herein, such as similarity  
15 analysis used to determine a domain for data.

[0067] In some embodiments, profile engine 326 may identify text data in a data set. The text data may correspond to each entity identified in the set of entities. A classification may be determined for each entity that is identified. Profile engine 326 may request knowledge service to identify a classification for the entity. Upon determining a set of classifications  
20 for a set of entities (e.g., entities in a column), profile engine 326 may compute a set of metrics ("classification metrics") to disambiguate between the set of classifications. Each of the set of metrics may be computed for a different one of the set of classifications. Profile engine 326 may statistically disambiguate the set of metrics by comparing them to each other to determine which classification most closely represents the set of entities. A classification  
25 of the set of entities may be chosen based on the classification that represents the set of entities.

[0068] Using the knowledge sources 340, knowledge service 310 can match, in context, the patterns identified by profile engine 326. Knowledge service 310 may compare the identified patterns in the data or the data if in text to entity information for different entities stored by a  
30 knowledge source. The entity information may be obtained from one or more knowledge sources 340 using knowledge service 310. Examples of known entity may include social security numbers, telephone numbers, address, proper names, or other personal information.

The data may be compared to entity information for different entities to determine if there is a match with one or more entities based on the identified pattern. For example, the knowledge service 310 can match the pattern “XXX-XX-XXXX” to the format of U.S. social security numbers. Furthermore, the knowledge service 310 can determine that social security  
5 numbers are protected or sensitive information, the disclosure of which is linked to various penalties.

[0069] In some embodiments, profile engine 326 can perform statistical analysis to disambiguate between multiple classifications identified for data processed by profile engine 326. For example, when text is classified with multiple domains, profile engine 326 can  
10 process the data to statistically determine the appropriate classification determined by knowledge service 310. The statistical analysis of the classification can be included in the metadata generated by profile engine 326.

[0070] In addition to pattern identification, profile engine 326 can analyze data statistically. The profile engine 326 can characterize the content of large quantities of data, and can  
15 provide global statistics about the data and a per-column analysis of the data’s content: e.g., its values, patterns, types, syntax, semantics, and its statistical properties. For example, numeric data can be analyzed statistically, including, e.g., N, mean, maximum, minimum, standard deviation, skewness, kurtosis, and/or a 20-bin histogram if N is greater than 100 and unique values is greater than K. Content may be classified for subsequent analysis.

[0071] In one example, global statistics may include, without restriction, the number of rows, the number of columns, the number of raw and populated columns and how they varies, distinct and duplicate rows, header information, the number of columns classified by type or subtype, and the number of columns with security or other alerts. Column-specific statistics may include populated rows (e.g., K-most frequent, K-least frequent unique values, unique  
25 patterns, and (where applicable) types), frequency distributions, text metrics (e.g., minimum, maximum, mean values of: text length, token count, punctuation, pattern-based tokens, and various useful derived text properties), token metrics, data type and subtype, statistical analysis of numeric columns, L-most/least probable simple or compound terms or n-grams found in columns with mostly unstructured data, and reference knowledge categories  
30 matched by this naive lexicon, date/time pattern discovery and formatting, reference data matches, and imputed column heading label.

[0072] The resulting profile can be used to classify content for subsequent analyses, to suggest, directly or indirectly, transformations of the data, to identify relationships among data sources, and to validate newly acquired data before applying a set of transformations designed based on the profile of previously acquired data.

5 [0073] The metadata produced by profile engine 326 can be provided to the recommendation engine 308 to generate one or more transform recommendations. The entities that match an identified pattern of the data can be used to enrich the data with those entities identified by classification determined using knowledge service 310. In some embodiments, the data to the identified patterns (e.g., city and state) may be provided to knowledge service 310 to  
10 obtain, from a knowledge source 340, entities that match the identified patterns. For example, knowledge service 310 may be invoked calling a routine (e.g., getCities() and getStates()) corresponding to the identified patterns to receive entity information. The information received from knowledge service 310 may include a list (e.g., canonical list) of entities that have properly spelled information (e.g., properly spelled cities and states) for the  
15 entities. Entity information corresponding to matching entities obtained from knowledge service 310 can be used to enrich data, e.g., normalize the data, repair the data, and/or augment the data.

[0074] In some embodiments, the recommendation engine 308 can generate transform recommendations based on the matched patterns received from the knowledge service 310.  
20 For example, for the data including social security numbers, the recommendation engine can recommend a transform that obfuscates the entries (e.g., truncating, randomizing, or deleting, all or a portion of the entries). Other examples of transformation may include, reformatting data (e.g., reformatting a date in data), renaming data, enriching data (e.g., inserting values or associating categories with data), searching and replacing data (e.g., correcting spelling of  
25 data), change case of letter (e.g., changing a case from upper to lower case), and filter based on black list or white list terms. In some embodiments, recommendations can be tailored for particular users, such that the recommendations describe at a high level what data repairs or enrichments are available. For example, an obfuscation recommendation may indicate that the first five digits of the entries will be deleted. In some embodiments, the  
30 recommendations can be generated based on past user activity (e.g., provide a recommended transform that was previously used when sensitive data was identified)



[0075] Transform engine 322 can generate transform scripts based on the recommendations provided by recommendation engine 308 (e.g., a script to obfuscate the social security numbers). A transform script may perform an operation to transform data. In some embodiments, a transform script may implement a linear transformation of data. A linear transformation may be implemented through use of an API (e.g., Spark API). The transform actions may be performed by operations invoked using the API. A transform script may be configured based on transform operations defined using the API. The operations may be performed based on the recommendations.

[0076] In some embodiments, the transform engine 322 can automatically generate transform scripts to repair data at the data source. Repairs may include automatically renaming columns, replacing strings or patterns within a column, modifying text case, reformatting data, etc. For example, the transform engine 322 can generate a transformation script to transform a column of dates based on a recommendation from recommendation engine 308 to modify, or convert, the formats of the dates in the column. The recommendation may be selected from multiple recommendations to enrich or modify the data from a data source that is processed by profile engine 326. The recommendation engine 308 may determine the recommendation based on metadata, or profile, provided by the profile engine 326. The metadata may indicate a column of dates identified for different formats (e.g., MM/DD/YYYY, DD-MM-YY, etc.). The transform script generated by transform engine 322 can, for example, split and/or join columns based on suggestions from the recommendation engine 308. The transform engine 322 may also remove columns based on the data source profiles received from profile engine 326 (e.g., to remove empty columns, or columns that include information that is not desired by the user).

[0077] A transform script may be defined using a syntax that describes operations with respect to one or more algorithms (e.g., Spark Operator Trees). As such, the syntax may describe operator-tree transduction/reduction. A transform script may be generated based on a chosen recommendation or requested by a user interactively through a graphical user interface. Examples of recommended transformations are described with reference to FIGS. 4A, 4B, 4C, and 4D. Based on the transform operations specified by a user through the graphical user interface, the transform engine 322 performs transform operations according to those operations. The transform operations may be recommended to the user to enrich a data set.

[0078] As described further below, the clients 304 can display recommendations describing or otherwise indicating each recommended transform. When a user selects a transform script to be run, the selected transform script can be run on all or more of the data from the data source in addition to the data analyzed to determine the recommended transform(s). The  
5 resulting transformed data can then be published to one or more data targets 330 by publish engine 324. In some embodiments, the data targets can be different data stores than the data sources. In some embodiments, the data targets can be the same data stores as the data sources. Data targets 330 can include a public cloud storage service 332, a private cloud storage service 334, various other cloud services 336, a URL or web-based data target 338, or  
10 any other accessible data target.

[0079] In some embodiments, recommendation engine 308 can query knowledge service 310 for additional data related to the identified platform. For example, where the data includes a column of city names, related data (e.g., location, state, population, country, etc.) can be identified and a recommendation to enrich the dataset with the related data can be presented.  
15 Examples of presenting recommendations and transforming data through a user interface are shown below with respect to FIGS. 4A-4D.

[0080] Knowledge service 310 can implement a matching method to compare the data to reference data available through knowledge service 310. Knowledge service 310 can include or have access to one or more knowledge sources 340. The knowledge sources can include  
20 publicly available information published by web sites, web services, curated knowledge stores, and other sources. Knowledge service 310 can implement a method to determine the semantic similarity between two or more datasets. This may also be used to match the user's data to reference data available through the knowledge service 310. Knowledge service 310 may perform similarity metric analysis as described in this disclosure. The techniques  
25 performed by knowledge service 310 may include those described in this disclosure including the techniques described by the references incorporated herein.

[0081] Knowledge service 310 can perform operations to implement automated data analyses. In some embodiments, knowledge service 310 can use an unsupervised machine learning tool, such as Word2Vec, to analyze an input data set. Word2Vec can receive a text  
30 input (e.g., a text corpus from a large data source) and generate a vector representation of each input word. The resulting model may then be used to identify how closely related are an arbitrary input set of words. For example, a Word2Vec model built using a large text corpus

(e.g., a news aggregator, or other data source) can be utilized to determine corresponding numeric vector for each input word. When these vectors are analyzed, it may be determined that the vectors are “close” (in the Euclidean sense) within a vector space. Although this can identify that input words are related (e.g., identifying input words that are clustered closely together within a vector space), Word2Vec may not be usable to identify a descriptive label for the words (e.g., “tire manufacturers”). Knowledge service 310 may implement operations to categorize the related words using a curated knowledge source 340 (e.g., YAGO, from the Max Planck Institute for Informatics). Using information from a knowledge source 340, knowledge service 310 can add additional, related data to the input data set.

10 [0082] In some embodiments, knowledge service 310 may implement operations to perform trigram modeling to further refine categorization of related terms. Trigram modeling can be used to compare sets of words for category identification. The input data set can be augmented with the related terms.

[0083] Using the input data set, which may include added data, knowledge service 310 can implement matching methods (e.g., a graph matching method) to compare the words from the augmented data set to categories of data from knowledge source 340. Knowledge service 310 can implement a method to determine the semantic similarity between the augmented data set and each category in knowledge source 340 to identify a name for the category. The name of the category may be chosen based on a highest similarity metric. The similarity metric may computed be based on the number of terms in the data set that match a category name. The category may be chosen based on the highest number of terms matching based on the similarity metric. Techniques and operations performed for similarity analysis and categorization are further described below.

[0084] In some embodiments, knowledge service 310 can augment an input data set and use information from a knowledge source 340 to add additional, related data to the input data set. For example, a data analysis tool such as Word2Vec can be used to identify semantically similar words to those included in the input data set from a knowledge source, such as a text corpus from a news aggregation service. In some embodiments, knowledge service 310 can implement trigram modeling to preprocess data obtained from a knowledge source 340 (such as YAGO) to generate an indexed table of words by category. Knowledge service 310 can then create trigrams for each word in the augmented data set and match the word to a word from the indexed knowledge source 340.

- [0085] Using the augmented data set (or the trigram matched augmented data set), knowledge service 310 can compare the words from the augmented data set to categories of data from knowledge source 340. For example, each category of data in the knowledge source 340 can be represented as a tree structure, with the root node representing the category, and each leaf node representing a different word belonging to that category. Knowledge service 310 can implement a method (e.g., Jaccard index, or other similarity metric) to determine the semantic similarity between the augmented data set and each category in knowledge source 510. The name of the category that matches the augmented data set (e.g., having a highest similarity metric) can then be applied as a label to the input data set.
- 10 [0086] In some embodiments, knowledge service 310 can determine the similarity of two data sets A and B, by determining the ratio of the size of the intersection of the data sets to the size of the union of the data sets. For example, a similarity metric may be computed based on the ratio of 1) the size of the intersection of an data set (e.g., an augmented data set) and a category and 2) the size of their union. The similarity metric may be computed for
- 15 comparison of a data set and a category as indicated above. As such, a “best match” may be determined based on comparing the similarity metrics. The data set used for the comparison may be enriched by being augmented with a label corresponding to the category for which the best match is determined using the similarity metric.
- [0087] As described above, other similarity metrics may be used in addition, or as an
- 20 alternative, to the Jaccard index. One of ordinary skill in the art would recognize that any similarity metric may be used with the above described techniques. Some examples of alternative similarity metrics include, but are not limited to: the Dice-Sørensen index; the Tversky index; the Tanimoto metric; and the cosine similarity metric.
- [0088] In some embodiments, knowledge service 310 may utilize a data analysis tool, such as
- 25 Word2Vec, to compute a refined metric (e.g., score) that indicates a degree of match between data from a knowledge source 340 and an input data, which may be augmented with data from a knowledge source. The score (“knowledge score”) may provide greater knowledge about the degree of similarity between an input data set and a category to which a comparison is made. The knowledge score may enable data enrichment service 302 to choose a category
- 30 name that bests represents the input data.

[0089] In the techniques described above, knowledge service 310 may count the number of matches of terms in the input data set to a candidate category (e.g., genus) name in a knowledge source 340. The result of the comparison may yield a value that represents a whole integer. As such the value, although indicative of the degree of match between terms, may not indicate a degree of match between an input data set and different terms in a knowledge source.

[0090] Knowledge service 310 may utilize Word2Vec to determine a similarity of a comparison of each term (e.g., a term for a genus) in a knowledge source and the terms of input data (e.g., species). Using Word2Vec, knowledge service 310 can compute a similarity metric (e.g., cosine similarity or distance) between an input data set and one or more terms obtained from a knowledge source. The cosine similarity may be computed as the cosine angle between a data set of terms (e.g., a domain or genus) obtained from a knowledge source and an input data set of terms. The cosine similarity metric may be computed in a manner similar to the Tanimoto metric. By computing a similarity metric based on a cosine similarity, each term in the input data set may be considered as a fraction of a whole-value integer, such as a value indicating a percentage of similarity between the term and candidate category. For example, computing a similarity metric between a tire manufacturer and a surname might result in a similarity metric of 0.3, while the similarity metric between a tire manufacturer and a company name might results in a similarity metric of be 0.5. Non-whole integer values representing similarity metrics can be close compared to provide greater accuracy for a closely matching category name. The closely matching category name may be chosen as the most applicable category name based on the similarity metric closest to a value of 1. In the example, above, based on the similarity metric, company name is more likely the correct category. As such, knowledge service 310 can associated “company” instead of “surname” with a user-supplied column of data containing tire manufactures.

[0091] Knowledge service 310 can determine information about knowledge groups (e.g., domains or categories). Information about knowledge groups can be presented in a graphical user interface. Information about knowledge domains may include a metric (e.g., a knowledge score) indicating a measure of similarity between a knowledge domain and an input data set of terms. Input data may be compared to data from a knowledge source 340. An input data set may correspond to a column of data of a data set specified by a user. The knowledge score may indicate a measure of similarity between an input data set and one or

more terms provided by a knowledge source, each term corresponding to a knowledge domain. The column of data may include terms that potentially belong to knowledge domain.

[0092] In at least one embodiment, knowledge service 310 may determine a more accurate matching score. The score may correspond to a value computing using a scoring formula  
5 using techniques disclosed herein including references that are incorporated herein. The scoring formula may determine a semantic similarity between two data sets, e.g., the input data set and terms in a domain (e.g., a candidate category) obtained from a knowledge source. The domain for which the matching score indicates the best match (e.g., the highest matching score), may be chosen as the domain having the greatest similarity with the input data set. As  
10 such, the terms in the input data set may be associated with the domain name as the category.

[0093] The scoring formula may be applied to an input data set and a domain (e.g., a category of terms obtained from a knowledge source) to determine a score that indicates a measure of a match between the input data and the domain. The domain may have one or more terms, which collectively define the domain. The score may be used to determine the domain to  
15 which an input data set is most similar. The input data set may be associated with a term descriptive of the domain to which the input data set is most similar.

[0094] In some embodiments, user interface 306 can generate one or more graphical visualizations based on metadata provided by profile engine 326. As explained above, the data provided by profile engine 326 may include statistical information indicating metrics  
20 about data that has been processed by profile engine 326. Examples of graphical visualizations of metrics of profiled data are shown in FIGS. 5A-5D. A graphical visualization can include a graphical dashboard (e.g., a visualization dashboard). The graphical dashboard may indicate a plurality of metrics, each of the plurality of metrics indicating a real time metric of the data relative to a time that the data is profiled. A  
25 graphical visualization may be displayed in a user interface. For example, the graphical visualization that is generated may be sent to a client device to cause the client device to display the graphical visualization in a user interface at the client device. In some embodiments, a graphical visualization may provide profiling results.

[0095] Additionally, the structural analyses by the profile engine 326 enable the  
30 recommendation engine to better focus its queries to knowledge service 310, improving processing speed and reducing load on system resources. For example, this information can

be used to limit the scope of knowledge being queried so that the knowledge service 310 does not attempt to match a column of numerical data to place names.

[0096] FIGS. 4A-4D depict examples of a user interface that provides interactive data enrichment, in accordance with an embodiment of the present invention. As shown in FIG. 4A, an example interactive user interface 400 can display transform scripts 402, recommended transforms 404, and at least a portion of the data 406 being analyzed/transformed. Transform scripts listed in panel 402 can include indicate transforms that have been applied to the data and are visible in panel 406. Each transform script 402 can be written in a simple declarative language intelligible to a business user. Transform scripts listed in panel 402 may be automatically applied to the data and reflected in the portion of the data 406 displayed in the interactive user interface 400. For example, the transform scripts listed in patent 402 include renaming columns to be descriptive of their content. Columns 408 shown in interactive user interface 400 have been renamed according to the transform scripts 402 (e.g., column 0003 is now named date\_\_time\_\_02, column 0007 is no named "url", etc.). Recommended transforms 404, however, have not been automatically applied to the user's data.

[0097] As shown in FIG. 4B, a user can view recommendations in recommendation panel 404 and based on the recommendation, identify the data to be changed. For example, recommendation 410 includes a recommendation to rename "Col\_0008 to city". Because the recommendation is written such that a business user can understand it (instead of in, e.g., code or pseudo code) the corresponding data 412 can be readily identified by the user. As shown in FIG. 4B, data 412 includes a column of strings (represented as a row in user interface 400). The profile engine 326 can analyze the data to determine that it includes strings of two or fewer words (or tokens). This pattern can be provided to recommendation engine 318 which can query knowledge service 310. In this case, knowledge service 310 has matched the data pattern to city names and recommendation 408 was generated to rename the column accordingly.

[0098] In some embodiments, transforms listed in panel 404 may have been applied at the direction of the user (e.g., in response to an instruction to apply the transform) or may have been applied automatically. For example, in some embodiments, knowledge service 310 can provide a confidence score for a given pattern match. A threshold can be set in

recommendation engine 318 such that matches having a confidence score greater than the threshold are applied automatically.

5 [0099] To accept the recommendation, the user can select an accept icon 414 (in this example an up arrow icon) associated with the recommendation. As shown in FIG. 4C, this moves the accepted recommendation 414 to transform scripts panel 402 and automatically applies the transform to the corresponding data 416. For example, in the embodiment shown in FIG. 4C, Col\_0008 has now been renamed to “city” in accordance with the selected transform.

10 [0100] In some embodiments, data enrichment service 302 can recommend additional columns of data to be added to a data source. As shown in FIG. 4D, continuing with the city example, transforms 418 have been accepted to enrich the data with new columns including city population, and city location detail including longitude and latitude. When selected, the user’s data set is enriched to include this additional information 420. The data set now includes information that was not previously available to the user in a comprehensive and automated fashion. The user’s data set can now be used to produce a nationwide map of  
15 locations and population zones associated with other data in the dataset (for example, this may be associated with a company’s web site transactions).

[0101] FIGS. 5A-5D depict examples of various user interfaces that provide visualizations of datasets, in accordance with an embodiment of the present invention.

20 [0102] FIG. 5A depicts an example of a user interface that provides visualizations of datasets, in accordance with an embodiment of the present invention. As shown in FIG. 5A, an example interactive user interface 500 can display a profile summary 502 (“Profile Results”), transform scripts 504, recommended transforms 506, and at least a portion of the data 508 being analyzed/transformed. Transforms listed in panel 504 can include transforms that have been applied to the data and are visible in panel 508.

25 [0103] Profile summary 502 can include global statistics (e.g., total rows and columns) as well as column-specific statistics. The column-specific statistics can be generated from analysis of data processed by data enrichment service 302. In some embodiments, the column-specific statistics can be generated based on column information determined by analysis of data process by data enrichment service 302.

30 [0104] Profile summary 502 may include a map (e.g., “a heat map”) of the United States, where different areas of the United States are shown in different colors, based on statistics



identified from the data being analyzed 508. The statistics may indicate how frequently those locations are identified as being associated with the data. In one illustrative example, data may represent purchase transactions at an online retailer, where each transaction can be associated with a location (e.g., based on shipping/billing addresses, or based on recorded IP  
5 addresses). Profile summary 502 may indicate locations of transactions based on processing of the data representing the purchase transactions. In some embodiments, visualizations can be modified based on user input to assist the user in searching the data and finding useful correlations. These features are described further below.

[0105] FIGs. 5B, 5C, and 5D show examples of results of interactive data enrichment for data  
10 sets. FIG. 5B shows a user interface 540 that can include a profile metric panel 542. Panel 542 can show a summary of metrics associated with the selected data source. In some embodiments, as shown in FIG. 5C, a profile metric panel 560 can include metrics for a particular column 562, instead of an entire data set. For example, the user can select the particular column on the user's client device and the corresponding column profile 564 can  
15 be displayed. In this example, the profiler indicates a 92% match of column\_0008 with known cities in the knowledge source. A high probability in some embodiments can cause the transform engine to automatically label col\_0008 to "city".

[0106] FIG. 5D shows a profile metric panel 580 that includes global metrics 582 (e.g., metrics related to an entire dataset), and column-specific visualizations 584. The column  
20 specific visualizations 584 can be selected by a user and/or used to navigate the data (e.g., by clicking, dragging, swiping, etc.). The examples described above represent simplified transforms to small data sets. Similar and more complex processing can also be applied automatically to large data sets comprising billions of records.

[0107] With reference to FIG. 6, individual embodiments may be described as a process  
25 which is depicted as a flowchart, a flow diagram, a data flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations may be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged. A process is terminated when its operations are completed, but could have additional steps not included in a figure. A process may  
30 correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination may correspond to a return of the function to the calling function or the main function.

[0108] The processes depicted FIG. 6 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors cores), hardware, or combinations thereof. The software may be stored in a memory (e.g., on a memory device, on a non-transitory computer-readable storage medium). The particular series of processing steps in FIG. 6 is not intended to be limiting. Other sequences of steps may also be performed according to alternative embodiments. For example, alternative embodiments of the present invention may perform the steps outlined above in a different order. Moreover, the individual steps illustrated in FIG. 6 may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step.

Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0109] In an aspect of some embodiments, each process in the flowchart of FIG. 6 can be performed by one or more processing units. A processing unit may include one or more processors, including single core or multicore processors, one or more cores of processors, or combinations thereof. In some embodiments, a processing unit can include one or more special purpose co-processors such as graphics processors, digital signal processors (DSPs), or the like. In some embodiments, some or all of processing units can be implemented using customized circuits, such as application specific integrated circuits (ASICs), or field programmable gate arrays (FPGAs).

[0110] FIG. 6 depicts a flowchart of a method of data enrichment, in accordance with an embodiment of the present invention. In some embodiments, process depicted in FIG. 6 can be implemented by a computing system of a data enrichment service 302. At block 602, a processing unit of the computing system may identify patterns in data from one or more data sources. At block 604, a processing unit of the computing system can match the patterns to entity information obtained from a knowledge service. Entity information may indicate information about an entity. For example, an entity can be an address, a business name, a location, a person name, an identification number, etc. As described above, a profile engine can identify patterns in the data using the knowledge service and one or more knowledge sources. Entity information may be obtained from a knowledge service in one or a combination of many types of data structures including, without limitation, an array, a record, a relational database table, a hash table, a linked list, or other types of data structures.

[0111] At block 606, a processing unit of the computing system can generate one or more transformation scripts for the data based on the entity information. For example, transformation scripts may be determined for transforming the data corresponding to those patterns matching entity information. Transforming the data may include updating, adding, or deleting portions of the data. The transformation scripts can include scripts to delete data, reformat (e.g., modify) data, and/or enrich (e.g., add) data. The entities corresponding to the entity information may be used to determine what data to transform and how to transform the data. Entity information may be identified in the data based on one or more attributes of the data. At block 608, a processing unit of the computing system may generate one or more recommendations corresponding to the one or more transformation scripts. At block 610, a processing unit of the computing system may display the one or more recommendations in a user interface.

[0112] At block 612, a processing unit of the computing system may receive transformation instructions based on the one or more recommendations. Transformation instructions may indicate how the data received from one or more data sources may be modified and/or enriched based on one or more recommendations that are chosen. In some embodiments, the transformation instructions include a transformation instruction to rename at least one column of data based on the entity information. In some embodiments, a further transformation instruction can be received to rename the at least one column of data to a default name. A default name may include a name that is pre-determined. A default name may be any name that is pre-defined when a name for a column of data cannot be determined or is not defined. The transformation instructions can include a transformation instruction to reformat at least one column of data based on the entity information, and a transformation instruction to obfuscate at least one column of data based on the entity information. In some embodiments, the transformation instructions can include an enrichment instruction to add one or more columns of data obtained from the knowledge service based on the entity information.

[0113] At block 614, a processing unit of the computing system may transform the data based on transformation instructions. As explained above, the transformation instructions indicate how data is enriched and/or modified. Data may be transformed by performing operations, such as those disclosed herein, for enriching and/or modifying the data based on chosen recommendations. For example, for a transformation instruction to reformat data, the data may be transformed by modification to reformat the data. As such, the transformed data may

be reformatted. At block 616, a processing unit of the computing system may publish the transformed data to one or more data targets based on the transformation instructions. The flowchart may end at 618.

[0114] FIG. 7 depicts a simplified diagram of a distributed system 700 for implementing an embodiment. In the illustrated embodiment, distributed system 700 includes one or more client computing devices 702, 704, 706, and 708, which are configured to execute and operate a client application such as a web browser, proprietary client (e.g., Oracle Forms), or the like over one or more network(s) 710. Server 712 may be communicatively coupled with remote client computing devices 702, 704, 706, and 708 via network 710.

[0115] In various embodiments, server 712 may be adapted to run one or more services or software applications such as services and applications that provide the document (e.g., webpage) analysis and modification-related processing. In certain embodiments, server 712 may also provide other services or software applications can include non-virtual and virtual environments. In some embodiments, these services may be offered as web-based or cloud services or under a Software as a Service (SaaS) model to the users of client computing devices 702, 704, 706, and/or 708. Users operating client computing devices 702, 704, 706, and/or 708 may in turn utilize one or more client applications to interact with server 712 to utilize the services provided by these components.

[0116] In the configuration depicted in FIG. 7, software components 718, 720 and 722 of system 700 are shown as being implemented on server 712. In other embodiments, one or more of the components of system 700 and/or the services provided by these components may also be implemented by one or more of the client computing devices 702, 704, 706, and/or 708. Users operating the client computing devices may then utilize one or more client applications to use the services provided by these components. These components may be implemented in hardware, firmware, software, or combinations thereof. It should be appreciated that various different system configurations are possible, which may be different from distributed system 700. The embodiment shown in FIG. 7 is thus one example of a distributed system for implementing an embodiment system and is not intended to be limiting.

[0117] Client computing devices 702, 704, 706, and/or 708 may include various types of computing systems. For example, client device may include portable handheld devices (e.g.,

an iPhone®, cellular telephone, an iPad®, computing tablet, a personal digital assistant (PDA)) or wearable devices (e.g., a Google Glass® head mounted display), running software such as Microsoft Windows Mobile®, and/or a variety of mobile operating systems such as iOS, Windows Phone, Android, BlackBerry 8, Palm OS, and the like. The devices may support various applications such as various Internet-related apps, e-mail, short message service (SMS) applications, and may use various other communication protocols. The client computing devices may also include general purpose personal computers including, by way of example, personal computers and/or laptop computers running various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems. The client computing devices can be workstation computers running any of a variety of commercially-available UNIX® or UNIX-like operating systems, including without limitation the variety of GNU/Linux operating systems, such as for example, Google Chrome OS. Client computing devices may also include electronic devices such as a thin-client computer, an Internet-enabled gaming system (e.g., a Microsoft Xbox gaming console with or without a Kinect® gesture input device), and/or a personal messaging device, capable of communicating over network(s) 710.

[0118] Although distributed system 700 in FIG. 7 is shown with four client computing devices, any number of client computing devices may be supported. Other devices, such as devices with sensors, etc., may interact with server 712.

[0119] Network(s) 710 in distributed system 700 may be any type of network familiar to those skilled in the art that can support data communications using any of a variety of available protocols, including without limitation TCP/IP (transmission control protocol/Internet protocol), SNA (systems network architecture), IPX (Internet packet exchange), AppleTalk, and the like. Merely by way of example, network(s) 710 can be a local area network (LAN), networks based on Ethernet, Token-Ring, a wide-area network, the Internet, a virtual network, a virtual private network (VPN), an intranet, an extranet, a public switched telephone network (PSTN), an infra-red network, a wireless network (e.g., a network operating under any of the Institute of Electrical and Electronics (IEEE) 802.11 suite of protocols, Bluetooth®, and/or any other wireless protocol), and/or any combination of these and/or other networks.

[0120] Server 712 may be composed of one or more general purpose computers, specialized server computers (including, by way of example, PC (personal computer) servers, UNIX®

servers, mid-range servers, mainframe computers, rack-mounted servers, etc.), server farms, server clusters, or any other appropriate arrangement and/or combination. Server 712 can include one or more virtual machines running virtual operating systems, or other computing architectures involving virtualization. One or more flexible pools of logical storage devices  
5 can be virtualized to maintain virtual storage devices for the server. Virtual networks can be controlled by server 712 using software defined networking. In various embodiments, server 712 may be adapted to run one or more services or software applications described in the foregoing disclosure. For example, server 712 may correspond to a server for performing processing as described above according to an embodiment of the present disclosure.

10 [0121] Server 712 may run an operating system including any of those discussed above, as well as any commercially available server operating system. Server 712 may also run any of a variety of additional server applications and/or mid-tier applications, including HTTP (hypertext transport protocol) servers, FTP (file transfer protocol) servers, CGI (common gateway interface) servers, JAVA® servers, database servers, and the like. Exemplary  
15 database servers include without limitation those commercially available from Oracle, Microsoft, Sybase, IBM (International Business Machines), and the like.

[0122] In some implementations, server 712 may include one or more applications to analyze and consolidate data feeds and/or event updates received from users of client computing devices 702, 704, 706, and 708. As an example, data feeds and/or event updates may include,  
20 but are not limited to, Twitter® feeds, Facebook® updates or real-time updates received from one or more third party information sources and continuous data streams, which may include real-time events related to sensor data applications, financial tickers, network performance measuring tools (e.g., network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like. Server 712 may also include one  
25 or more applications to display the data feeds and/or real-time events via one or more display devices of client computing devices 702, 704, 706, and 708.

[0123] Distributed system 700 may also include one or more databases 714 and 716. These databases may provide a mechanism for storing information such as user interactions information, usage patterns information, adaptation rules information, and other information  
30 used by embodiments of the present invention. Databases 714 and 716 may reside in a variety of locations. By way of example, one or more of databases 714 and 716 may reside on a non-transitory storage medium local to (and/or resident in) server 712. Alternatively,

databases 714 and 716 may be remote from server 712 and in communication with server 712 via a network-based or dedicated connection. In one set of embodiments, databases 714 and 716 may reside in a storage-area network (SAN). Similarly, any necessary files for performing the functions attributed to server 712 may be stored locally on server 712 and/or  
5 remotely, as appropriate. In one set of embodiments, databases 714 and 716 may include relational databases, such as databases provided by Oracle, that are adapted to store, update, and retrieve data in response to SQL-formatted commands.

[0124] In some embodiments, the document analysis and modification services described above may be offered as services via a cloud environment. FIG. 8 is a simplified block  
10 diagram of one or more components of a system environment 800 in which services may be offered as cloud services, in accordance with an embodiment of the present disclosure. In the illustrated embodiment in FIG. 8, system environment 800 includes one or more client computing devices 804, 806, and 808 that may be used by users to interact with a cloud infrastructure system 802 that provides cloud services, including services for dynamically  
15 modifying documents (e.g., webpages) responsive to usage patterns. Cloud infrastructure system 802 may comprise one or more computers and/or servers that may include those described above for server 812.

[0125] It should be appreciated that cloud infrastructure system 802 depicted in FIG. 8 may have other components than those depicted. Further, the embodiment shown in FIG. 8 is only  
20 one example of a cloud infrastructure system that may incorporate an embodiment of the invention. In some other embodiments, cloud infrastructure system 802 may have more or fewer components than shown in the figure, may combine two or more components, or may have a different configuration or arrangement of components.

[0126] Client computing devices 804, 806, and 808 may be devices similar to those described  
25 above for 702, 704, 706, and 708. Client computing devices 804, 806, and 808 may be configured to operate a client application such as a web browser, a proprietary client application (e.g., Oracle Forms), or some other application, which may be used by a user of the client computing device to interact with cloud infrastructure system 802 to use services provided by cloud infrastructure system 802. Although exemplary system environment 800  
30 is shown with three client computing devices, any number of client computing devices may be supported. Other devices such as devices with sensors, etc. may interact with cloud infrastructure system 802.

[0127] Network(s) 810 may facilitate communications and exchange of data between clients 804, 806, and 808 and cloud infrastructure system 802. Each network may be any type of network familiar to those skilled in the art that can support data communications using any of a variety of commercially-available protocols, including those described above for network(s) 810.

[0128] In certain embodiments, services provided by cloud infrastructure system 802 may include a host of services that are made available to users of the cloud infrastructure system on demand. In addition to services related to dynamic document modification responsive usage patterns, various other services may also be offered including without limitation online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services, and the like. Services provided by the cloud infrastructure system can dynamically scale to meet the needs of its users.

[0129] In certain embodiments, a specific instantiation of a service provided by cloud infrastructure system 802 may be referred to herein as a “service instance.” In general, any service made available to a user via a communication network, such as the Internet, from a cloud service provider’s system is referred to as a “cloud service.” Typically, in a public cloud environment, servers and systems that make up the cloud service provider’s system are different from the customer’s own on-premises servers and systems. For example, a cloud service provider’s system may host an application, and a user may, via a communication network such as the Internet, on demand, order and use the application.

[0130] In some examples, a service in a computer network cloud infrastructure may include protected computer network access to storage, a hosted database, a hosted web server, a software application, or other service provided by a cloud vendor to a user, or as otherwise known in the art. For example, a service can include password-protected access to remote storage on the cloud through the Internet. As another example, a service can include a web service-based hosted relational database and a script-language middleware engine for private use by a networked developer. As another example, a service can include access to an email software application hosted on a cloud vendor’s web site.

[0131] In certain embodiments, cloud infrastructure system 802 may include a suite of applications, middleware, and database service offerings that are delivered to a customer in a



self-service, subscription-based, elastically scalable, reliable, highly available, and secure manner. An example of such a cloud infrastructure system is the Oracle Public Cloud provided by the present assignee.

[0132] Cloud infrastructure system 802 may also provide “big data” elated computation and analysis services. The term “big data” is generally used to refer to extremely large data sets that can be stored and manipulated by analysts and researchers to visualize large amounts of data, detect trends, and/or otherwise interact with the data. This big data and related applications can be hosted and/or manipulated by an infrastructure system on many levels and at different scales. Tens, hundreds, or thousands of processors linked in parallel can act upon such data in order to present it or simulate external forces on the data or what it represents. These data sets can involve structured data, such as that organized in a database or otherwise according to a structured model, and/or unstructured data (e.g., emails, images, data blobs (binary large objects), web pages, complex event processing). By leveraging an ability of an embodiment to relatively quickly focus more (or fewer) computing resources upon an objective, the cloud infrastructure system may be better available to carry out tasks on large data sets based on demand from a business, government agency, research organization, private individual, group of like-minded individuals or organizations, or other entity.

[0133] In various embodiments, cloud infrastructure system 802 may be adapted to automatically provision, manage and track a customer’s subscription to services offered by cloud infrastructure system 802. Cloud infrastructure system 802 may provide the cloud services via different deployment models. For example, services may be provided under a public cloud model in which cloud infrastructure system 802 is owned by an organization selling cloud services (e.g., owned by Oracle Corporation) and the services are made available to the general public or different industry enterprises. As another example, services may be provided under a private cloud model in which cloud infrastructure system 802 is operated solely for a single organization and may provide services for one or more entities within the organization. The cloud services may also be provided under a community cloud model in which cloud infrastructure system 802 and the services provided by cloud infrastructure system 802 are shared by several organizations in a related community. The cloud services may also be provided under a hybrid cloud model, which is a combination of two or more different models.

[0134] In some embodiments, the services provided by cloud infrastructure system 802 may include one or more services provided under Software as a Service (SaaS) category, Platform as a Service (PaaS) category, Infrastructure as a Service (IaaS) category, or other categories of services including hybrid services. A customer, via a subscription order, may order one or  
5 more services provided by cloud infrastructure system 802. Cloud infrastructure system 802 then performs processing to provide the services in the customer's subscription order.

[0135] In some embodiments, the services provided by cloud infrastructure system 802 may include, without limitation, application services, platform services and infrastructure services. In some examples, application services may be provided by the cloud infrastructure system  
10 via a SaaS platform. The SaaS platform may be configured to provide cloud services that fall under the SaaS category. For example, the SaaS platform may provide capabilities to build and deliver a suite of on-demand applications on an integrated development and deployment platform. The SaaS platform may manage and control the underlying software and infrastructure for providing the SaaS services. By utilizing the services provided by the SaaS  
15 platform, customers can utilize applications executing on the cloud infrastructure system. Customers can acquire the application services without the need for customers to purchase separate licenses and support. Various different SaaS services may be provided. Examples include, without limitation, services that provide solutions for sales performance management, enterprise integration, and business flexibility for large organizations.

[0136] In some embodiments, platform services may be provided by cloud infrastructure system 802 via a PaaS platform. The PaaS platform may be configured to provide cloud services that fall under the PaaS category. Examples of platform services may include without limitation services that enable organizations (such as Oracle) to consolidate existing applications on a shared, common architecture, as well as the ability to build new applications  
20 that leverage the shared services provided by the platform. The PaaS platform may manage and control the underlying software and infrastructure for providing the PaaS services. Customers can acquire the PaaS services provided by cloud infrastructure system 802 without the need for customers to purchase separate licenses and support. Examples of platform services include, without limitation, Oracle Java Cloud Service (JCS), Oracle Database Cloud  
25 Service (DBCS), and others.  
30

[0137] By utilizing the services provided by the PaaS platform, customers can employ programming languages and tools supported by the cloud infrastructure system and also

control the deployed services. In some embodiments, platform services provided by the cloud infrastructure system may include database cloud services, middleware cloud services (e.g., Oracle Fusion Middleware services), and Java cloud services. In one embodiment, database cloud services may support shared service deployment models that enable organizations to pool database resources and offer customers a Database as a Service in the form of a database cloud. Middleware cloud services may provide a platform for customers to develop and deploy various business applications, and Java cloud services may provide a platform for customers to deploy Java applications, in the cloud infrastructure system.

[0138] Various different infrastructure services may be provided by an IaaS platform in the cloud infrastructure system. The infrastructure services facilitate the management and control of the underlying computing resources, such as storage, networks, and other fundamental computing resources for customers utilizing services provided by the SaaS platform and the PaaS platform.

[0139] In certain embodiments, cloud infrastructure system 802 may also include infrastructure resources 830 for providing the resources used to provide various services to customers of the cloud infrastructure system. In one embodiment, infrastructure resources 830 may include pre-integrated and optimized combinations of hardware, such as servers, storage, and networking resources to execute the services provided by the PaaS platform and the SaaS platform, and other resources.

[0140] In some embodiments, resources in cloud infrastructure system 802 may be shared by multiple users and dynamically re-allocated per demand. Additionally, resources may be allocated to users in different time zones. For example, cloud infrastructure system 802 may enable a first set of users in a first time zone to utilize resources of the cloud infrastructure system for a specified number of hours and then enable the re-allocation of the same resources to another set of users located in a different time zone, thereby maximizing the utilization of resources.

[0141] In certain embodiments, a number of internal shared services 832 may be provided that are shared by different components or modules of cloud infrastructure system 802 to enable provision of services by cloud infrastructure system 802. These internal shared services may include, without limitation, a security and identity service, an integration service, an enterprise repository service, an enterprise manager service, a virus scanning and

white list service, a high availability, backup and recovery service, service for enabling cloud support, an email service, a notification service, a file transfer service, and the like.

[0142] In certain embodiments, cloud infrastructure system 802 may provide comprehensive management of cloud services (e.g., SaaS, PaaS, and IaaS services) in the cloud infrastructure system. In one embodiment, cloud management functionality may include capabilities for provisioning, managing and tracking a customer's subscription received by cloud infrastructure system 802, and the like.

[0143] In one embodiment, as depicted in FIG. 8, cloud management functionality may be provided by one or more modules, such as an order management module 820, an order orchestration module 822, an order provisioning module 824, an order management and monitoring module 826, and an identity management module 828. These modules may include or be provided using one or more computers and/or servers, which may be general purpose computers, specialized server computers, server farms, server clusters, or any other appropriate arrangement and/or combination.

[0144] In an exemplary operation, at 834, a customer using a client device, such as client device 804, 806 or 808, may interact with cloud infrastructure system 802 by requesting one or more services provided by cloud infrastructure system 802 and placing an order for a subscription for one or more services offered by cloud infrastructure system 802. In certain embodiments, the customer may access a cloud User Interface (UI) such as cloud UI 812, cloud UI 814 and/or cloud UI 816 and place a subscription order via these UIs. The order information received by cloud infrastructure system 802 in response to the customer placing an order may include information identifying the customer and one or more services offered by the cloud infrastructure system 802 that the customer intends to subscribe to.

[0145] At 836, the order information received from the customer may be stored in an order database 818. If this is a new order, a new record may be created for the order. In one embodiment, order database 818 can be one of several databases operated by cloud infrastructure system 818 and operated in conjunction with other system elements.

[0146] At 838, the order information may be forwarded to an order management module 820 that may be configured to perform billing and accounting functions related to the order, such as verifying the order, and upon verification, booking the order.

[0147] At 840, information regarding the order may be communicated to an order orchestration module 822 that is configured to orchestrate the provisioning of services and resources for the order placed by the customer. In some instances, order orchestration module 822 may use the services of order provisioning module 824 for the provisioning. In  
5 certain embodiments, order orchestration module 822 enables the management of business processes associated with each order and applies business logic to determine whether an order should proceed to provisioning.

[0148] As shown in the embodiment depicted in FIG. 8, at 842, upon receiving an order for a new subscription, order orchestration module 822 sends a request to order provisioning  
10 module 824 to allocate resources and configure resources needed to fulfill the subscription order. Order provisioning module 824 enables the allocation of resources for the services ordered by the customer. Order provisioning module 824 provides a level of abstraction between the cloud services provided by cloud infrastructure system 800 and the physical implementation layer that is used to provision the resources for providing the requested  
15 services. This enables order orchestration module 822 to be isolated from implementation details, such as whether or not services and resources are actually provisioned on the fly or pre-provisioned and only allocated/assigned upon request.

[0149] At 844, once the services and resources are provisioned, a notification may be sent to the subscribing customers indicating that the requested service is now ready for use. In some  
20 instance, information (e.g., a link) may be sent to the customer that enables the customer to start using the requested services.

[0150] At 846, a customer's subscription order may be managed and tracked by an order management and monitoring module 826. In some instances, order management and monitoring module 826 may be configured to collect usage statistics regarding a customer  
25 use of subscribed services. For example, statistics may be collected for the amount of storage used, the amount data transferred, the number of users, and the amount of system up time and system down time, and the like.

[0151] In certain embodiments, cloud infrastructure system 800 may include an identity management module 828 that is configured to provide identity services, such as access  
30 management and authorization services in cloud infrastructure system 800. In some embodiments, identity management module 828 may control information about customers

who wish to utilize the services provided by cloud infrastructure system 802. Such information can include information that authenticates the identities of such customers and information that describes which actions those customers are authorized to perform relative to various system resources (e.g., files, directories, applications, communication ports, memory segments, etc.) Identity management module 828 may also include the management of descriptive information about each customer and about how and by whom that descriptive information can be accessed and modified.

[0152] FIG. 9 illustrates an exemplary computer system 900 that may be used to implement an embodiment of the present invention. In some embodiments, computer system 900 may be used to implement any of the various servers and computer systems described above. As shown in FIG. 9, computer system 900 includes various subsystems including a processing unit 904 that communicates with a number of peripheral subsystems via a bus subsystem 902. These peripheral subsystems may include a processing acceleration unit 906, an I/O subsystem 908, a storage subsystem 918 and a communications subsystem 924. Storage subsystem 918 may include tangible computer-readable storage media 922 and a system memory 910.

[0153] Bus subsystem 902 provides a mechanism for letting the various components and subsystems of computer system 900 communicate with each other as intended. Although bus subsystem 902 is shown schematically as a single bus, alternative embodiments of the bus subsystem may utilize multiple buses. Bus subsystem 902 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. For example, such architectures may include an Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus, which can be implemented as a Mezzanine bus manufactured to the IEEE P1386.1 standard, and the like.

[0154] Processing subsystem 904 controls the operation of computer system 900 and may comprise one or more processing units 932, 934, etc. A processing unit may include be one or more processors, including single core or multicore processors, one or more cores of processors, or combinations thereof. In some embodiments, processing subsystem 904 can include one or more special purpose co-processors such as graphics processors, digital signal processors (DSPs), or the like. In some embodiments, some or all of the processing units of

processing subsystem 904 can be implemented using customized circuits, such as application specific integrated circuits (ASICs), or field programmable gate arrays (FPGAs).

[0155] In some embodiments, the processing units in processing subsystem 904 can execute instructions stored in system memory 910 or on computer readable storage media 922. In various embodiments, the processing units can execute a variety of programs or code instructions and can maintain multiple concurrently executing programs or processes. At any given time, some or all of the program code to be executed can be resident in system memory 910 and/or on computer-readable storage media 922 including potentially on one or more storage devices. Through suitable programming, processing subsystem 904 can provide various functionalities described above for dynamically modifying documents (e.g., webpages) responsive to usage patterns.

[0156] In certain embodiments, a processing acceleration unit 906 may be provided for performing customized processing or for off-loading some of the processing performed by processing subsystem 904 so as to accelerate the overall processing performed by computer system 900.

[0157] I/O subsystem 908 may include devices and mechanisms for inputting information to computer system 900 and/or for outputting information from or via computer system 900. In general, use of the term “input device” is intended to include all possible types of devices and mechanisms for inputting information to computer system 900. User interface input devices may include, for example, a keyboard, pointing devices such as a mouse or trackball, a touchpad or touch screen incorporated into a display, a scroll wheel, a click wheel, a dial, a button, a switch, a keypad, audio input devices with voice command recognition systems, microphones, and other types of input devices. User interface input devices may also include motion sensing and/or gesture recognition devices such as the Microsoft Kinect® motion sensor that enables users to control and interact with an input device, the Microsoft Xbox® 360 game controller, devices that provide an interface for receiving input using gestures and spoken commands. User interface input devices may also include eye gesture recognition devices such as the Google Glass® blink detector that detects eye activity (e.g., “blinking” while taking pictures and/or making a menu selection) from users and transforms the eye gestures as input into an input device (e.g., Google Glass®). Additionally, user interface input devices may include voice recognition sensing devices that enable users to interact with voice recognition systems (e.g., Siri® navigator), through voice commands.

- [0158] Other examples of user interface input devices include, without limitation, three dimensional (3D) mice, joysticks or pointing sticks, gamepads and graphic tablets, and audio/visual devices such as speakers, digital cameras, digital camcorders, portable media players, webcams, image scanners, fingerprint scanners, barcode reader 3D scanners, 3D printers, laser rangefinders, and eye gaze tracking devices. Additionally, user interface input devices may include, for example, medical imaging input devices such as computed tomography, magnetic resonance imaging, position emission tomography, medical ultrasonography devices. User interface input devices may also include, for example, audio input devices such as MIDI keyboards, digital musical instruments and the like.
- 10 [0159] User interface output devices may include a display subsystem, indicator lights, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device, such as that using a liquid crystal display (LCD) or plasma display, a projection device, a touch screen, and the like. In general, use of the term “output device” is intended to include all possible types of devices and mechanisms for
- 15 outputting information from computer system 900 to a user or other computer. For example, user interface output devices may include, without limitation, a variety of display devices that visually convey text, graphics and audio/video information such as monitors, printers, speakers, headphones, automotive navigation systems, plotters, voice output devices, and modems.
- 20 [0160] Storage subsystem 918 provides a repository or data store for storing information that is used by computer system 900. Storage subsystem 918 provides a tangible non-transitory computer-readable storage medium for storing the basic programming and data constructs that provide the functionality of some embodiments. Software (programs, code modules, instructions) that when executed by processing subsystem 904 provide the functionality
- 25 described above may be stored in storage subsystem 918. The software may be executed by one or more processing units of processing subsystem 904. Storage subsystem 918 may also provide a repository for storing data used in accordance with the present invention.
- [0161] Storage subsystem 918 may include one or more non-transitory memory devices, including volatile and non-volatile memory devices. As shown in FIG. 9, storage subsystem
- 30 918 includes a system memory 910 and a computer-readable storage media 922. System memory 910 may include a number of memories including a volatile main random access memory (RAM) for storage of instructions and data during program execution and a non-



volatile read only memory (ROM) or flash memory in which fixed instructions are stored. In some implementations, a basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within computer system 900, such as during start-up, may typically be stored in the ROM. The RAM typically contains data and/or program modules that are presently being operated and executed by processing subsystem 904. In some implementations, system memory 910 may include multiple different types of memory, such as static random access memory (SRAM) or dynamic random access memory (DRAM).

[0162] By way of example, and not limitation, as depicted in FIG. 9, system memory 910 may store application programs 912, which may include client applications, Web browsers, mid-tier applications, relational database management systems (RDBMS), etc., program data 914, and an operating system 916. By way of example, operating system 916 may include various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems, a variety of commercially-available UNIX® or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems, the Google Chrome® OS, and the like) and/or mobile operating systems such as iOS, Windows® Phone, Android® OS, BlackBerry® 8 OS, and Palm® OS operating systems.

[0163] Computer-readable storage media 922 may store programming and data constructs that provide the functionality of some embodiments. Software (programs, code modules, instructions) that when executed by processing subsystem 904 a processor provide the functionality described above may be stored in storage subsystem 918. By way of example, computer-readable storage media 922 may include non-volatile memory such as a hard disk drive, a magnetic disk drive, an optical disk drive such as a CD ROM, DVD, a Blu-Ray® disk, or other optical media. Computer-readable storage media 922 may include, but is not limited to, Zip® drives, flash memory cards, universal serial bus (USB) flash drives, secure digital (SD) cards, DVD disks, digital video tape, and the like. Computer-readable storage media 922 may also include, solid-state drives (SSD) based on non-volatile memory such as flash-memory based SSDs, enterprise flash drives, solid state ROM, and the like, SSDs based on volatile memory such as solid state RAM, dynamic RAM, static RAM, DRAM-based SSDs, magnetoresistive RAM (MRAM) SSDs, and hybrid SSDs that use a combination of DRAM and flash memory based SSDs. Computer-readable media 922 may provide storage

of computer-readable instructions, data structures, program modules, and other data for computer system 900.

[0164] In certain embodiments, storage subsystem 900 may also include a computer-readable storage media reader 920 that can further be connected to computer-readable storage media 922. Together and, optionally, in combination with system memory 910, computer-readable storage media 922 may comprehensively represent remote, local, fixed, and/or removable storage devices plus storage media for storing computer-readable information.

[0165] In certain embodiments, computer system 900 may provide support for executing one or more virtual machines. Computer system 900 may execute a program such as a hypervisor for facilitating the configuring and managing of the virtual machines. Each virtual machine may be allocated memory, compute (e.g., processors, cores), I/O, and networking resources. Each virtual machine typically runs its own operating system, which may be the same as or different from the operating systems executed by other virtual machines executed by computer system 900. Accordingly, multiple operating systems may potentially be run concurrently by computer system 900. Each virtual machine generally runs independently of the other virtual machines.

[0166] Communications subsystem 924 provides an interface to other computer systems and networks. Communications subsystem 924 serves as an interface for receiving data from and transmitting data to other systems from computer system 900. For example, communications subsystem 924 may enable computer system 900 to establish a communication channel to one or more client devices via the Internet for receiving and sending information from and to the client devices.

[0167] Communication subsystem 924 may support both wired and/or wireless communication protocols. For example, in certain embodiments, communications subsystem 924 may include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology, such as 3G, 4G or EDGE (enhanced data rates for global evolution), WiFi (IEEE 802.11 family standards, or other mobile communication technologies, or any combination thereof), global positioning system (GPS) receiver components, and/or other components. In some embodiments communications subsystem 924 can provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface.

[0168] Communication subsystem 924 can receive and transmit data in various forms. For example, in some embodiments, communications subsystem 924 may receive input communication in the form of structured and/or unstructured data feeds 926, event streams 928, event updates 930, and the like. For example, communications subsystem 924 may be configured to receive (or send) data feeds 926 in real-time from users of social media networks and/or other communication services such as Twitter® feeds, Facebook® updates, web feeds such as Rich Site Summary (RSS) feeds, and/or real-time updates from one or more third party information sources.

[0169] In certain embodiments, communications subsystem 924 may be configured to receive data in the form of continuous data streams, which may include event streams 928 of real-time events and/or event updates 930, that may be continuous or unbounded in nature with no explicit end. Examples of applications that generate continuous data may include, for example, sensor data applications, financial tickers, network performance measuring tools (e.g., network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like.

[0170] Communications subsystem 924 may also be configured to output the structured and/or unstructured data feeds 926, event streams 928, event updates 930, and the like to one or more databases that may be in communication with one or more streaming data source computers coupled to computer system 900.

[0171] Computer system 900 can be one of various types, including a handheld portable device (e.g., an iPhone® cellular phone, an iPad® computing tablet, a PDA), a wearable device (e.g., a Google Glass® head mounted display), a personal computer, a workstation, a mainframe, a kiosk, a server rack, or any other data processing system.

[0172] Due to the ever-changing nature of computers and networks, the description of computer system 900 depicted in FIG. 9 is intended only as a specific example. Many other configurations having more or fewer components than the system depicted in FIG. 9 are possible. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

[0173] In at least one embodiment of the present invention, a data enrichment system is provided. The data enrichment system includes one or more server computers, wherein the data enrichment system is communicatively coupled to a plurality of data sources (e.g., the

data sources 104 as shown in FIG. 1) and a plurality of data targets (e.g., the data targets 106 as shown in FIG. 1) over at least one communication network.

[0174] The data enrichment system further comprises a profile engine, a recommendation engine, a transform engine, and a publish engine. The profile engine, recommendation  
5 engine, transform engine, and publish engine can be, for example, the profile engine 326, recommendation engine 308, transform engine 322, and publish engine 324 as shown in FIG. 3, respectively.

[0175] The profile engine is configured to identify patterns in data from at least one data source specified in a data enrichment request, when the data enrichment request is received  
10 from a client device. The recommendation engine is configured to match the patterns to entity information from a knowledge service (e.g., the knowledge service 310 as shown in FIG. 3). The transformation engine is configured to generate one or more transformation scripts for the data based on the entity information, wherein said recommendation engine generates one or more recommendations corresponding to the one or more transformation  
15 scripts, and the one or more recommendations are caused to be displayed in a user interface on the client device. The publish engine is configured to, after transformation instructions being received from the client device based on the one or more recommendations and the data being transformed based on the transformation instructions, publish the transformed data to one or more data targets based on the transformation instructions.

[0176] In one embodiment of the invention, when a selection of at least one column of data is received from at least one of the data sources, a data profile of the at least one column of data is caused to be displayed in the user interface on the client device, wherein the data profile indicates a plurality of patterns identified in the at least one column of data and at least one data visualization associated with the at least one column of data.

[0177] In one embodiment of the invention, the transformation instructions include a transformation instruction to rename at least one column of data based on the entity information.

[0178] In one embodiment of the invention, a further transformation instruction is received to rename the at least one column of data to a default name.

[0179] In one embodiment of the invention, the transformation instructions include a transformation instruction to reformat at least one column of data based on the entity information.

5 [0180] In one embodiment of the invention, the transformation instructions include a further transformation instruction to obfuscate at least one column of data based on the entity information.

[0181] In one embodiment of the invention, the transformation instructions include an enrichment instruction to add one or more columns of data obtained from the knowledge service based on the entity information.

10 [0182] It is apparent for those skilled in the art that, for the particular operation processes of the units/modules (e.g., engines) described above, reference may be made to the corresponding steps/components in the related method/system embodiment sharing the same concept and the reference is regarded as the disclosure of the related units/modules also. And therefore some of the particular operation processes will not be described repeatedly or in  
15 detail for convenience and concision of description.

[0183] It is also apparent for those skilled in the art that the units/modules can be implemented in an electronic device in the manner of software, hardware and/or combination of software and hardware. Components described as separate components may or may not be physically separated. In particular, components according to each embodiment of the present  
20 invention may be integrated in one physical component or may exist in various separate physical components. The various implementations of units in the electronic device are all included within the scope of protection of the invention.

[0184] It should be understood that the unit, apparatus, and device may be implemented in form of software, hardware known or developed in the future and/or the combination of such  
25 software and hardware.

[0185] It is apparent for persons in the art that the operations described in Fig. 3 may be implemented in form of software, hardware and/or the combination of such software and hardware, depending on the particular application environment. It is apparent for persons in the art that at least some of the steps can be implemented by running instructions in general  
30 processor, with the instructions stored in memory. It is apparent for persons in the art that at least some of the steps can also be implemented by various hardware, including but not

limited to DSP, FPGA, ASIC etc. For example, the “operations” in some of the embodiments may be implemented by instructions running in CPU or by special processor such as DSP, FPGA, ASIC implementing the functionality of “operations”.

[0186] Although specific embodiments of the invention have been described, various  
5 modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. Embodiments of the present invention are not restricted to operation within certain specific data processing environments, but are free to operate within a plurality of data processing environments. Additionally, although embodiments of the present invention have been described using a particular series of transactions and steps, it  
10 should be apparent to those skilled in the art that the scope of the present invention is not limited to the described series of transactions and steps. Various features and aspects of the above-described embodiments may be used individually or jointly.

[0187] Further, while embodiments of the present invention have been described using a particular combination of hardware and software, it should be recognized that other  
15 combinations of hardware and software are also within the scope of the present invention. Embodiments of the present invention may be implemented only in hardware, or only in software, or using combinations thereof. The various processes described herein can be implemented on the same processor or different processors in any combination. Accordingly, where components or modules are described as being configured to perform certain  
20 operations, such configuration can be accomplished, e.g., by designing electronic circuits to perform the operation, by programming programmable electronic circuits (such as microprocessors) to perform the operation, or any combination thereof. Processes can communicate using a variety of techniques including but not limited to conventional techniques for interprocess communication, and different pairs of processes may use different  
25 techniques, or the same pair of processes may use different techniques at different times.

[0188] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without departing from the broader spirit and scope as set forth in the claims. Thus, although specific invention  
30 embodiments have been described, these are not intended to be limiting. Various modifications and equivalents are within the scope of the following claims.

WHAT IS CLAIMED IS:

- 1                   1.     A method comprising:  
2                   identifying, by a computer system, patterns in data from one or more data  
3                   sources;  
4                   matching the patterns to entity information from a knowledge service;  
5                   generating one or more transformation scripts for the data based on the entity  
6                   information;  
7                   generating one or more recommendations corresponding to the one or more  
8                   transformation scripts;  
9                   causing the one or more recommendations to be displayed in a user interface;  
10                  receiving transformation instructions based on the one or more  
11                  recommendations;  
12                  transforming the data based on the transformation instructions; and  
13                  publishing the transformed data to one or more data targets based on the  
14                  transformation instructions.
- 1                   2.     The method of claim 1 or claim 2, further comprising:  
2                   receiving a selection of at least one column of data from at least one of the one  
3                   or more data sources; and  
4                   causing a data profile of the at least one column of data to be displayed,  
5                   wherein the data profile indicates a plurality of patterns identified in the at least one column  
6                   of data and at least one data visualization associated with the at least one column of data.
- 1                   3.     The method of claim 1, wherein the transformation instructions include  
2                   a transformation instruction to rename at least one column of data based on the entity  
3                   information.
- 1                   4.     The method of claim 3, further comprising:  
2                   receiving a further transformation instruction to rename the at least one  
3                   column of data to a default name.
- 1                   5.     The method of any one of the preceding claims, wherein the  
2                   transformation instructions include a transformation instruction to reformat at least one  
3                   column of data based on the entity information.

1                   6.       The method of claim 5, wherein the transformation instructions include  
2   a further transformation instruction to obfuscate at least one column of data based on the  
3   entity information.

1                   7.       The method of claim any one of the preceding claims, wherein the  
2   transformation instructions include an enrichment instruction to add one or more columns of  
3   data obtained from the knowledge service based on the entity information.

1                   8.       A system comprising:  
2                   a plurality of data sources;  
3                   a plurality of data targets; and  
4                   a cloud computing infrastructure system comprising:  
5                         one or more processors communicatively coupled to the plurality of  
6                   data sources and communicatively coupled to the plurality of data targets, over at least  
7                   one communication network; and  
8                         a memory coupled to the one or more processors, the memory storing  
9                   instructions to provide a data enrichment service, wherein the instructions,  
10                  when executed by the one or more processors, cause the one or more  
11                  processors to:  
12                         receive a data enrichment request from a client device;  
13                         identify patterns in data from at least one data source specified in the  
14                  data enrichment request;  
15                         match the patterns to entity information from a knowledge service;  
16                         generate one or more transformation scripts for the data based on the  
17                  entity information;  
18                         generate one or more recommendations corresponding to the one or  
19                  more transformation scripts;  
20                         cause the one or more recommendations to be displayed in a user  
21                  interface on the client device;  
22                         receive transformation instructions based on the one or more  
23                  recommendations;  
24                         transform the data based on the transformation instructions; and



25                   publish the transformed data to one or more data targets based on the  
26           transformation instructions.

1                   9.       The system of claim 8, wherein the instructions, when executed by the  
2   one or more processors, further cause the one or more processors to:  
3                   receive a selection of at least one column of data from at least one of the  
4   plurality of data sources; and  
5                   cause a data profile of the at least one column of data to be displayed in the  
6   user interface on the client device, wherein the data profile indicates a plurality of patterns  
7   identified in the at least one column of data and at least one data visualization associated with  
8   the at least one column of data.

1                   10.     The system of claim 8, wherein the transformation instructions include  
2   a transformation instruction to rename at least one column of data based on the entity  
3   information.

1                   11.     The system of claim 10, wherein the instructions, when executed by  
2   the one or more processors, further cause the one or more processors to:  
3                   receive a further transformation instruction to rename the at least one column  
4   of data to a default name.

1                   12.     The system of claim 8, wherein the transformation instructions include  
2   a transformation instruction to reformat at least one column of data based on the entity  
3   information.

1                   13.     The system of claim 12, wherein the transformation instructions  
2   include a further transformation instruction to obfuscate at least one column of data based on  
3   the entity information.

1                   14.     The system of claim 8, wherein the transformation instructions include  
2   an enrichment instruction to add one or more columns of data obtained from the knowledge  
3   service based on the entity information.

1                   15.     A non-transitory computer readable storage medium including  
2   instructions stored thereon which, when executed by a processor, cause the processor to  
3   perform operations comprising:

4 identifying, by a computer system, patterns in data from one or more data  
5 sources;  
6 matching the patterns to entity information from a knowledge service;  
7 generating one or more transformation scripts for the data based on the entity  
8 information;  
9 generating one or more recommendations corresponding to the one or more  
10 transformation scripts;  
11 causing the one or more recommendations to be displayed in a user interface;  
12 receiving transformation instructions based on the one or more  
13 recommendations;  
14 transforming the data based on the transformation instructions; and  
15 publishing the transformed data to one or more data targets based on the  
16 transformation instructions.

1 16. The non-transitory computer readable storage medium of claim 15,  
2 wherein the operations further comprise:  
3 receiving a selection of at least one column of data from at least one of the one  
4 or more data sources; and  
5 causing a data profile of the at least one column of data to be displayed,  
6 wherein the data profile indicates a plurality of patterns identified in the at least one column  
7 of data and at least one data visualization associated with the at least one column of data.

1 17. The non-transitory computer readable storage medium of claim 15,  
2 wherein the transformation instructions include a transformation instruction to rename at least  
3 one column of data based on the entity information.

1 18. The non-transitory computer readable storage medium of claim 17,  
2 wherein the operations further comprise:  
3 receiving a further transformation instruction to rename the at least one  
4 column of data to a default name.

1 19. The non-transitory computer readable storage medium of claim 15,  
2 wherein the transformation instructions include:

3                   a transformation instruction to reformat at least one column of data based on  
4   the entity information; and  
5                   a further transformation instruction to obfuscate at least one column of data  
6   based on the entity information.

1                   20.     The non-transitory computer readable storage medium of claim 15,  
2   wherein the transformation instructions include an enrichment instruction to add one or more  
3   columns of data obtained from the knowledge service based on the entity information.

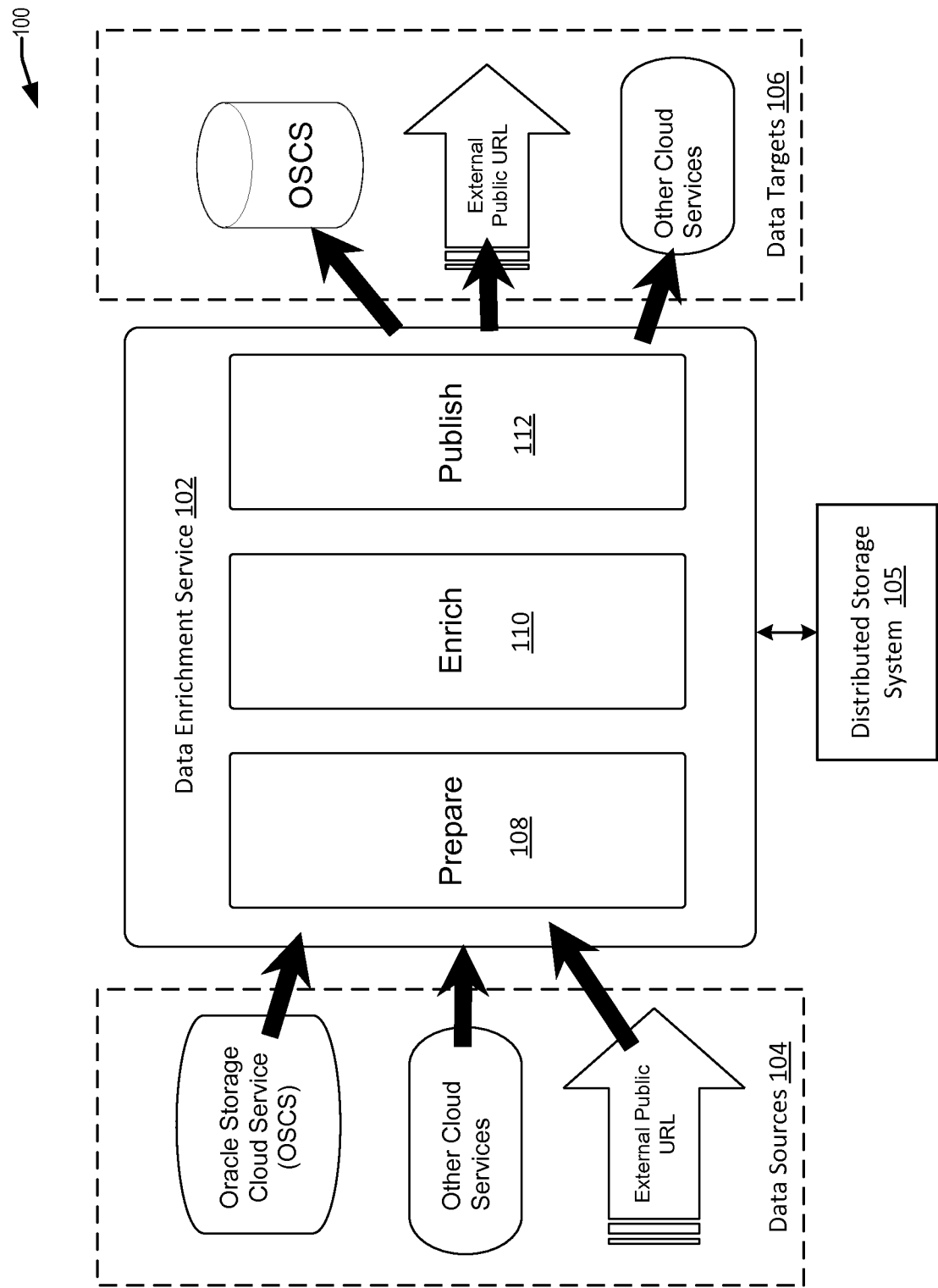


FIG. 1

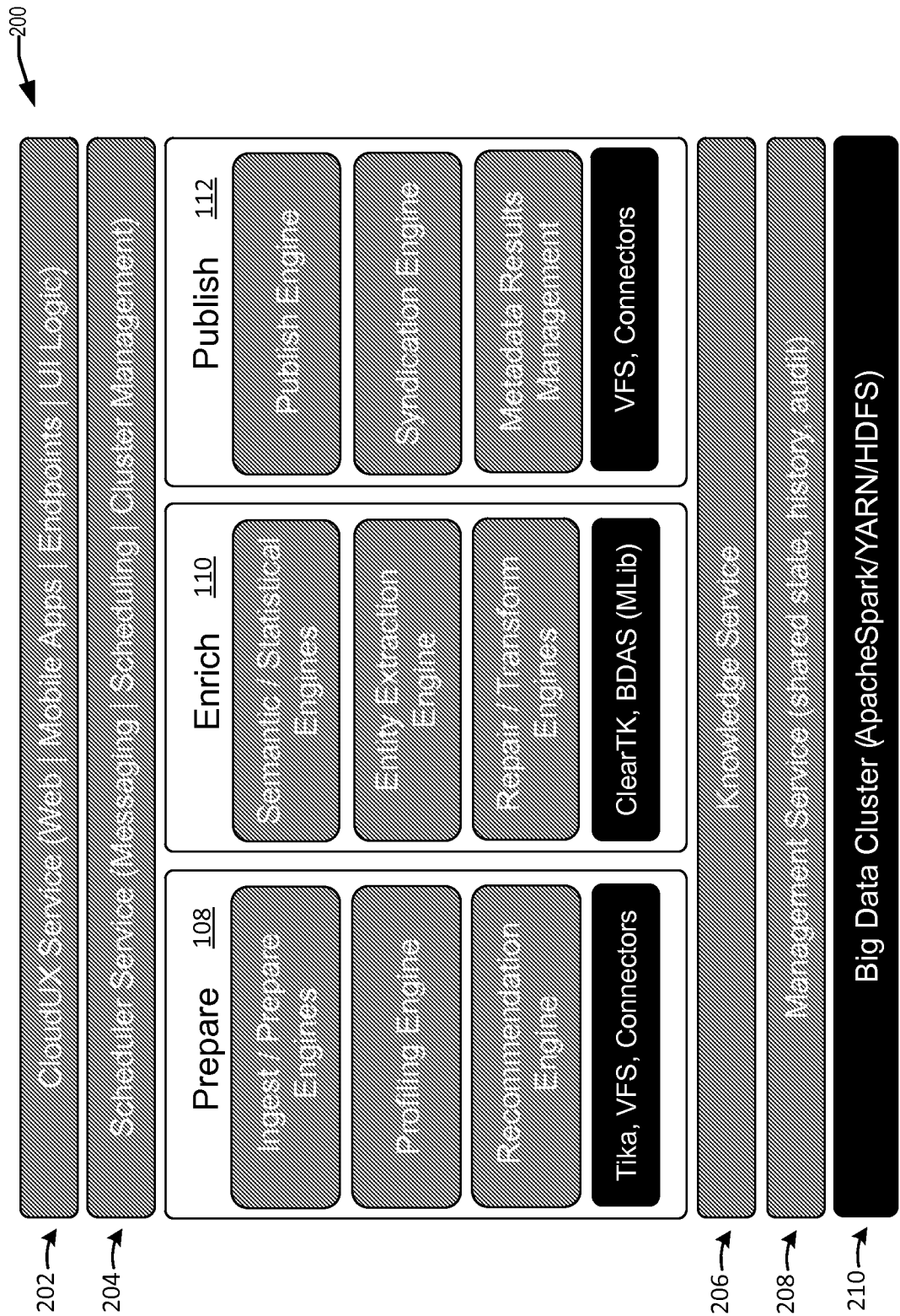


FIG. 2

3/15

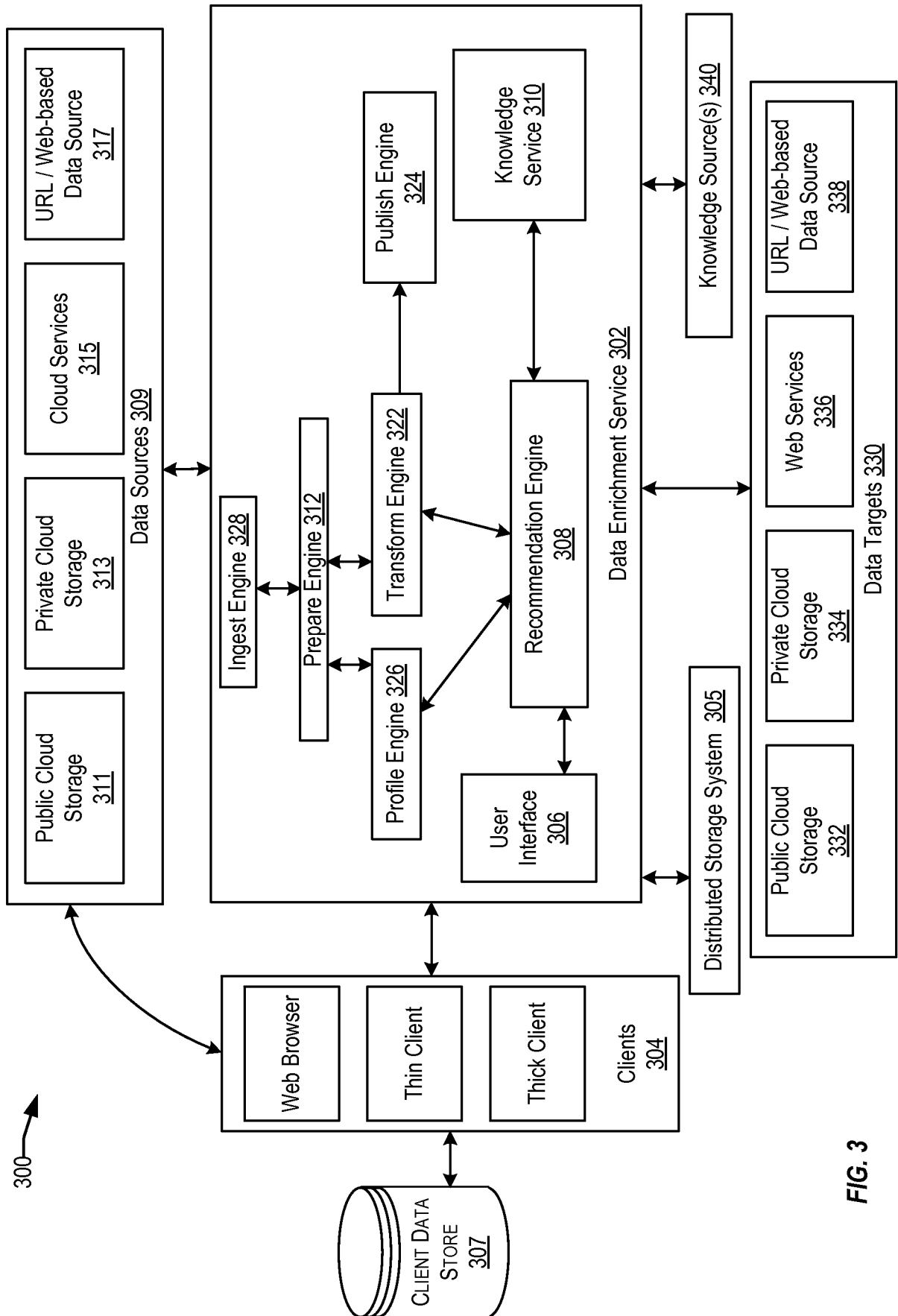


FIG. 3

Data Enrichment Cloud Service

<Catalog | pvo\_20140917\_test1

Catalog
 Dashboard
 Policies
 Documentation

| 
 Refresh | 
 Reset | 
 Undo | 
 Redo | 
 Recommendations

Save Cancel

Search

Transform Script 402

Remove 123 columns

Rename Col\_0001 to date\_time

Rename Col\_0003 to date\_time\_02

Rename Col\_0004 to url

Rename Col\_0007 to url

Rename Col\_0008 to city

Rename Col\_0010 to state

Rename Col\_0011 to hexid

Rename Col\_0019 to lp

Rename Col\_0063 to url\_02

Recommendations for All 404

Format epoch date as MM/dd/yyyy for column ...
 Format epoch date as MM/dd/yyyy for column ...
 Format epoch date as MM/dd/yyyy hh:mm:ss aa
 Format date as yy-M-d H:mm:s z for column dat...
 Remove time from date\_time\_02
 Remove date from date\_time\_02

	date_time_02	date
2012-03-12 09:21:45; 2012-03-07 18:47:25; 2012-03-12 09:35:37; 2012-03-12 08:50:35; 2012-03-06 19:03:37; 2012-03-06 18:59:39; 2012-03-12 09:25:46 2012-03-06 19:08:00 2012-03-15 18:56:34		
http://www.acme.com/SH55126545/VD5517927; http://www.acme.com/SH55126545/VD55179433; http://www.acme.com/SH55126545/VD55149415; http://www.acme.com/SH55126545/VD55170364; http://www.acme.com/SH5587637/VD55178699; http://www.acme.com/SH55126545/VD55173061; http://www.acme.com/SH559040/VD55175948; http://www.acme.com/SH5580165/VD55173281; http://www.acme.com/SH5580165/VD55156528; http://www.acme.com/SH55126545/VD55147564	string	
shoes; clothing; movies; handbags; home&garden; tools; outdoors; games; computers; automotive	string	
WABC; WOJ; KLKN; WJBF; WFTS; WPVI; KSTP; WLS; KSAT	string	
comcast.net; rr.com; qwest.net; verizon.net; 66.231.145.174; morrisbb.com; mercydesmoines.org; clearwire-wmx.net; windstream.net; 99.126.116.121	string	
hawthorne; hendersonville; seminole; adel; new york; carpentersville; eden prairie; chicago; wahoo; charlotte	string	
usa	string	
fl; ne; nc; il; ia; mn; pa; in; ny; tx	string	
D84F550B-FD75-422A-AB3F-F514357BA6AC; E733FA17-4501-42BF-A1CB-1C5BA9AD5ED2; 36F9EA4-9D76-44AE-BC08-6014DFB65B2D; B7C591A2-9277-40CE-87C4-EB26FFC1BC7; 6186EABD-7700-F17F-0809-5C717AC6CC9; B15166CA-CFA1-48B9-A884-3C49292AC5F6; 2A078DC0-007E-4C55-B4B4-2CD7A9D2E946; 48B3B5EA-CD07-B65D-DB03-C35C19CD9FEF; D4A080B5-E392-4F73-854C-0FE220AEC3D3B; B0DD8E45-455D-42C7-8AEF-BC2C65D14CD3	string	
25-Sep-89; 15-Sep-88; 26-Mar-85; 20-Sep-91; 30-Apr-74; 17-Feb-87; 14-Aug-88; 1-Aug-93; 14-Oct-88; 30-Nov-85	string	
F; M; U	string	

FIG. 4A

| 
 Refresh | 
 Undo | 
 Redo | 
 Recommendations

400

---

### Transform Script

Remove Col_0001 to date_time_02
Rename Col_0002 to date_time
Rename Col_0003 to date_time_02
Rename Col_0004 to url
Rename Col_0007 to url
Rename Col_0010 to state
Rename Col_0011 to hexid
Rename Col_0019 to ip
Rename Col_0063 to url_02

**412**

---

### Recommendations for All

Extract quarter_year from date_time_02
Extract year from date_time_02
Rename Col_0008 to city
Enrich column Col_0008 with city_state
Enrich column Col_0008 with city_county
Enrich column Col_0008 with city_population
Enrich column Col_0008 with city_lat

---

date_time_02	date
2012-03-12 09:21:45; 2012-03-07 18:47:25; 2012-03-12 09:25:37; 19:03:37; 2012-03-06 18:59:39; 2012-03-12 09:25:46 2012-03-06	string
url	string
Col_0005	string
Col_0006	string
uri	string
Col_0008	string
Col_0009	string
state	string
hexid	string
410	
Col_0012	string
Col_0013	string
Col_0014	number

http://www.acme.com/SK55126545/VU55177927; http://www.acme.com/SK55126545/VU55149415; http://www.acme.com/SK55126545/VU55178699; http://www.acme.com/SK55126545/VU55175948; http://www.acme.com/SK55126545/VU55156528; http://www.acme.com/shoes; clothing; movies; handbag; home&garden; tools; outdoor; WABC; WOJ; KERN; WCJB; WJBF; WFTS; WPVI; KSTP; WLS; KSAT contrast.net; m.com; qwest.net; verizon.net; 56.23.1.145; morris@vixx.net; windstream.net; 99.126.116.121

hawthorne; hendersonville; sunnyside; adel; new york; carpenterville

fl; ne; nc; il; la; mn; pa; nj; ny; tx

D84F5508-FD75-422A-AB3F-F5143578A6AC; E733FA17-4501-42BE-8C08-6014D8E6B2D; B7C9%1A2-9277-40CE-B7C4-FBC26FFC1BC7; 81516GCA-CFA1-48B9-A884-3C49292AC5F6; 2A078DCD-007E-4C%5-D803-C35C19CD9FEF; DAA08085-E392-4F73-854C-0FE20AFC3D38

25 Sep-89; 15-Sep-88; 26 Mar-85; 20-Sep-81; 30 Apr-74; 17-Feb-87;

P; M; U

2835601809/91661455; 2836075983683109491; 2807235783486

**FIG. 4B**



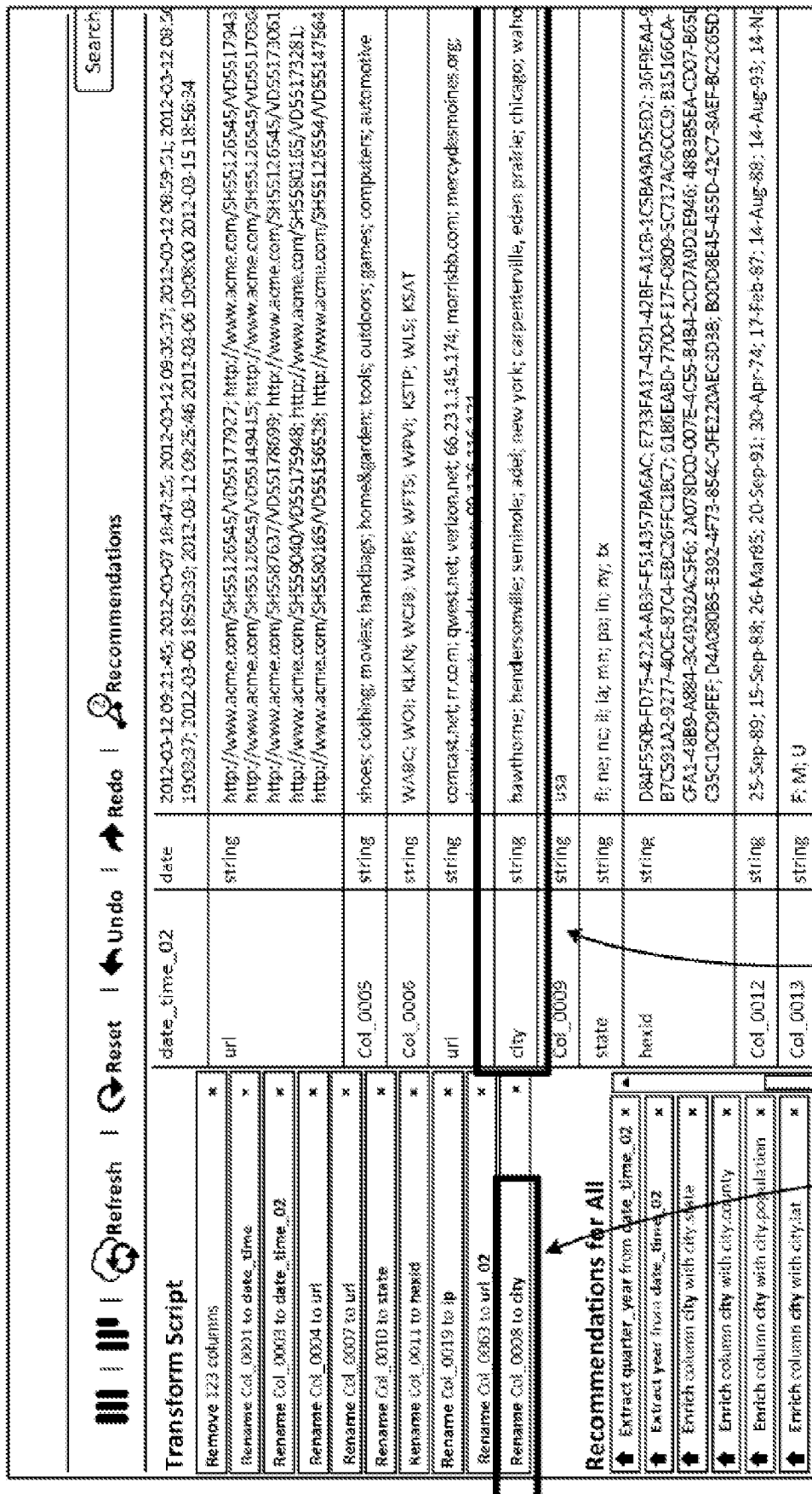


FIG. 4C

415

附註

400

Refresh

Undo

Redo

Recommendations

Save

Cancel

Search

Transform Script

Rename Col\_0001 to date\_time

Rename Col\_0003 to date\_time\_02

Rename Col\_0004 to url

Rename Col\_0007 to url

Rename Col\_0010 to state

Rename Col\_0011 to hexid

Rename Col\_0019 to ip

Rename Col\_0063 to url\_02

Rename Col\_0008 to city

Enrich column city with city.population

Enrich column city with city.lat

Recommendations for All

Extract day \_\_ of \_\_year from date\_time\_02

Extract month from date\_time\_02

Extract quarter from date\_time\_02

Extract quarter from date\_time\_02

Extract year from date\_time\_02

Enrich column city with city.state

date\_time\_02

date

url

Col\_0005

Col\_0006

uri

city

lon

lat

population

Col\_0009

state

hexid

2012-03-12 09:21:45; 2012-03-07 18:47:25; 2012-03-12 09:35:37; 2012-03-12 08:59:51; 2012-03-12 08:50:35; 2012-03-06 19:03:37; 2012-03-06 18:59:39; 2012-03-12 09:25:46 2012-03-06 19:08:00 2012-03-15 18:56:34

http://www.acme.com/SH55126545/VD55177927; http://www.acme.com/SH55126545/VD55179433; http://www.acme.com/SH55126545/VD55149415; http://www.acme.com/SH55126545/VD55170364; http://www.acme.com/SH5587637/VD55178699; http://www.acme.com/SH55126545/VD55173061; http://www.acme.com/SH559040/VD55175948; http://www.acme.com/SH5580165/VD55173281; http://www.acme.com/SH5580165/VD55156528; http://www.acme.com/SH55126554/VD55147564

shoes; clothing; movies; handbags; home&garden; tools; outdoors; games; computers; automotive

WABC; WOI; KLNK; WCJB; WJBF; WFTS; WPVI; KSTP; WLS; KSAT

comcast.net; rr.com; qwest.net; verizon.net; 66.231.145; morrisbb.com; mercydesmoines.org; cleanwire-wmx.net; windstream.net; 99.126.116.121

hawthorne; hendersonville; seminole; adel; new york; carpenterville

-118.35; -86.6; -82.79; -83.42; -73.94; -88.27; -93.46; -87.68; -96.62; -80.83

33.91; 36.29; 27.84; 31.14; 40.67; 42.12; 44.85; 41.84; 41.21; 35.2

84716; 48533; 18934; 5422; 8413351; 38298; 61754; 2868594; 3953; 705567

usa

fl; ne; nc; il; ia; mn; pa; in; ny; tx

D84F550B-FD75-422A-AB3F-F514357BA6AC; E733FA17-4501-42BF-A1CB-1C5BA9AD5ED2; 36F9EA4-9D76-44AE-BC08-6014D B7C591A2-9277-40CE-87C4-EBC26FC1BC7; 6186EABD-7700-F17F-0809-5C717AC6CC9; B15166CA-

FIG. 4D

420

418

### Transforms ~ 504

☐ Import ☐ Export View All ▾

---

Split data repeatedly on newline into rows x  
 Split split repeatedly on ',' x  
 Promote row 0 to header x  
 Delete empty rows x

### Recommendations ~ 506

☒ Remove column NationalID x  
☒ Remove punctuation NationalID x

☐ Clear All

☐ Apply All ☐ Clear All

### Clickstream ~ 508

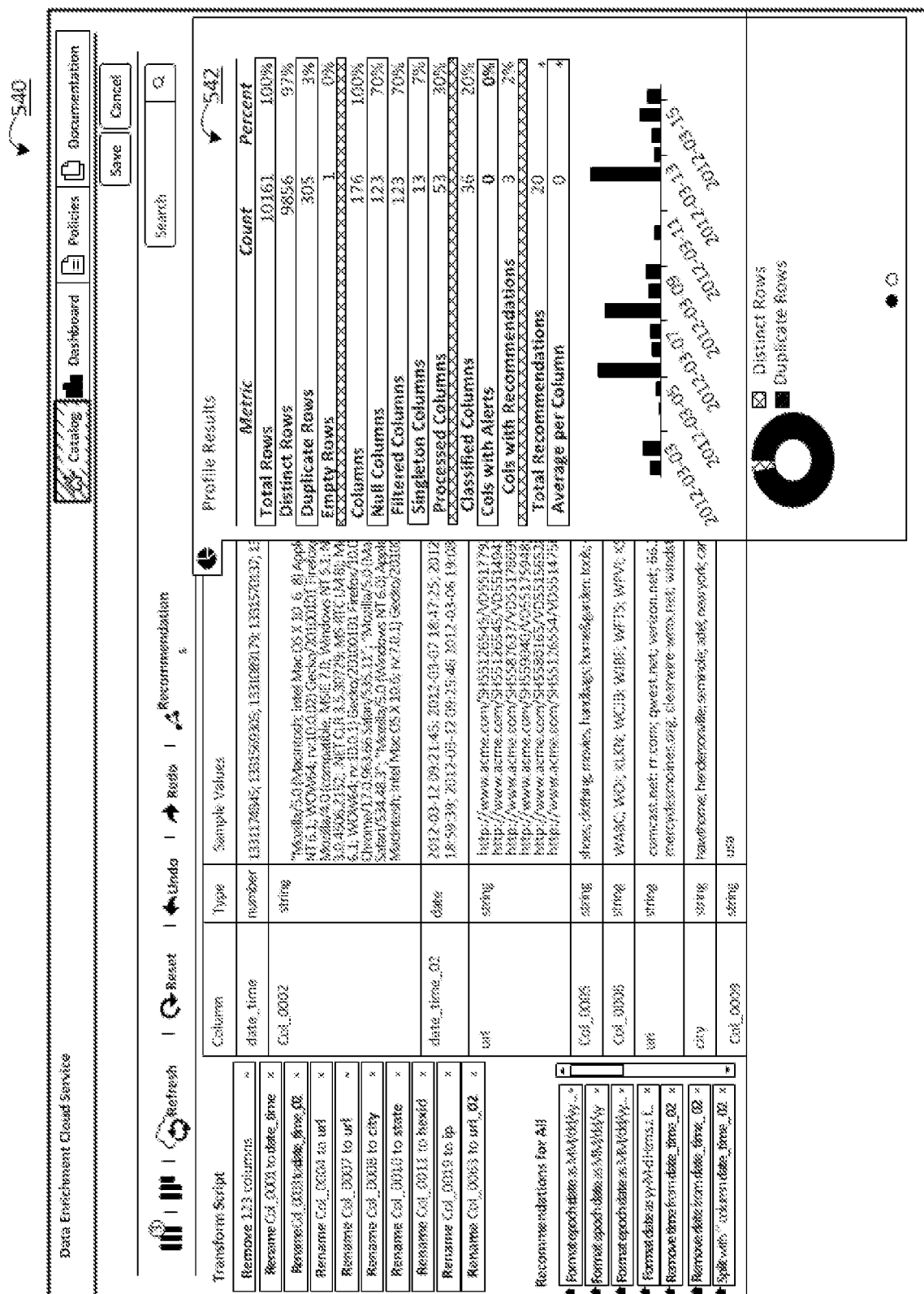
||| |
☐ Undo ☐ Redo ☐ Reset ☐ Refresh

Columm	Type	Sample Values
id	NUM	***** , ***** , *****
unknown 1	ABC	Mozilla/5.0 (Windows NT 6.1); WOW64; rv 10.0.2
date_time	DATE	2012-03-14 20:48:58; 2012-03-14 20:50:13; 2012-03-14 20:59:15; 2012-
url	ABC	http://www.acme.com; http://www.acme.com/SH5582037VD5582082
category	ABC	accessories; computers; books; clothing; home&garden; movies; automo
local_broadcaster	ABC	WPLG; KXLY; WTEN; WKRT
major_broadcaster	ABC	KGO; LK MBC; WABC
unknown2	ABC	swbell.net; sbcflobal.net; comcast.net; cox.net; ameritech.neoxr.com
city	ABC	corsicana; oklahoma; city; west; hartford; lehi; san diego; topeka
state	ABC	oklahoma; new york; new jersey; california; texas; florida
category	ABC	accessories; computers; books; clothing; home&garden; movies; automo
local_broadcaster	ABC	WPLG; KXLY; WTEN; WKRT
major_broadcaster	ABC	KGO; LK MBC; WABC
unknown2	ABC	swbell.net; sbcflobal.net; comcast.net; cox.net; ameritech.neoxr.com
city	ABC	corsicana; oklahoma; city; west; hartford; lehi; san diego; topeka
state	ABC	oklahoma; new york; new jersey; california; texas; florida

Page 2 of 24 (1-8 of 24 items)    K < 1 2 3 > >I > >I

**FIG. 5A**

9/15



396

10/15

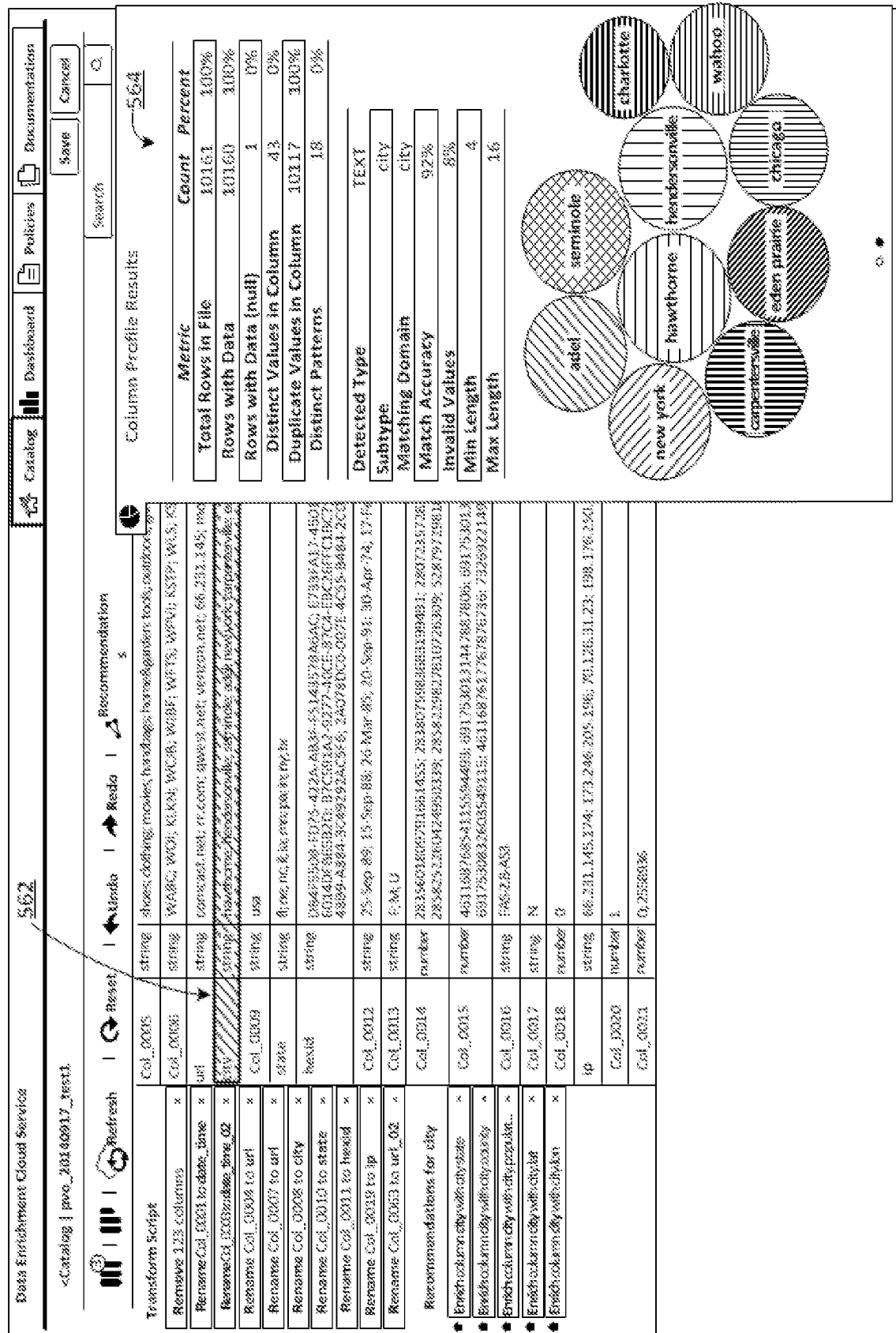
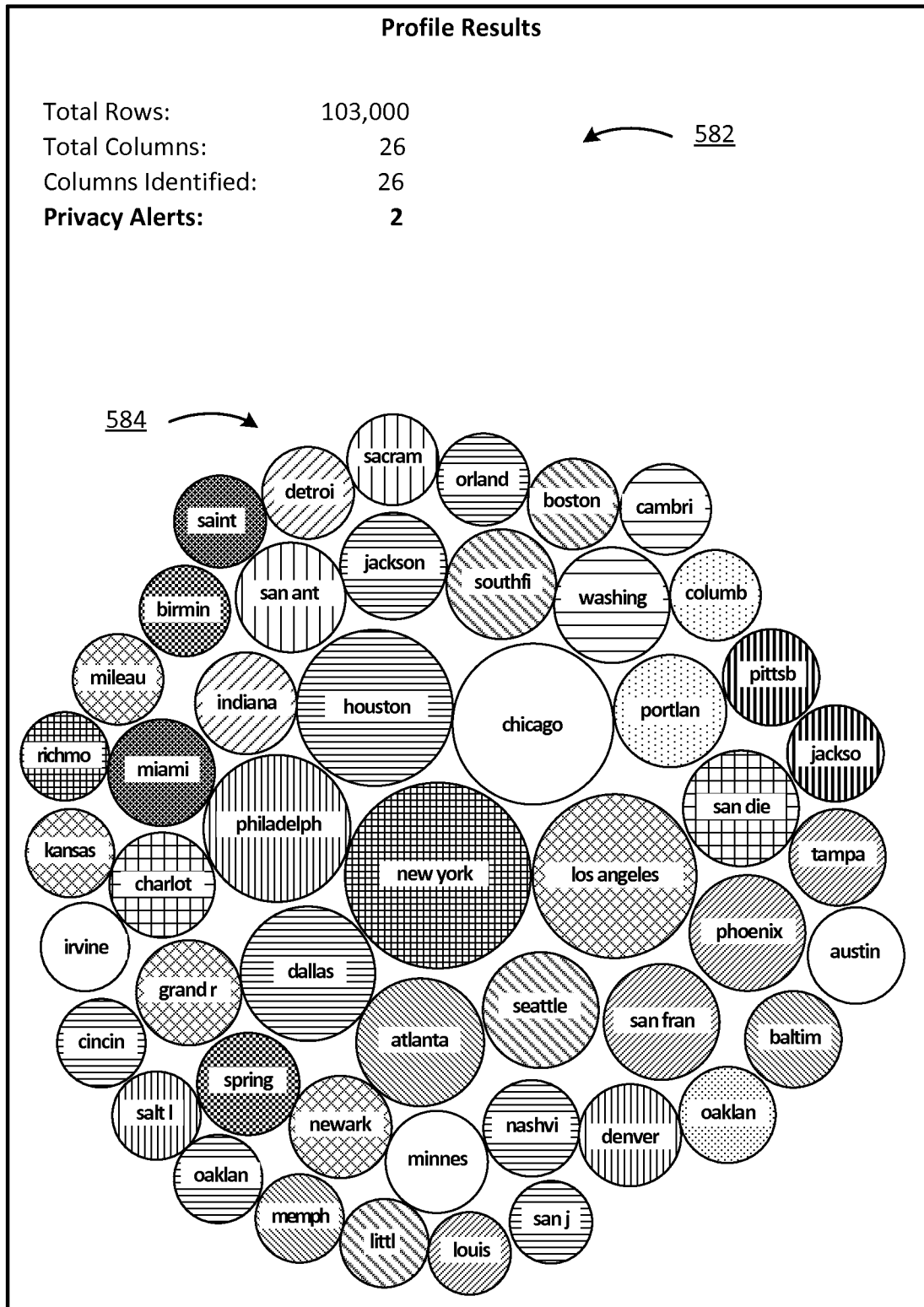
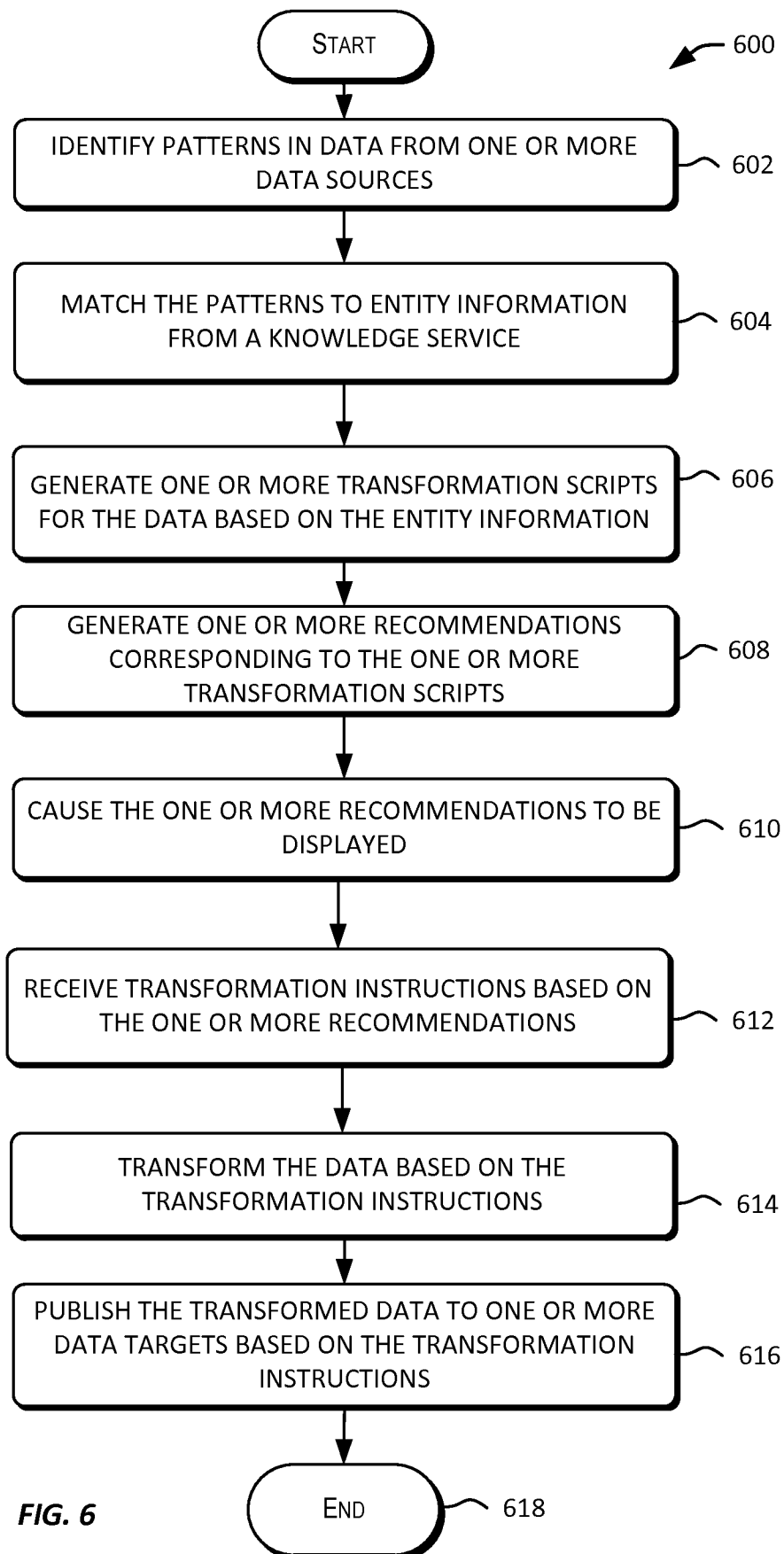


FIG. 5C



**FIG. 5D**

12/15

**FIG. 6**

13/15

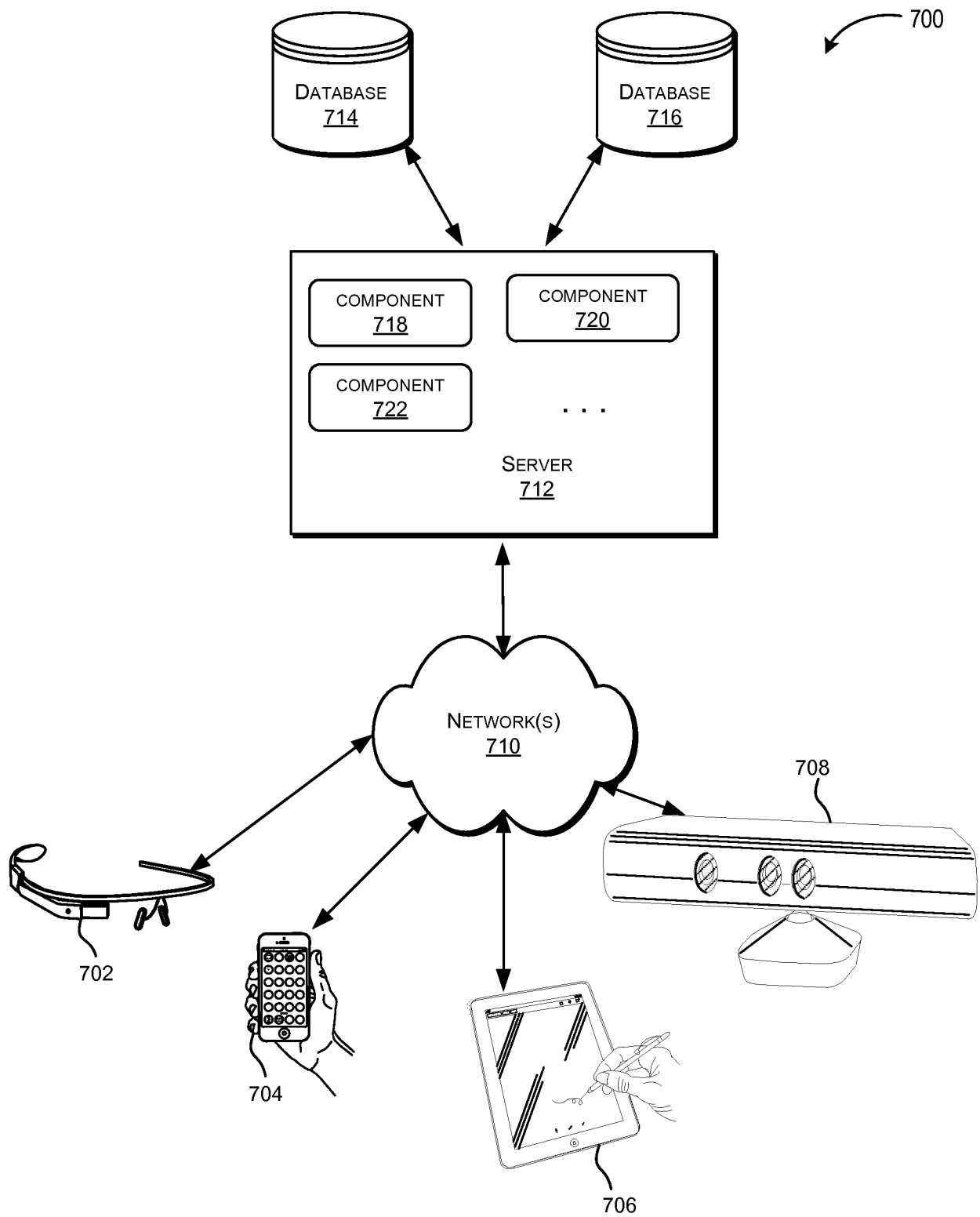


FIG. 7



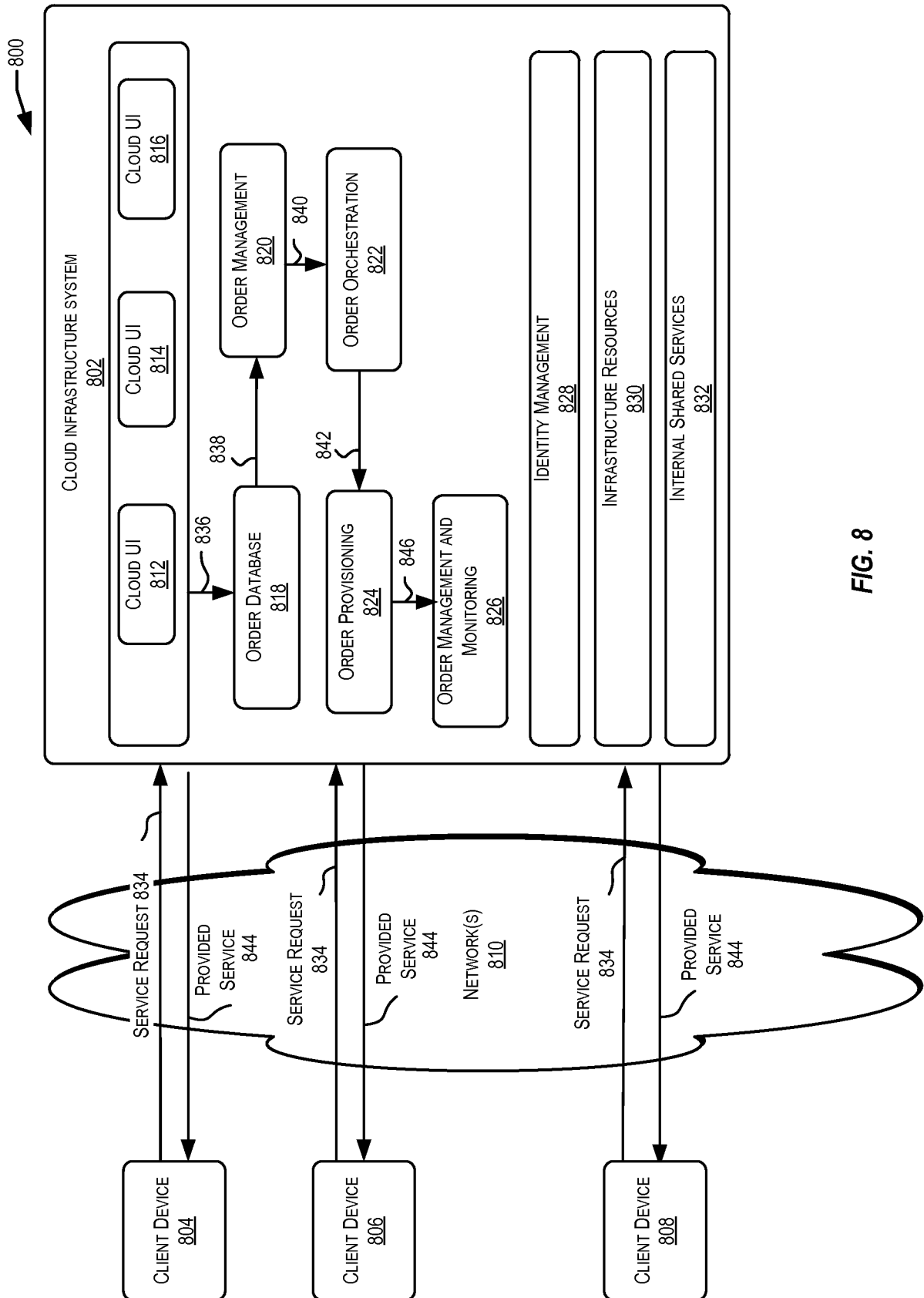


FIG. 8

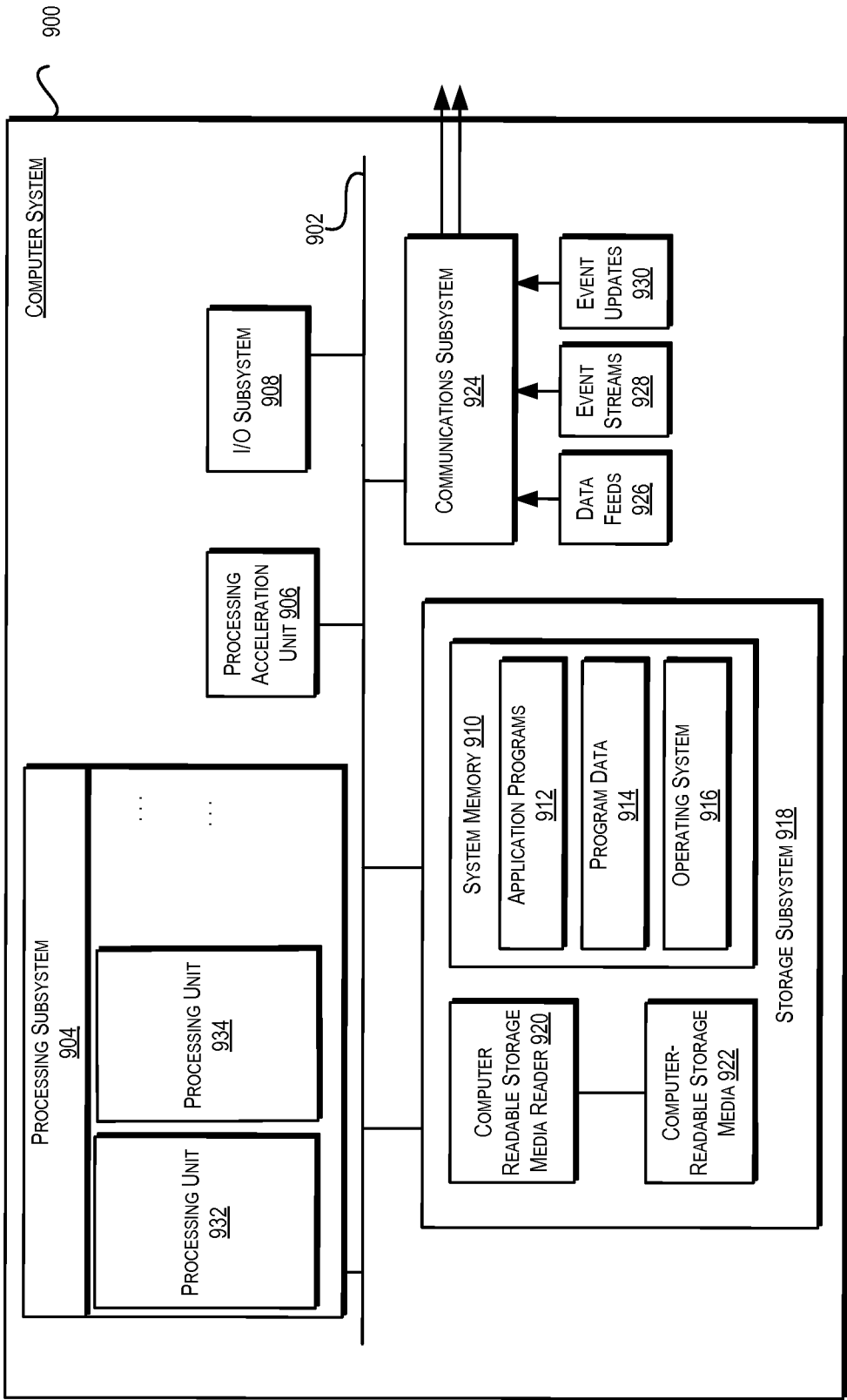


FIG. 9

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2015/052228

A. CLASSIFICATION OF SUBJECT MATTER  
INV. G06F17/30  
ADD.

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)  
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2008/281820 A1 (DO HONG-HAI [DE] ET AL) 13 November 2008 (2008-11-13) abstract; figures 1,6 paragraphs [0022] - [0032] -----	1-20
X	US 2007/112827 A1 (DETINGER RICHARD D [US] ET AL) 17 May 2007 (2007-05-17) abstract; figure 2 paragraphs [0035] - [0042] -----	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"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)

"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

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search

11 November 2015

Date of mailing of the international search report

18/11/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer

Hackelbusch, Richard

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2015/052228

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008281820 A1	13-11-2008	EP 1990740 A1	12-11-2008
		US 2008281820 A1	13-11-2008
-----			
US 2007112827 A1	17-05-2007	NONE	
-----			