



(51) International Patent Classification:
G06F 17/40 (2006.01)

(21) International Application Number: PCT/CN2016/093092

(22) International Filing Date: 3 August 2016 (03.08.2016)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) Priority Data:
14/839,226 28 August 2015 (28.08.2015) US

(71) **Applicant:** HUAWEI TECHNOLOGIES CO., LTD.
[CN/CN]; Huawei Administration Building, Bantian,
Longgang District, Shenzhen, Guangdong 518129 (CN).

(72) **Inventor: EZZAT, Ahmed Kamal**; 10438 Plum Tree Lane, Cupertino, CA 95014 (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).

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- with international search report (Art. 21(3))

(54) Title: SYSTEM AND METHOD FOR PROVIDING DATA AS A SERVICE (DAAS) IN REAL-TIME

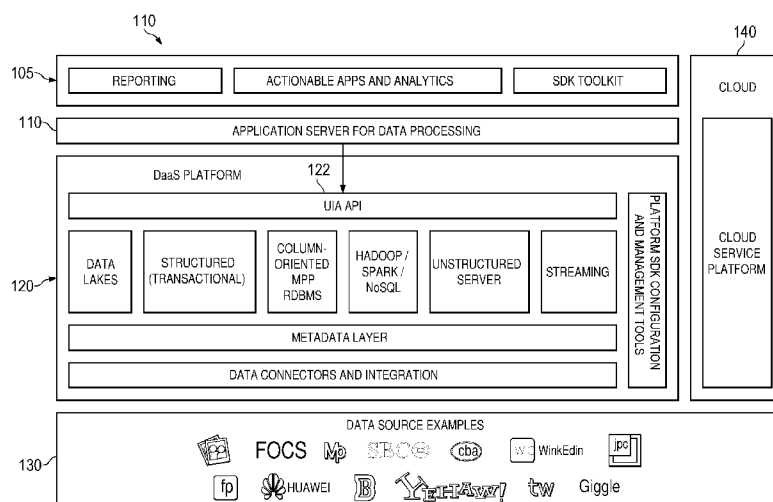


FIG. 1

(S7) Abstract: System and method embodiments are provided for implementing Data as a Service (DaaS). The system is implemented using a client side library, on a user device, and a server or proxy server to extract relevant data from relevant data sources, and process the data before returning results to the client. The client sends a user query to the proxy server, which then sends sub-queries and receive responses from multiple data sources in real-time or near real-time. The system also uses a data model that handles varying data reliability or accuracy levels in heterogeneous data sources and indicates the confidence levels in the data provided to a user or client application. The data model assigns different confidence levels for various data to distinguish between high quality data and low quality data. Thus, users are provided with more information from multiple sources without diluting high quality data with low quality data.

SYSTEM AND METHOD FOR PROVIDING DATA AS A SERVICE (DAAS) IN REAL-TIME**CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] This application claims priority to U.S. non-provisional patent application Serial No. 14/839,226, filed on August 28, 2015 and entitled “System and Method for Providing Data as a Service (DaaS) in Real-time,” which is incorporated herein by reference as if reproduced in its entirety.

TECHNICAL FIELD

[0002] The present invention relates to networking and data query, and, in particular embodiments, to a system and method for providing Data as a Service (DaaS) in real-time.

BACKGROUND

[0003] In communications systems and networks, data can be exchanged to perform functions and services, to be processed in a desired manner, and/or to request information. For example, users of mobile devices (e.g., smartphones) or other user equipment (UEs) request and process data to run applications on their devices or to display the data in some form that conveys information to the users. For example, the requested data can be viewed in web browsing applications, emails, maps, weather applications, or any other suitable form according to user needs. Networks such as the Internet provide access to a reservoir of information to users around the globe. The users may access data via any suitable and available network services and equipment.

[0004] A user may request data from multiple sources to obtain information of some content or topic. However, the quality and reliability of the data may vary depending on the sources. Assimilating and filtering through such data in a manner that improves the reliability and increase the amount or value of information is beneficial to users and provides better quality of

service. Benefit is further added when the information is provided in real-time. This is the case when users request data on UEs using available network connection rates and expect practically instant or short delay responses. Thus, there is a need for a scheme that can obtain requested data from multiple sources of variable reliability and process the data to present, in real-time, information to users.

SUMMARY OF THE INVENTION

[0005] In accordance with an embodiment, a method for providing data as a service (DaaS) in near or real-time includes receiving, at a proxy server, a query from a client application, and requesting, by the proxy server from a plurality of data sources, data to service the query. The data sources are heterogeneous in term of data quality, or data structure, or both data quality and data structure. The method further includes obtaining the data from each data source in near or real-time, assigning a confidence level to the data from each data source in accordance with reliability of the data source, and joining the data including the confidence level from the data sources. The joined data is sent back to the client application.

[0006] In accordance with another embodiment, a method for providing DaaS in near or real-time includes sending, by a client application on a user device a query to a proxy server, and receiving from the proxy server, in near or real-time, a joined response from a plurality of data sources. The joined response includes responses from the data sources with confidence levels associated with the responses in accordance with reliability of the corresponding data sources.

[0007] In accordance with another embodiment, a network server for providing DaaS in near or real-time includes a processor and a non-transitory computer readable storage medium storing programming for execution by the processor. The programming includes instructions to receive a query from a client application, and request, from a plurality of data sources, data to service the query. The data sources are heterogeneous in term of data quality or data structure. The programming includes further instructions to obtain the data from each data source in near or real-time, assign a confidence level to the data from each data source in accordance with reliability of the data source, join the data including the confidence level from the data sources, and send the joined data to the client application.

[0008] In accordance with yet another embodiment, a user device for providing DaaS in near or real-time includes a processor and a non-transitory computer readable storage medium storing programming for execution by the processor. The programming includes instructions to send to a server proxy a query and receive from the server proxy, in near or real-time, a joined response from a plurality of data sources. The joined response includes responses from the data sources with confidence levels associated with the responses in accordance with reliability of the corresponding data sources.

[0009] In accordance with another embodiment, a network server is used to provide DaaS in near or real-time. The network server includes a receiving element that receives, at a proxy server, a query from a client application; a requesting element that requests, by the proxy server from a plurality of data sources, data to service the query, wherein the data sources are heterogeneous in term of data quality, or data structure, or both data quality and data structure; an obtaining element that obtains the data from each data source in near or real-time; an assigning element that assigns a confidence level to the data from each data source in accordance with reliability of the data source; a joining element that joins the data including the confidence level from the data sources; and a sending element that sends the joined data back to the client application.

[0010] In accordance with another embodiment, a user device is used to provide data as a service (DaaS) in near or real-time. The user device includes a sending element that sends, by a client application on a user device a query to a proxy server; and a receiving element that receives from the proxy server, in near or real-time, a joined response from a plurality of data sources, wherein the joined response includes responses from the data sources with confidence levels associated with the responses in accordance with reliability of the corresponding data sources.

[0011] The foregoing has outlined rather broadly the features of an embodiment of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of embodiments of the invention will be described hereinafter, which form the subject of the claims of the invention. It should be appreciated by those skilled in the art that the conception and specific embodiments disclosed may be readily utilized as a basis for modifying or designing other structures or processes for

carrying out the same purposes of the present invention. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the spirit and scope of the invention as set forth in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawing, in which:

[0013] Figure 1 illustrates an embodiment of a system for providing Data as a Service (DaaS) in real-time;

[0014] Figure 2 illustrates another embodiment of a system for providing DaaS in real-time;

[0015] Figure 3 illustrates an embodiment method for implementing DaaS in real-time; and

[0016] Figure 4 is a diagram of a processing system that can be used to implement various embodiments.

[0017] Corresponding numerals and symbols in the different figures generally refer to corresponding parts unless otherwise indicated. The figures are drawn to clearly illustrate the relevant aspects of the embodiments and are not necessarily drawn to scale.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0018] The making and using of the presently preferred embodiments are discussed in detail below. It should be appreciated, however, that the present invention provides many applicable inventive concepts that can be embodied in a wide variety of specific contexts. The specific embodiments discussed are merely illustrative of specific ways to make and use the invention, and do not limit the scope of the invention.

[0019] System and method embodiments are provided for implementing Data as a Service (DaaS). The systems can be implemented using a client side library and a proxy server to extract relevant data from relevant data sources, and process the data before returning the results back to the client. The client side library can use an application programming interface (API) to send a

user query to a server or proxy server. The proxy server is configured to send sub-queries and receive responses for the sub-queries from multiple data sources (e.g., databases) in real-time or near real-time. The data can be obtained from the data sources in real-time without integrating and storing the data at a data warehouse (DW) separate from the data sources and the requesting device. Near real-time response can be achieved by querying the data sources rather than copying the data from the sources to a DW and then querying the DW. The data sources can be heterogeneous in terms of reliability, content format, type, or other features. The reliability of the data represents the data accuracy and/or the confidence in the reliability of the data source.

[0020] The embodiments also include a data model that handles the variable data reliability/quality/accuracy levels in the heterogeneous data sources and determines the confidence levels in the provided data to a user or client application. The data model is configured to assign different confidence levels for various data, thereby distinguishing between and presenting high quality data and low quality data to users. In an embodiment, the data model represents each data value by the 3-attributes <key, value, probability>, where the key field is used to identify the data, and the probability field indicates a confidence level in the data or source. The confidence level depends on the reliability of the source. High quality data reflects high confidence in the source, while low quality data reflects low confidence in the source. The probability field can be represented as a percentage, as a value that scores the confidence level (e.g., between 0 and 1), or any other indicator of the confidence in the source or the data quality/reliability. Since varying quality data is also considered in the data model, the information made available to users by this system is increased without diluting the high quality data with low quality data.

[0021] Figure 1 shows an embodiment of a system 100 including various components for providing DaaS in real-time. A DaaS platform 120 serves as an interface between a plurality of data sources 130 and an application server 110 for data processing. The application server 110 receives queries for data from client applications 105 and responds with the data. The client application 105 can run on a UE or any other user device that communicates with the DaaS platform 120. As used herein, the term user device includes any device operated by a user or other devices such as machine-to-machine (M2M) devices and sensor devices. The DaaS

platform 120 sends the queries to and receives in response the data from various (heterogeneous) data sources. Examples of data sources include network services such WikipediaTM, GoogleTM, FacebookTM, TwitterTM, The Weather ChannelTM, Linked inTM, You TubeTM, or other online data services. Further examples of data sources are shown in Figure 1. The application layer 105, the application server 110 and the DasS platform are software components that can be implemented at least partially using a cloud service platform 140, e.g., on a UE and one or more servers accessible by the Internet.

[0022] The DaaS server 110 includes an API 122 for handling a query with multiple data sources. The API 122 sends the query from a client application (e.g., on a UE) to multiple data sources and integrates the information received from the sources to respond with data back to the user. A single API 122, referred to herein as Unified Information Access (UIA) API, serves as a single client API used to connect to the proxy server over a network passing the user query, forward the query to and receive information from multiple heterogeneous data sources, and return the information back to the client. To handle the queries for the client, the UIA API 122 can interact with multiple engines (e.g., servers) or services (e.g., in the cloud) that requests the various data formats from heterogeneous sources and forwards the corresponding data to the API 122. The UIA API 122 can send sub-queries to the engines in the native language of the data sources to request the data. Examples of such engines/services include data lakes, structured (transactional) data servers, column oriented massively parallel processing (MPP) relational database management systems (RDBMS), unstructured data servers, streaming services, engines such as HADOOP, Spark, NoSQL, or others. The engines/services can interact with the various data sources to obtain the data using a metadata layer, data connectors and integration. The components of the application layer 105 and the DaaS platform 120 can be developed and configured by software developers using software development kits (SDKs).

[0023] Figure 2 shows another embodiment of a system 200 including various components for providing DaaS in real-time. A DaaS platform 220 serves as an interface between a client application 210 (e.g., at a UE) and a plurality of feeds 230 belonging to heterogeneous data sources. The data sources provide data of different formats to the DaaS platform 220 in response to a query from the client application 210. The DaaS platform 220 includes a UIA API 221 that

operates as an interface between the client application 210 and various engines that handle the different data feeds 230. Examples of such engines include online transaction processing (OLTP) servers and MPP RDBMS for handling structured data, unstructured data engines for handling unstructured data, and streaming services for handling streamed data. The DaaS platform 220 may also include services, such as Hadoop MPP RDBMS and streaming services, that exchange (send/receive) data between each other in response to queries from the UIA API 221.

[0024] In the systems above, the client application can access data relevant to the user query, from multiple data sources, using the native query language of each queried data source. The user query can be written in SQL and leverage Table-Valued User Defined Functions (TVUDFs). A SQL engine can be used as the proxy server to coordinate executing a different TVUDF for each of the relevant data sources, get the results back from the data sources and translate each into a temporary/virtual table, and then perform a JOIN operation between the relevant base tables and the virtual tables and return the results in a standard SQL data type format. The TVUDF allows mapping the result returned from a given data source in the SQL engine into a virtual table. The virtual tables, which represent the output of TVUDFs, are deleted automatically by the SQL engine at the end of the client query execution. Deleting the virtual tables prevents unnecessary increase of storage size for processing the queries and data. An example of a query process can be in the following form:

```
SELECT  x, y, ... , z
FROM T1, TVUDF1(...) AS T2,
        TVUDF2(...) AS T3,
        TVUDF3(...) AS T4
WHERE   predicate1 and
        predicate2 and
        (conf(x) > 0.9 and conf(y) > 0.8);
```

The TVUDF takes a string representing the native query to a specific external data source and returns a virtual table to be used in computing the client result to the above query.

[0025] The systems above allow query responses from different data sources to have different data quality or reliability with various degrees of confidence, as described above. This can be achieved using a probabilistic data model which includes a probability value that serves as a confidence indicator in the data according to the reliability or accuracy of the source. Thus, the high quality data responses to the query can be combined but are not convoluted with the low quality responses. By including data of different quality in the response to a query, the information value is increased.

[0026] Figure 3 is a flowchart of an embodiment method 300 for implementing DaaS in real-time. For instance, the method can be implemented by the system 100 or 200. At step 310, a client application on a UE sends a query for data to a proxy server or UIA API, e.g., in the cloud. The client application is any application that requests information. For example, the client application can be an app on a mobile device operated by a user. The query can be sent in SQL and using the TVUDF and may indicate a desired data quality or reliability of response. At step 320, the UIA API generates one or more virtual tables for collecting response data from various sources to service the query. The virtual tables can be temporarily stored at a DW or any suitable data repository. At step 330, the UIA API interacts with various engines/services to request the data from the data sources, which are connected to the cloud for example. At step 340, the engines/services (in the DaaS platform) collect the data in real-time from the sources and return the data to the UIA API. At step 350, the UIA API adds a probability parameter to data from each source to indicate the confidence level in the data. The data and probability values can be inserted in the corresponding virtual tables generated to service the query. Alternatively, a probability or confidence indicator can be provided with the data from the sources or the engines/services collecting the data. At step 360, a joint SQL function is performed on the virtual tables to send the response, including data and probability, to the client application. The response may include responses from one or more data sources with corresponding confidence levels that meet the desired quality of response by the client application. In absence of an indication for quality of response by the client application, the joined response includes responses with varying confidence levels from all considered data sources with varying reliability of data. After receiving the response to the query, the user may select the relevant data. At step 370, the virtual tables are deleted upon completing the query request. The method 300 can be performed without

persistent storage of data at a DW. However, the data can be temporarily stored, e.g., in virtual tables, at a DW or any suitable storage mechanism for real-time processing.

[0027] Figure 4 is a block diagram of a processing system 400 that can be used to implement various embodiments. The processing system 400 can be part of a wireless communications terminal/device such as a user equipment (UE), e.g., a smart phone, tablet computer, a laptop, or a desktop computer. The system can also be part of a network entity or component that communicates with the terminal, such as a server in the cloud or any suitable network connected to the terminal and data sources. Specific devices may utilize all of the components shown, or only a subset of the components, and levels of integration may vary from device to device. Furthermore, a device may contain multiple instances of a component, such as multiple processing units, processors, memories, transmitters, receivers, etc. The processing system 400 may comprise a processing unit 401 equipped with one or more input/output devices, such as a speaker, microphone, mouse, touchscreen, keypad, keyboard, printer, display, and the like. The processing unit 401 may include a central processing unit (CPU) 410, a memory 420, a mass storage device 430, a video adapter 440, and an I/O interface 460 connected to a bus. The bus may be one or more of any type of several bus architectures including a memory bus or memory controller, a peripheral bus, a video bus, or the like.

[0028] The CPU 410 may comprise any type of electronic data processor. The memory 420 may comprise any type of system memory such as static random access memory (SRAM), dynamic random access memory (DRAM), synchronous DRAM (SDRAM), read-only memory (ROM), a combination thereof, or the like. In an embodiment, the memory 420 may include ROM for use at boot-up, and DRAM for program and data storage for use while executing programs. In embodiments, the memory 420 is non-transitory. The mass storage device 430 may comprise any type of storage device configured to store data, programs, and other information and to make the data, programs, and other information accessible via the bus. The mass storage device 430 may comprise, for example, one or more of a solid state drive, hard disk drive, a magnetic disk drive, an optical disk drive, or the like.

[0029] The video adapter 440 and the I/O interface 460 provide interfaces to couple external input and output devices to the processing unit. As illustrated, examples of input and output devices include a display 490 coupled to the video adapter 440 and any combination of

mouse/keyboard/printer 470 coupled to the I/O interface 460. Other devices may be coupled to the processing unit 401, and additional or fewer interface cards may be utilized. For example, a serial interface card (not shown) may be used to provide a serial interface for a printer.

[0030] The processing unit 401 also includes one or more network interfaces 450, which may comprise wired links, such as an Ethernet cable or the like, and/or wireless links to access nodes or one or more networks 480. The network interface 450 allows the processing unit 401 to communicate with remote units via the networks 480. For example, the network interface 450 may provide wireless communication via one or more transmitters/transmit antennas and one or more receivers/receive antennas. In an embodiment, the processing unit 401 is coupled to a local-area network or a wide-area network for data processing and communications with remote devices, such as other processing units, the Internet, remote storage facilities, or the like.

[0031] While several embodiments have been provided in the present disclosure, it should be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted, or not implemented.

[0032] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, modules, techniques, or methods without departing from the scope of the present disclosure. Other items shown or discussed as coupled or directly coupled or communicating with each other may be indirectly coupled or communicating through some interface, device, or intermediate component whether electrically, mechanically, or otherwise. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and could be made without departing from the spirit and scope disclosed herein.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for providing data as a service (DaaS) in near or real-time comprising:
receiving, at a proxy server, a query from a client application;
requesting, by the proxy server from a plurality of data sources, data to service the query, wherein the data sources are heterogeneous in term of data quality, or data structure, or both data quality and data structure;
obtaining the data from each data source in near or real-time;
assigning a confidence level to the data from each data source in accordance with reliability of the data source;
joining the data including the confidence level from the data sources; and
sending the joined data back to the client application.
2. The method according to claim 1, further comprising:
collecting the data from the data sources into virtual tables using Table Valued User Defined Functions (TVUDFs) embedded in the query; and
deleting the virtual tables created by executing the TVUDFs upon completing execution of the query.
3. The method according to claim 1 or 2, wherein the query is received as a Structured Query Language (SQL) query with a Table Valued User Defined Function (TVUDF).
4. The method according to any one of claims 1-3, wherein the data is in a form of a three-attributes including a key identifying the data, a value of the data, and a probability value indicating the confidence level.
5. The method according to any one of claims 1-4, wherein the data is obtained and sent back to the client application without persistent storage of the data at a data warehouse.

25 6. The method according to any one of claims 1-5, wherein the data is requested from the
26 data sources via a plurality of corresponding data engines for handling the data quality or data
27 structure of the data sources in real-time.

28 7. The method according to claim 6, wherein at least some of the data engines exchange
29 between each other at least some of the data to service the request.

30 8. The method according to any one of claims 1-7, further comprising:
31 forwarding the query in native query language of each data source to a plurality of data
32 engines corresponding to the data sources; and
33 obtaining, in near or real-time, by the data engines from the data sources, data in response
34 to the query, wherein the data is joined at the proxy server.

35 9. The method according to any one of claims 1-8, wherein the server proxy is implemented
36 in a cloud based computing platform, and wherein the client application is implemented on a user
37 device.

38 10. A method for providing data as a service (DaaS) in near or real-time comprising:
39 sending, by a client application on a user device a query to a proxy server; and
40 receiving from the proxy server, in near or real-time, a joined response from a plurality of
41 data sources, wherein the joined response includes responses from the data sources with
42 confidence levels associated with the responses in accordance with reliability of the
43 corresponding data sources.

44 11. The method according to claim 10 further comprising:
45 indicating with the sent query a desired quality of response; and
46 receiving a response from one or more data sources with corresponding confidence levels
47 that meet the desired quality of response.

12. The method according to claim 10 or 11, wherein in absence of an indication for quality of response by the client application, the received joined response includes responses with varying confidence levels from the data sources with varying reliability of data.

13. The method according to any one of claims 10-12, wherein the data sources are heterogeneous in term of data quality or data structure.

14. The method according to any one of claims 10-13, wherein the query is sent using Structured Query Language (SQL) query and a table value user defined function (TVUDF).

15. The method according to any one of claims 10-14, wherein the data is obtained and returned to the client application without persistent storage of the data at a data warehouse.

16. A network server for providing data as a service (DaaS) in near or real-time, the network server comprising:

a processor; and

a non-transitory computer readable storage medium storing programming for execution by the processor, the programming including instructions to:

receive a query from a client application;

request, from a plurality of data sources, data to service the query, wherein the data sources are heterogeneous in term of data quality or data structure;

obtain the data from each data source in near or real-time;

assign a confidence level to the data from each data source in accordance with reliability of the data source;

join the data including the confidence level from the data sources; and

send the joined data to the client application.

17. The network server according to claim 16, wherein the programming includes further instructions to:

72 collect the data from the data sources into virtual tables using Table Valued User
73 Defined Functions (TVUDFs) embedded in the query; and
74 delete the virtual tables created by executing the TVUDFs upon completing execution of
75 the query.

76 18. The network server according to claim 16 or 17, wherein the programming includes
77 further instructions to:
78 forward, to a plurality of data engines corresponding to a plurality of data sources, the
79 query in native query language of each data source; and
80 obtain, in near or real-time, by the data engines of the data sources, the data in response
81 to the query, wherein the data engines handle the data structure of the corresponding data sources
82 in real-time.

83
84 19. The network server according to any one of claims 16-18, wherein the client application
85 runs on a user device, and wherein the network server communicates with the user device
86 through a cloud based computing platform.

87
88 20. A user device for providing data as a service (DaaS) in near or real-time, the user device
89 comprising:

90 a processor; and
91 a non-transitory computer readable storage medium storing programming for execution
92 by the processor, the programming including instructions to:
93 send to a server proxy a query; and
94 receive from the server proxy, in near or real-time, a joined response from a plurality of
95 data sources, wherein the joined response includes responses from the data sources with
96 confidence levels associated with the responses in accordance with reliability of the
97 corresponding data sources.

98 21. The user device according to claim 20, wherein the programming includes further
99 instructions to:

100 indicate with the sent query a desired quality of response; and

101 receive a response from one or more data sources with corresponding confidence levels
102 that meet the desired quality of response.

103 22. The user device according to claim 20 or 21, wherein the programming includes further
104 instructions to

105 in absence of an indication for quality of response by the user device, receive the joined response
106 including responses with varying confidence levels from the data sources with varying reliability
107 of data.

108 23. The user device according to any one of claims 20-22, wherein the data sources are
109 heterogeneous in term of data quality or data structure.

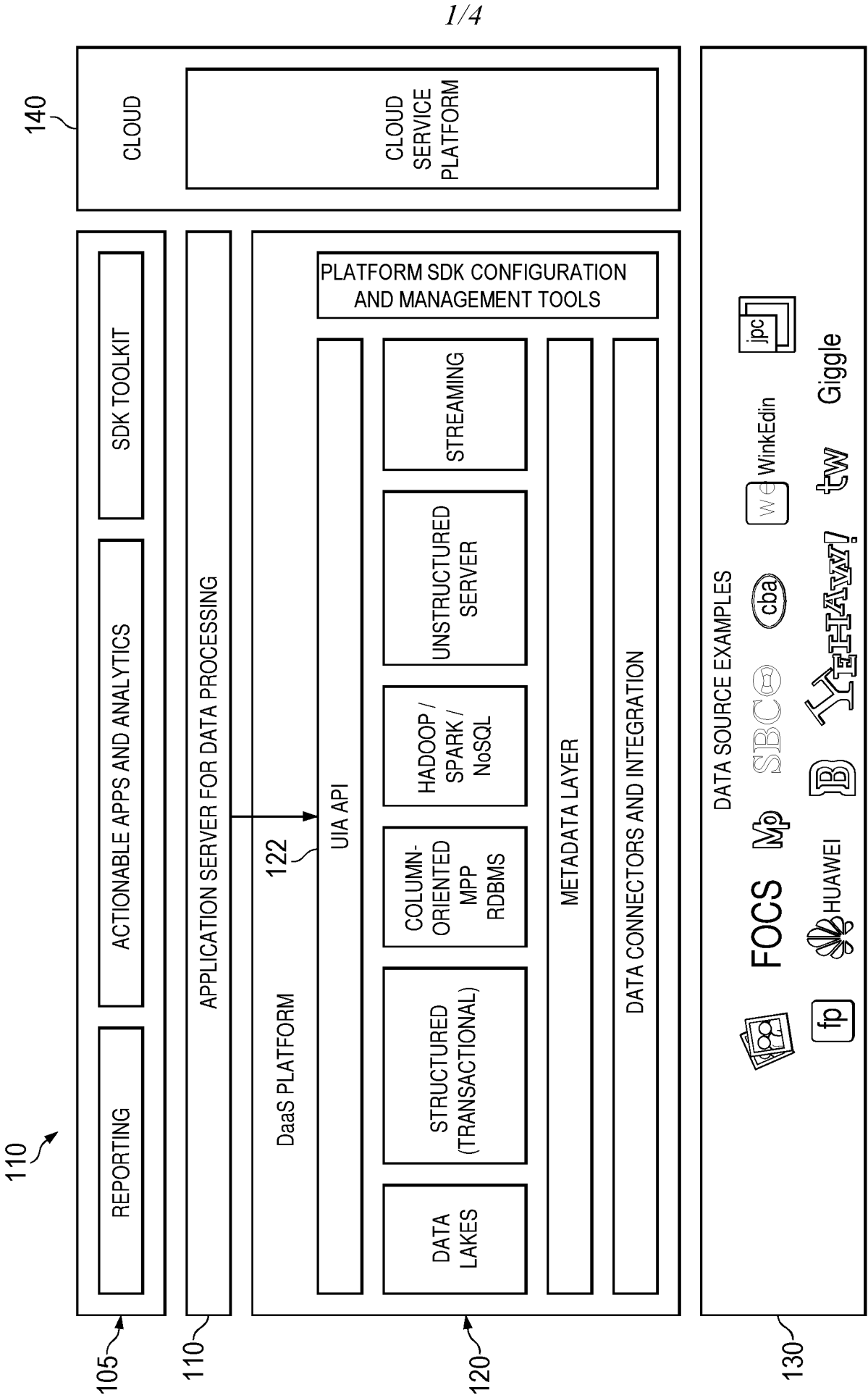


FIG. 1

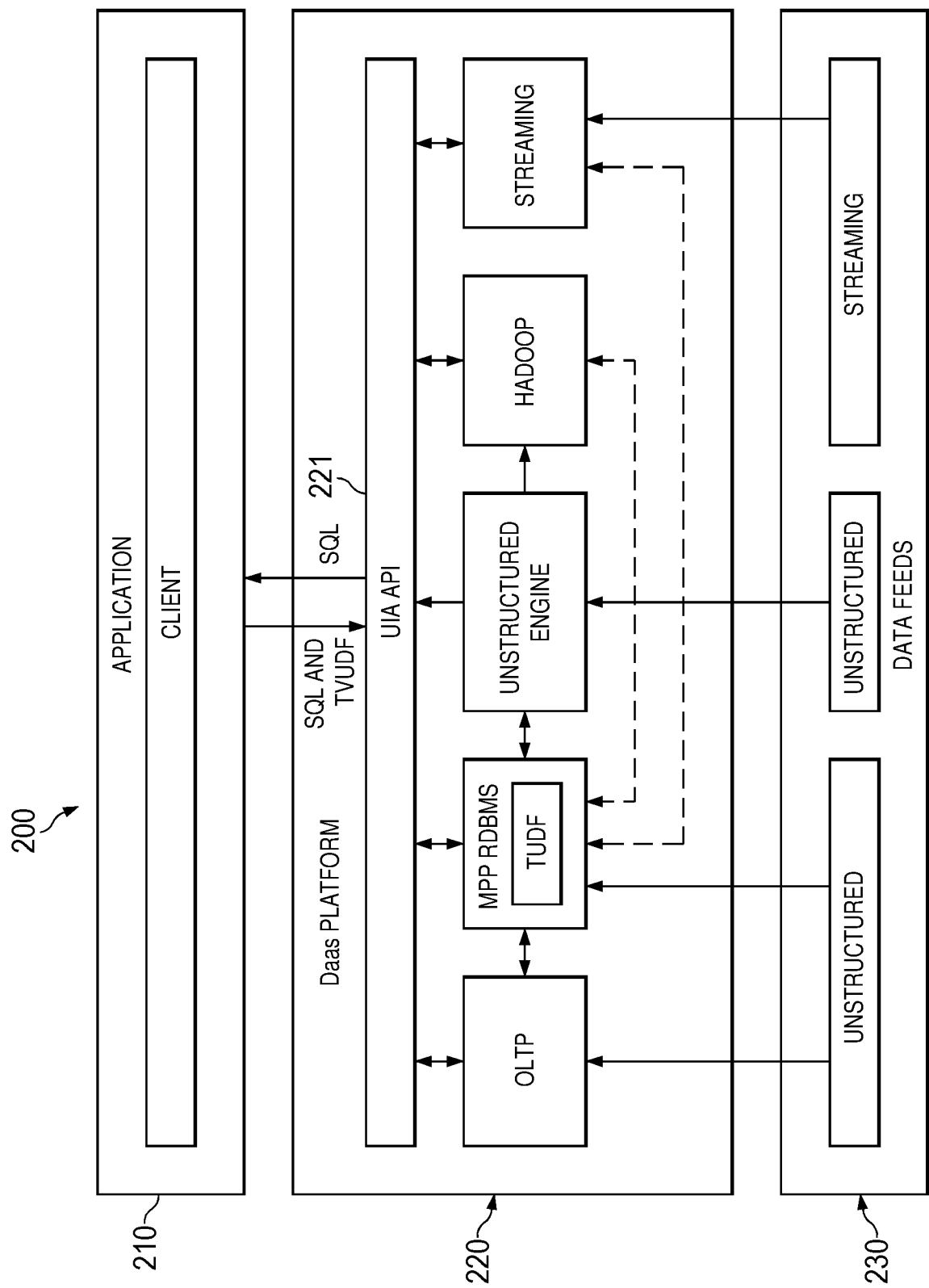


FIG. 2

3/4

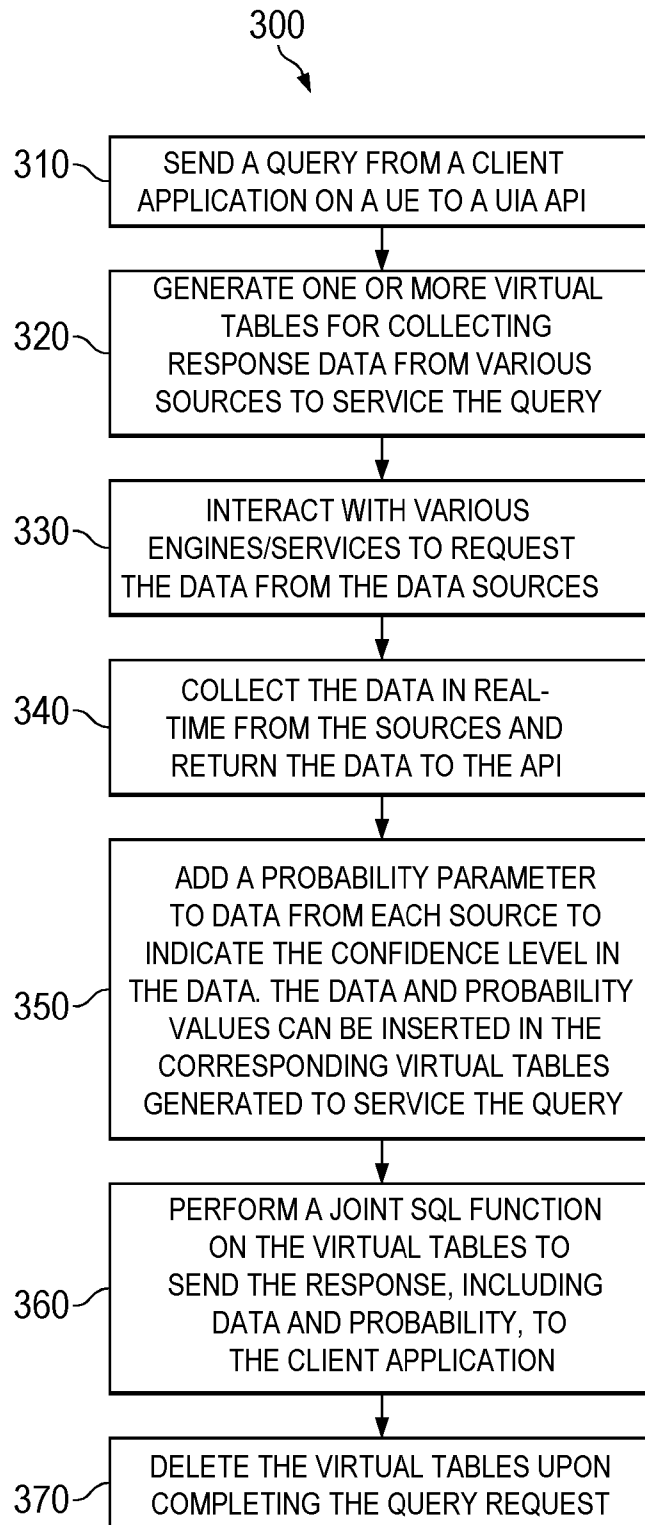


FIG. 3

4/4

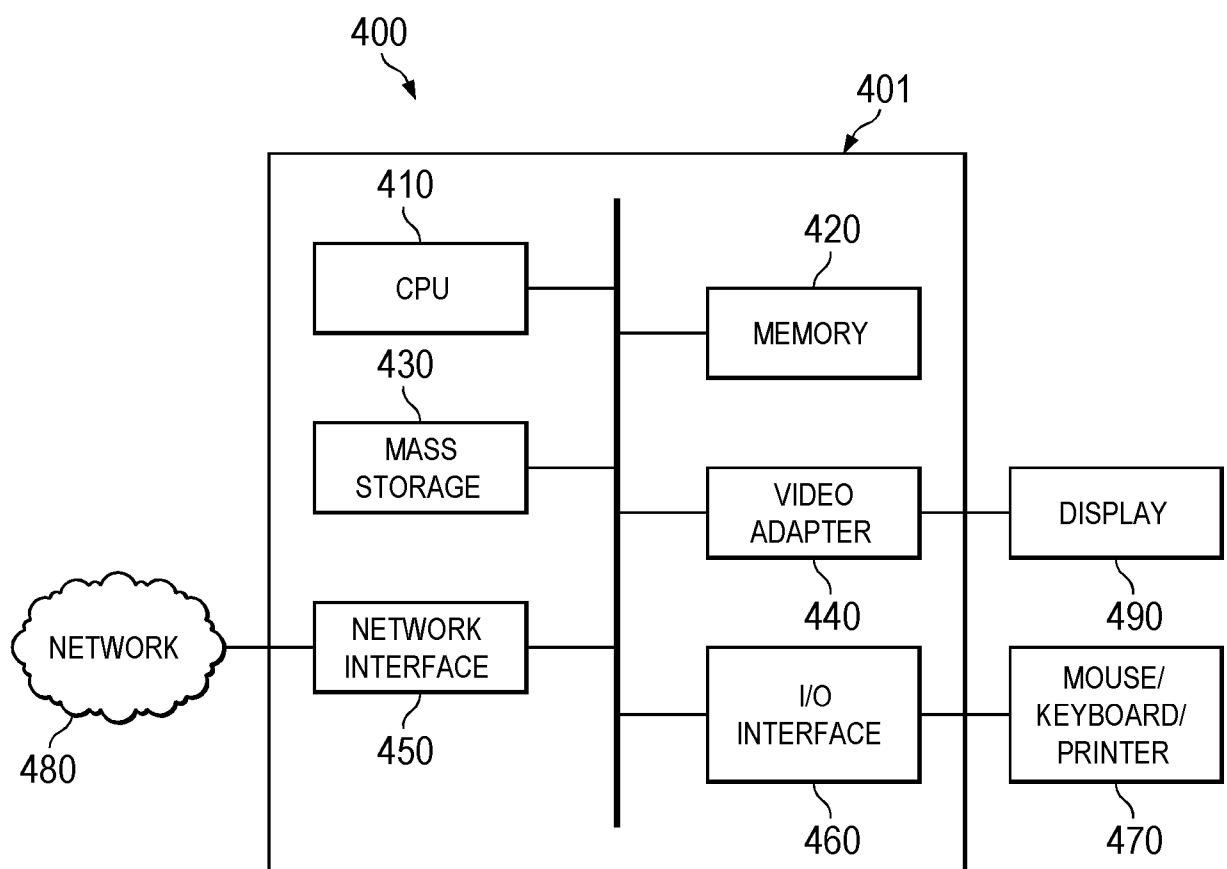


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/093092

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/40(2006.01)i

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; H04L

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)

CNPAT, CNKI, WPI, EPODOC: data, level, confidence, quality, reliability, data as a service, daas, real w time, join

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 102663021 A (ZHEJIANG PANSHI INFORMATION TECHNOLOGY CO., LTD) 12 September 2012 (2012-09-12) description, paragraphs 0038-0077	1-23
Y	CN 102801489 A (ZTE CORPORATION) 28 November 2012 (2012-11-28) claims 1-5	1-23
A	CN 104618324 A (BAIDU ONLINE NETWORK TECHNOLOGY BEIJING CO., LTD) 13 May 2015 (2015-05-13) the whole document	1-23
A	US 2013275585 A1 (INFOSYS LIMITED) 17 October 2013 (2013-10-17) the whole document	1-23
A	US 2014129707 A1 (AMAZON TECHNOLOGIES, INC.) 08 May 2014 (2014-05-08) the whole document	1-23

☐ 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

09 October 2016

Date of mailing of the international search report

26 October 2016

Name and mailing address of the ISA/CN

**STATE INTELLECTUAL PROPERTY OFFICE OF THE
P.R.CHINA
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China**

Facsimile No. (86-10)62019451

Authorized officer

JIN, Jing

Telephone No. (86-10)62413414

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2016/093092

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	102663021	A	12 September 2012	None			
CN	102801489	A	28 November 2012	WO	2012159473	A1	29 November 2012
CN	104618324	A	13 May 2015	None			
US	2013275585	A1	17 October 2013	WO	2012052998	A1	26 April 2012
				LU	92029	A1	21 October 2012
US	2014129707	A1	08 May 2014	US	2015263927	A1	17 September 2015
				US	2013054796	A1	28 February 2013
				US	2011302304	A1	08 December 2011
				US	8051166	B1	01 November 2011