Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3000734 A1 2010/09/16

(21) 3 000 734

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2010/02/24

(41) Mise à la disp. pub./Open to Public Insp.: 2010/09/16

(62) Demande originale/Original Application: 2 754 747

(30) Priorité/Priority: 2009/03/13 (US12/404,181)

(51) Cl.Int./Int.Cl. *G06Q 10/10* (2012.01),
G06F 17/00 (2006.01)

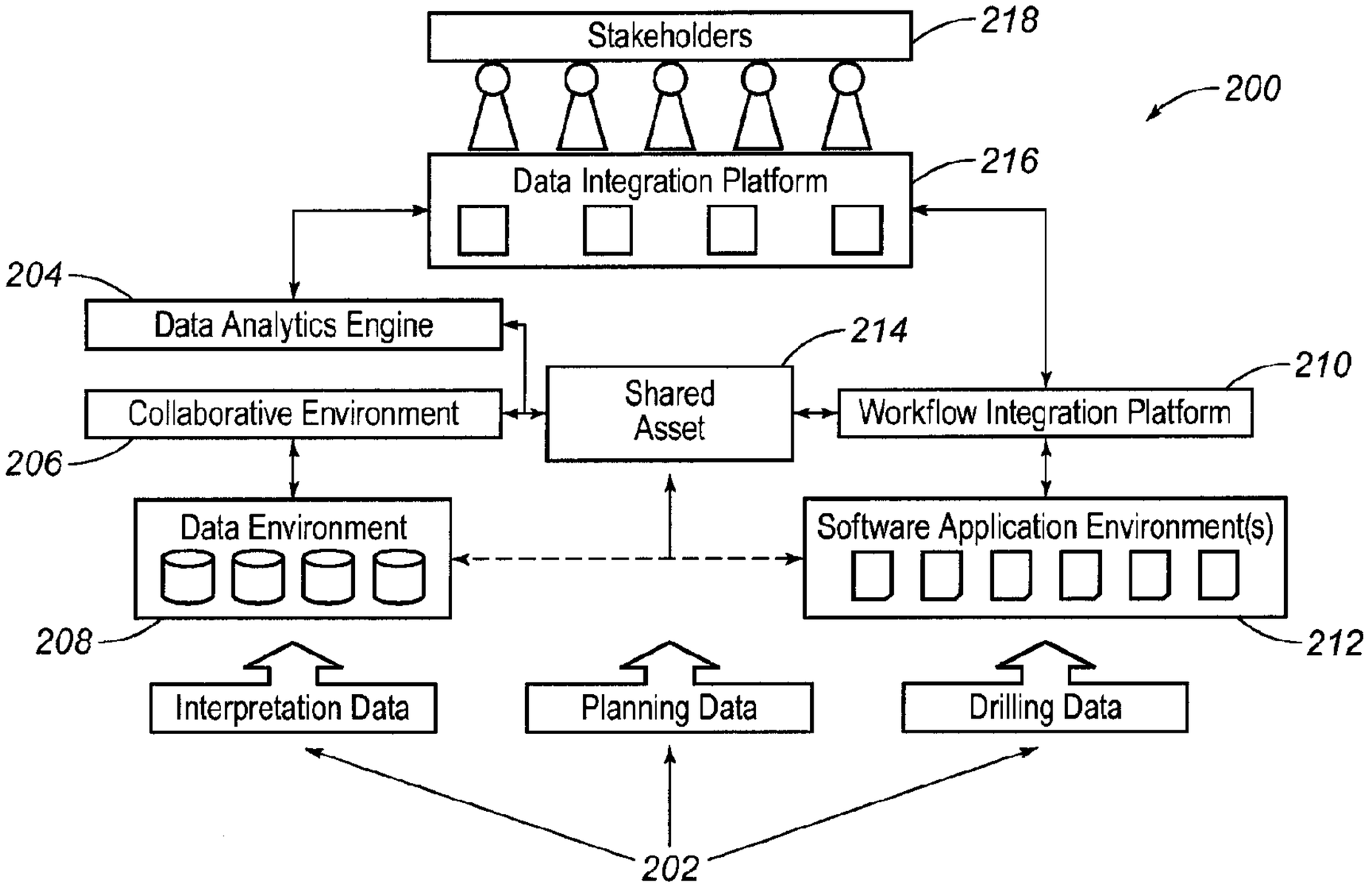
(71) Demandeur/Applicant:
LANDMARK GRAPHICS CORPORATION, US

(72) Inventeurs/Inventors:
PAULK, MARTIN D., US;
JOHNSON, DAVID, US

(74) Agent: PARLEE MCLAWS LLP

(54) Titre : SYSTEME ET METHODES DE GESTION DE DONNEES EN TEMPS REEL DANS UN ENVIRONNEMENT COLLABORATIF

(54) Title: SYSTEMS AND METHODS FOR REAL TIME DATA MANAGEMENT IN A COLLABORATIVE ENVIRONMENT



(57) Abrégé/Abstract:
Systems and methods for real time data management in a collaborative environment. In one embodiment, the systems and methods dynamically link stakeholders, through a collaborative environment, to current and updated data. The systems and methods also cross- reference data objects between stakeholders for calling data objects by a common name, and determine a preferred method for each operation related to processing data for a data object.

ABSTRACT

Systems and methods for real time data management in a collaborative environment. In one embodiment, the systems and methods dynamically link stakeholders, through a collaborative environment, to current and updated data. The systems and methods also cross-reference data objects between stakeholders for calling data objects by a common name, and determine a preferred method for each operation related to processing data for a data object.

SYSTEMS AND METHODS FOR REAL TIME DATA MANAGEMENT IN A COLLABORATIVE ENVIRONMENT

[0001] Not applicable.

FIELD OF THE INVENTION

[0002] The present invention generally relates to systems and methods for managing data, in real time, through a collaborative environment.

BACKGROUND OF THE INVENTION

[0003] Understanding and operating an oil and gas production asset as a single holistic system has been frustrated by significant impediments. For any asset, there are typically multiple applications, multiple data sets, multiple taxonomies and multiple users, also referred to herein as stakeholders, some or all of which may be sharing common data across the asset. Interoperability among these programs, persons, and structures as a single system, while desired, has been frustrated by the lack of an underlying framework for handling the necessary transformations, translations, and definitions required between and among the various system components.

[0004] Attempts to provide this understanding and operation have previously focused on providing data replication, where each stakeholder group develops or receives its own version of the logical network and data model that includes all of its requirements. In these attempts, the act of transforming the data model by correlating changes between the data model representations has not been done or has been done crudely. Although some level of interoperability has been achieved by point to point integration, it is largely limited to supporting single workflows. Moreover, changes to the data model representations cannot be effectively controlled when each stakeholder can decide whether such changes should be applied (accepted) and communicated to the other stakeholder's.

[0005] Other impediments to real time data management include, for example, the use of different calls or names for the same data object and different methods for the same operation. Each data object and each operation may represent a separate data model representation. Consequently, each stakeholder's ability to effectively access updates to a data object and

uniformly apply a preferred method to a particular operation are significantly impaired. A stakeholder, for example, may refer to the same data object by a different name thus, impairing the ability to communicate changes to the data object between different stakeholders. Each stakeholder may also utilize a different method for a particular operation when a preferred method is desired to maintain uniform operation results. Moreover, a stakeholder is often required to log onto a network to access a client server before downloading the desired data model representations or any changes made thereto. This sequence is typically required each time access to the data model is desired.

[0006] There is therefore, a need for systems and methods to provide uninterrupted interoperability among the various data sets, applications, taxonomies and stakeholders sharing data across a production asset. In other words, there is a need for dynamic linking between the stakeholders and the data model representations that allows the stakeholders to maintain access to each database and receive data model updates in real time, In addition, there is a need for a uniform system that calls objects within an object class by a common name and determines a preferred method for each operation within a collaborative environment.

SUMMARY OF THE INVENTION

[0007] The present invention therefore, meets the above needs and overcomes one or more deficiencies in the prior art by providing systems and methods for managing and processing data in a collaborative environment.

[0007a] Certain exemplary embodiments can provide a computer-implemented method for managing data in a collaborative environment, comprising: creating a system name on a data-integration platform to identify an object class; selecting an object from the object class; linking the object and the system name on the data-integration platform using a computer processor; identifying each unique name in the collaborative environment used by a respective plurality of stakeholders to identify the object; and linking each unique name for the object and the system name on the data-integration platform.

[0007b] Certain exemplary embodiments can provide a non-transitory program carrier device carrying compute executable instructions for managing data in a collaborative environment, the instructions being executable to implement: creating a system name on a data-integration platform to identify an object class; selecting an object from the object class; linking the object and the system name on the data-integration platform; identifying each unique name in the collaborative environment used by a respective plurality of stakeholders to identify the object; and linking each unique name for the object and the system name on the data-integration platform

[0008] In one embodiment the present invention includes a computer implemented method for managing data in a collaborative environment, which comprises: i) creating a system name on a data-integration platform to identify an object class; ii) selecting an object from the object class using a unique name for the object; iii) linking the object and the system name on the data-integration platform; iv) identifying each unique name for the object in the collaborative environment; and v) linking each unique name for the object and the system name on the data-integration platform.

[0009] In another embodiment, the present invention includes a program carrier device for carrying computer executable instructions for managing data in a collaborative environment. The instructions are executable to implement: i) creating a system name to identify an object class; ii) selecting an object from the object class using a unique name for the object; iii) linking the object and the system name; iv) identifying each unique name for the object in the collaborative environment; and v) linking each unique name for the object and the system name.

[0010] In yet another embodiment, the present invention includes a computer implemented method for processing data in a collaborative environment, which comprises: i) indexing multiple methods in a shared asset according to an intended operation for each method; ii) identifying a preferred method in the shared asset for each intended operation; iii) selecting an operation; and iv) processing data using the preferred method for the selected operation.

[0011] In yet another embodiment, the present invention includes a program carrier device for carrying computer executable instructions for processing data in a collaborative environment. The instructions are executable to implement: i) indexing multiple methods according to an intended operation for each method; ii) identifying a preferred method for each intended operation; iii) selecting an operation; and iv) processing data using the preferred method for the selected operation.

[0012] Additional aspects, advantages and embodiments of the invention will become apparent to those skilled in the art from the following description of the various embodiments and related drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The present invention is described below with references to the accompanying drawings in which like elements are referenced with like reference numerals, and in which:

[0014] **FIG. 1** is a block diagram illustrating one embodiment of a system for implementing the present invention.

[0015] **FIG. 2** is a block diagram illustrating one embodiment of a system architecture for implementing the present invention.

[0016] **FIG. 3A** is a flow diagram illustrating one embodiment of a method for implementing the present invention.

[0017] **FIG. 3B** is a flow diagram illustrating the continuation of the method in **FIG. 3A**.

[0018] **FIG. 4** is a flow diagram illustrating another embodiment of a method for implementing the present invention.

[0019] **FIG. 5** is a flow diagram illustrating a standard workflow incorporating the present invention.

[0020] **FIGS. 6A-6F** illustrate enabling networking environments for implementing the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0021] The subject matter of the present invention is described with specificity, however, the description itself is not intended to limit the scope of the invention. The subject matter thus, might also be embodied in other ways, to include different steps or combinations of steps similar to the ones described herein, in conjunction with other present or future technologies. Moreover, although the term "step" may be used herein to describe different elements of methods employed, the term should not be interpreted as implying any particular order among or between various steps herein disclosed unless otherwise expressly limited by the description to a particular order. While the following description refers to the oil and gas industry, the systems and methods of the present invention are not limited thereto and may also be applied to other industries to achieve similar results.

System Description

[0022] The present invention may be implemented through a computer-executable program of instructions, such as program modules, generally referred to as software applications or application programs executed by a computer. The software may include, for example, routines, programs, objects, components, and data structures that perform particular tasks or implement particular abstract data types. The software forms an interface to allow a computer to react according to a source of input. DecisionSpace®, which is a commercial software application marketed by Landmark Graphics Corporation, may be used as an interface application to implement the present invention. The software may also cooperate with other code segments to initiate a variety of tasks in response to data received in conjunction with the source of the received data. The software may be stored and/or carried on any variety of memory media such as CD-ROM, magnetic disk, bubble memory and

semiconductor memory (*e.g.*, various types of RAM or ROM). Furthermore, the software and its results may be transmitted over a variety of carrier media such as optical fiber, metallic wire, free space and/or through any of a variety of networks such as the Internet.

[0023] Moreover, those skilled in the art will appreciate that the invention may be practiced with a variety of computer-system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable-consumer electronics, minicomputers, mainframe computers, and the like. Any number of computer-systems and computer networks are acceptable for use with the present invention. The invention may be practiced in distributed-computing environments where tasks are performed by remote-processing devices that are linked through a communications network. In a distributed-computing environment, program modules may be located in both local and remote computer-storage media including memory storage devices. The present invention may therefore, be implemented in connection with various hardware, software or a combination thereof, in a computer system or other processing system.

[0024] Referring now to **FIG. 1**, a block diagram of one embodiment of a system for implementing the present invention on a computer is illustrated. The system includes a computing unit, sometimes referred to as a computing system, which contains memory, application programs, a client interface, and a processing unit. The computing unit is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention.

[0025] The memory primarily stores the application programs, which may also be described as program modules containing computer-executable instructions, executed by the computing unit for implementing the present invention described herein and illustrated in **FIGS. 2-5**. The memory therefore, includes a XREF Module, which enables the method illustrated and described in reference to **FIGS. 3A** and **3B**. The memory also includes a

Shared Asset Module, which enables the method illustrated and described in reference to **FIG. 4**. The Shared Asset Module includes a shared asset, also known as a system model, which is illustrated and described in reference to **FIG. 2**. The Shared Asset Model may also be used to enable the methods described. The systems and methods of the present invention therefore, may be based on a system model that correlates meta-data-model representations of a production asset with the various asset-logic model representations (or taxonomies), in real time, to provide a collaborative, consistent, and relevant representation of the asset to the different stakeholders. The system model may perform the correlations between different meta-data model representations using the various asset-logic model representations for common data items and attributes shared across the system thus, addressing the many key activities relating to the asset that may have different meta-model representations representing their pertinent data requirements. In this manner, effective interoperability may be achieved. The XREF Module and Shared Asset Module also interact with DecisionSpace®, EDM™, StrataSteer, INSITE™, WITSML™ and BPM™ as further described in reference to **FIG. 2**.

[0026] Although the computing unit is shown as having a generalized memory, the computing unit typically includes a variety of computer readable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. The computing system memory may include computer storage media in the form of volatile and/or nonvolatile memory such as a read only memory (ROM) and random access memory (RAM). A basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within the computing unit, such as during start-up, is typically stored in ROM. The RAM typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by

the processing unit. By way of example, and not limitation, the computing unit includes an operating system, application programs, other program modules, and program data.

[0027] The components shown in the memory may also be included in other removable/nonremovable, volatile/nonvolatile computer storage media. For example only, a hard disk drive may read from or write to nonremovable, nonvolatile magnetic media, a magnetic disk drive may read from or write to a removable, non-volatile magnetic disk, and an optical disk drive may read from or write to a removable, nonvolatile optical disk such as a CD ROM or other optical media. Other removable/non-removable, volatile/non-volatile computer storage media that can be used in the exemplary operating environment may include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The drives and their associated computer storage media discussed above therefore, store and/or carry computer readable instructions, data structures, program modules and other data for the computing unit.

[0028] A stakeholder may enter commands and information into the computing unit through the client interface, which may be input devices such as a keyboard and pointing device, commonly referred to as a mouse, trackball or touch pad. Input devices may include a microphone, joystick, satellite dish, scanner, or the like.

[0029] These and other input devices are often connected to the processing unit through the client interface that is coupled to a system bus, but may be connected by other interface and bus structures, such as a parallel port or a universal serial bus (USB). A monitor or other type of display device may be connected to the system bus via an interface, such as a video interface. In addition to the monitor, computers may also include other peripheral output devices such as speakers and printer, which may be connected through an output peripheral interface.

[0030] Although many other internal components of the computing unit are not shown, those of ordinary skill in the art will appreciate that such components and their interconnection are well known.

[0031] Referring now to **FIG. 2**, a block diagram of one embodiment of a system architecture **200** for implementing the present invention is illustrated. The shared asset **214** is illustrated within the wider system architecture **200** for collaboration among multiple stakeholders **218**. Additional system components typically associated with a shared asset may include, for example, different types of data **202** (*e.g.* interpretation data, planning data and drilling data), data analytics engine **204**, data-integration platform **216**, data environment **208**, workflow integration platform **210**, software application environment(s) **212** and collaborative environment **206**. The various components of the system architecture **200** enable specific inter-related functions and therefore, must interact with each other seamlessly in order to avoid discontinuities and/or errors. Other components of the system architecture **200** may include interfaces to and from an Enterprise Resource Planning System (e.g. SAP) or a computerized Maintenance Management System, or a computerized Work Management System controlling work in the physical asset and necessary status of equipment and system isolations.

[0032] The data analytics engine **204** allows data to be manipulated to produce numerous reports and analysis using simple to complex proprietary and industry standard algorithms. It typically supports data filtering and cleaning as well as appropriate visualization technologies to present data in meaningful, relevant and insightful ways into the viewing or collaborative environment **206** for use by the stakeholders **218**.

[0033] The collaborative environment **206** typically represents a web-based (intranet or internet) access and viewing environment that can be used by a diverse range of potential geographically located stakeholders **218**, which will provide secure access and bandwidths

appropriate to the range of data and software applications included in the system architecture **200**. The use of portals will typically allow diverse stakeholders **218** to have relevant custom views of only the data they require. The collaborative environment **206** therefore, cooperates with the workflow integration platform **210**, the shared asset **214** and the data-integration platform **216** to present data to the stakeholders, as needed, in real time based on the operations conducted in the data analytics engine **204**. Thus, cooperation among the foregoing described components for the system architecture **200** enable a data environment capable of changing in real time and/or an applications environment capable of selecting and using preferred methods in real time.

[0034] The data environment **208** typically interfaces with some regulatory or distributed control system incorporating data to and from instruments, meters, and control devices, such as actuators, valves and the like, across the physical asset. Considering the typical functions of each component, the data environment **208** typically includes multiple types of data **202**, each designed to best suit particular types of data. For example, a relational database for interpretation data, a data historian for high frequency planning data and a data warehouse for drilling data such as well tables and production allocation tables. The data will exist in one or more data languages, such as, for example, OPC, ODBC, HTML, WITSML, and PRODML, which are well known in the art and conform to respective corresponding data standards.

[0035] The workflow integration and orchestration platform **210** spans the various IT operating environments, connects to the numerous software applications, maps to the respective data tags and items used by the software applications and interfaces with some viewing or shared collaborative environments to interface and interact with numerous stakeholders **218** involved across the workflows. The workflows are numerous but might include frequent and infrequent activities such as, for example, well test validation,

production allocation, production surveillance, production optimization and others.

[0036] The software application environment(s) **212** includes different IT operating environments such as Windows and Linux, appropriate to the range of applications used across the asset. The software application environment(s) **212** may also include well known, proprietary applications for providing diverse, complex, and relatively simple functionality. These applications, for example, may include petro-technical and geoscience applications common to the industry as well as more common software applications commonly used such as MS Office. These applications may also be used for, but not limited to, reservoir and well simulation, modeling of hydraulic networks and facilities systems, material balance, nodal analysis, production allocation, production surveillance, hydrocarbon accounting, regulatory reporting, and economics. These applications may be proprietary or customized and will invariably incorporate and use a multitude of algorithms, units, and annotations. These applications will also manipulate and transform existing data and generate further data in ways unique to the applications.

[0037] The shared asset **214** provides interoperability across the various diverse applications and datasets within the overall system architecture **200**. In other words, the shared asset **214** manages the numerous interfaces and interactions between the data items and their correct and consistent usage within the various logic schemes that are representative of the system architecture **200**. It does so in a dynamic operating environment where all or many components may be used simultaneously and/or continuously by the stakeholders **218**, each with specific and varying access and authority rights. The shared asset **214** therefore, may enable the method illustrated and described in reference to **FIG. 4** and the methods described in reference to FIG. 4.

[0038] The data-integration platform **216** typically provides data connectors appropriate to each data language and database. This component allows the stakeholders **218** to map to particular data sets such as asset equipment tags or unique well identifiers, thus enabling data flow throughout the system architecture **200**. The data-integration platform **216** therefore, may enable the method illustrated and described in reference to **FIGS. 3A** and **3B**.

[0039] DecisionSpace® for Production may be used to provide the system architecture **200**, component integration and set of functionality illustrated in **FIG. 2**. The XREF Module provides the data-integration platform **216**. WITSML™ provides the collaborative environment **206** for dynamically linking data organization and visualization within the overall system architecture **200**. DecisionSpace®, EDM™, StrataSteer™ and INSITE™ provide the technical applications for the software application environment(s) **212**. The data environment **208** may include numerous databases that correspond with applications in the software application environment(s) **212**. BPM™ provides the workflow integration platform **210**, which enables orchestration or technical workflows that utilize technical applications and data from the related data **202** and software application environment(s) **212**. DecisionSpace®, StrataSteer™ and EDM™ may also be used as the data analytics engine **204**, which is capable of handling both high volume and real time data about operations and aggregate historical data used for engineering purposes. The Shared Asset Module provides the shared asset **214**, which includes the system model. Each of the components jointly and separately benefit from the functionality provided by the shared asset **214**.

[0040] The system architecture **200** integrates data entry/access by various stakeholders **218** over one or more servers and makes new data available for the system model, in real time, to all stakeholders **218** without the need for updated data requests from each stakeholder **218**. The system architecture **200** insures that all stakeholders **218** are

dynamically linked to current data – meaning that each database **202** is constantly accessible, in real time, to all stakeholders **218** without having to perform the traditional procedures to access a particular database. In other words, the system model is updated, in real time, each time new data enters the system model through a client, which is accessible by the stakeholders **218** through the dynamic link.

[0041] Referring now to **FIGS. 6A-6F**, various optional networking environments are illustrated in accordance with multiple embodiments of the invention. The computers illustrated in **FIGS. 6A-6F** may include desktop computers, laptop computers, dedicated servers, supercomputers, personal digital assistants (PDA's), other well known computing devices, or any combination thereof.

[0042] **FIG. 6A** illustrates a local environment. The local environment may include locally running a server/client **600**. The server/client **600** may process data according to the methods described herein.

[0043] **FIG. 6B** illustrates a collaboration hub networking environment. A collaboration hub may include any computer that transmits data to and receives data from multiple other computers. A collaboration hub may also be configured to incorporate changes received from multiple other computers into a single data object or other data instance. A collaboration hub may also control application security settings for one or more other computers. The collaboration hub networking environment may include a server/collaboration hub **602**, which may process data according to the methods described herein, as well as functioning as a collaboration hub. The collaboration hub networking environment may also include one or more collaboration clients **604, 606** that remotely access data, which is processed on the server/collaboration hub **602**. In one embodiment, the collaboration clients **604, 606** may also alter or manipulate the processed data. These changes may be tracked, processed, or stored by the server/collaboration hub **602**.

[0044] **FIG. 6C** illustrates a remote execution networking environment. The remote execution networking environment may include a server **608**, which processes data according to the methods described herein. The remote execution networking environment may also include one or more clients **610** that remotely access data processed on the server **608**.

[0045] **FIG. 6D** illustrates a remote execution and collaboration networking environment. The remote execution and collaboration networking environment may include a server **612**, which visibly or invisibly processes data according to the methods described herein. The remote execution and collaboration networking environment may also include one or more client/collaboration hubs **614**. Each client/collaboration hub **614** may remotely access data that is processed on the server **612**. Each client/collaboration hub **614** may transmit data to, and receive data from, one or more collaboration clients **616, 618**. Each client/collaboration hub **614** may also be configured to incorporate changes received from collaboration clients **616, 618** into a single data object or other data instance. The client/collaboration hub **614** may also control security settings for collaboration clients **616, 618**. The collaboration clients **616, 618** may remotely access data that is processed on the server **612** and transmitted via the client/collaboration hub **614**. In one embodiment, the collaboration clients **616, 618** may also alter or manipulate the processed data. These changes may be tracked, processed, or stored by each client/collaboration hub **614** through window and display state changes that are sent to the server **612** for processing and synchronization of all client interactions.

[0046] **FIG. 6E** illustrates an application service provider (ASP) networking environment. The ASP networking environment may include one or more servers **620, 622, 624**, which process data according to the methods described herein. The ASP networking environment may also include one or more client/ASP servers **626**. Each client/ASP server **626** may receive data that is processed on the servers **620, 622, 624**. Each client/ASP server

626 may include a computer that is configured to manage and distribute software-based services and solutions to customers across a wide area or other network from a central data center. In one embodiment, each client/ASP server **626** may be a third-party server that is owned or operated by an entity separate from the entity owning and operating the servers **620, 622, 624** or the client **628**. Each client/ASP server **626** may be an enterprise ASP, which is designed to deliver high-end business applications; a local or regional ASP, which is designed to supply a wide variety of application services for smaller businesses in a local area; a specialist ASP, which is designed to provide applications for a specific need; a vertical market ASP, which is designed to provide support to a specific industry; or a volume business ASP, which is designed to supply small or medium-sized businesses with prepackaged application services in volume. The ASP networking environment may also include one or more clients **628** that remotely access data processed on the servers **620, 622, 624** and transmit the data to each client **628** via the client/ASP server **626**.

[0047] **FIG. 6F** illustrates an ASP and collaboration networking environment. The ASP and collaboration networking environment may include one or more servers **630, 632, 634**, which process data according to the methods described herein. The ASP and collaboration networking environment may also include one or more client/ASP servers **636**. Each client/ASP server **636** may receive data that is processed on the servers **630, 632, 634**. Each client/ASP server **636** may include a computer that is configured to manage and distribute software-based services and solutions to customers across a wide area or other network from a central data center. In one embodiment, each client/ASP server **636** may include a third-party server that is owned or operated by an entity separate from the entity owning and operating the servers **630, 632, 634**, the client/collaboration hub **638**, or the client **640, 642**. Each client/ASP server **636** may be an enterprise ASP, which is designed to deliver high-end business applications; a local or regional ASP, which is designed to supply a

wide variety of application services for smaller businesses in a local area; a specialist ASP, which is designed to provide applications for a specific need; a vertical market ASP, which is designed to provide support to a specific industry; or a volume business ASP, which is designed to supply small or medium-sized businesses with prepackaged application services in volume.

[0048] The ASP and collaboration networking environment may also include one or more client/collaboration hubs **638**. Each client/collaboration hub **638** may remotely access data that is processed on the servers **630, 632, 634** and is transmitted via the client/ASP server **636**. Each client/collaboration hub **638** may transmit data to and receive data from multiple clients **640, 642.** The client/collaboration hub **638** may be configured to incorporate changes received from clients **640, 642** into a single data object or other data instance. Each client/collaboration hub **638** may also control security settings for the clients **640, 642.** The clients **640, 642** may also remotely access data that is processed on the servers **630, 632, 634,** and transmitted via the client/ASP server **636** and the client/collaboration hub **638.** In another embodiment, the collaboration clients **640, 642** may also alter or manipulate the rendered data through window and display state changes that are sent to the servers **630, 632, 634** for processing and synchronization of all client interactions.

Method Description

[0049] Referring now to **FIG. 3A**, a flow diagram illustrates one embodiment of a method **300** for implementing the present invention.

[0050] In step **302,** a database is defined using a GUI or other means well known in the art. A database, for example, may be defined for interpretation data, planning data or drilling data relating to particular wells in an oil/gas field. Other databases may be defined for the same or different data relating to other wells in the same oil/gas field or other oil/gas fields. A database may be defined by creating it and populating it with input data, or a

database may be defined by identifying a preexisting database. A preexisting database therefore, may be proprietary or owned by a third party.

[0051] In step **304**, a server is identified using a GUI or other means well known in the art. The server may be identified by its URL and password.

[0052] In step **306**, the method **300** determines whether the server identified in step **304** is working. If the server is working, then the method **300** continues to step **308**. If the server is not working, then the method **300** returns to step **304**.

[0053] In step **308**, the method **300** determines whether to identify a client by using a GUI or other means well known in the art to prompt for this request. If a client needs to be identified, then the method **300** continues to step **310**. If a client does not need to be identified, then the method **300** continues to step **314**.

[0054] In step **310**, a client is identified using a GUI or other means well known in the art. The client may be identified by its URL and password.

[0055] In step **312**, the method **300** determines if the client identified in step **310** is working. If the client is working, then the method **300** continues to step **314**. If the client is not working, then the method **300** returns to step **310**.

[0056] In step **314**, the method **300** determines if more databases need to be defined by using a GUI or other means well known in the art to prompt for this request. If more databases need to be defined, then the method **300** returns to step **302**. If more databases do not need to be defined, then the method **300** continues to step **316**.

[0057] In step **316**, a data-integration platform is populated with each server and/or client identified in steps **304** and **310**, respectively, and the location of each database defined in step **302** for a respective server and/or client. If only a server is identified, then the data-integration platform is limited to only reading from the database(s) defined in step **302**. If,

however, a server and a client are identified, then the data-integration platform is capable of reading and writing from/to the database(s) defined in step **302**.

[0058] In step **318**, a data type (object class) is selected from the database(s) that were populated in step **316** using a GUI or other means well known in the art. An object class, for example, may include wells for a particular production field.

[0059] In step **320**, a name is created to identify the object class selected in step **318** using a GUI or other means well known in the art. The name, for example, may be "Field 1 Wells."

[0060] In step **322**, an object is selected from the object class and is linked to the name created in step **320** using a GUI or other means well known in the art. Thus, for example, the object class name may be "Field 1 Wells" and the object selected from this class may be a particular well. The well is therefore, linked to the name Field 1 Wells.

[0061] In step **324**, each unique name used by different respective stakeholders for calling the selected object is identified and linked to the name created for the object class in step **320**. Each unique name for calling the selected object may be identified by comparing data profiles or data attributes for the selected object and one or more related objects, each controlled by a respective stakeholder, to determine whether the data profile or the data attribute for the selected object and the data profile or the data attribute for the one or more related objects substantially identify the same object. If each unique name substantially identifies the same object, then each unique name may be identified as being used for calling the selected object. The results of this step are stored in the data-integration platform, as a cross-reference table, which includes each server and/or client identified in steps **304** and **310**, respectively. The data-integration platform therefore, may access interpretation data for wellbores (object class) associated with a particular well, which include interpretation data in the form of well logs, trajectories and reports (objects) for each wellbore. In the same

manner, the data-integration platform may access interpretation data for wells (object class) associated with a particular oil/gas field, which may include wells and wellbores, surfaces and faults around each well (objects).

[0062] In step **326**, the method **300** determines whether to select another object by using a GUI or other means well known in the art to prompt for this request. If another object needs to be selected from the object class, then the method **300** returns to step **322**. If another object from the object class does not need to be selected, then the method **300** continues to step **328**.

[0063] In step **328**, the method **300** determines whether to repeat. If the method **300** is repeated, then the method **300** returns to step **318**. The method **300** may be repeated for the same, or different, object classes and the same, or different, objects. If the method **300** does not repeat, then the method **300** ends.

[0064] Steps **322-328** may be performed automatically for each object in each object class in a computer implemented method **300** until all objects have been selected (exhausted) for each object class. Alternatively, steps **322-328** may be selectively performed for each object and each object class using a GUI or other means well known in the art.

[0065] The method **300** therefore, permits all objects (*e.g.* wells) from a named object class (*e.g.* Field 1 Wells) to be called (accessed) through a common system name, regardless of what unique name each stakeholder uses to identify a particular object. This process, for example, creates a cross-reference table in the data-integration platform comprising one or more objects, which permits a stakeholder to modify data related to an object identified by a unique name and another stakeholder to call the modified data related to the object using another unique name. If, for example, a stakeholder refers to a Well XYZ and another stakeholder refers to the same well as Well ABC, then the method **300** calls that well by a common system name created to identify the object class in step **320**. Thus, when a

stakeholder modifies, or adds to, the data for Well XYZ, another stakeholder may call that data by accessing (calling) Well ABC. In other words, when a stakeholder describes an object with a unique name and another stakeholder describes the same object with another unique name, the method **300** permits the object to be identified to either stakeholder using each stakeholder's respective unique name. Each stakeholder therefore, may access all objects within an object class, and modified data related thereto, using a respective unique name.

[0066] Referring now to **FIG. 4**, a flow diagram illustrates another embodiment of a method for implementing the present invention.

[0067] In step **402**, a method is added to a shared asset using a GUI or other means well known in the art. The method may include, for example, subroutines that contain calculations or re-usable procedures. The shared asset, which includes the methods, also may include data related to the shared asset.

[0068] In step **404**, the methods added to the shared asset in step **402** are indexed by the shared asset according to operation. For example, interpretation calculations may be indexed to separate them from drilling calculations. In addition, various interpretation calculations may be indexed according to specific constraints. In other words, the methods may be indexed (organized) according to their intended use (operation) and their constraints.

[0069] In step **406**, a preferred method is identified for each operation using the index (intended use and constraints) for each method. Optionally, additional predetermined criteria may be used to identify a preferred method.

[0070] In step **408**, an operation is selected using a GUI or other means well known in the art.

[0071] In step **410**, the preferred method for the operation selected in step **408** is accessed from the shared asset. Applications may be built (or modified if they already exist) so that they can access a preferred method from the shared asset for a selected operation.

[0072] In step **412**, data is processed using the preferred method for the selected operation. Applications have the ability to execute the preferred method from the shared asset, which means that the calculations may be done on a computer controlled by the shared asset. Alternatively, applications may request a dynamically-bound instance of the preferred method physically compiled within the application for processing the data using the preferred method.

[0073] In step **414**, the method **400** determines whether to add a method by using a GUI or other means well known in the art to prompt for this request. If the method **400** determines that another method needs to be added, then the method **400** returns to step **402**. If the method **400** determines that another method does not need to be added, then the method **400** ends. As alternative methods are developed, they may be added to the shared asset in the manner thus described.

[0074] The method **400** therefore, provides an index of preferred methods for selected operations, which may be accessed through a collaborative environment from a shared asset (system model) to insure uniform operation results.

[0075] Referring now to **FIG. 5**, a flow diagram illustrates a standard workflow **500** incorporating the present invention.

[0076] In step **502**, an environment is defined, which may include the type of data and methods disclosed herein .

[0077] In step **504**, a subset is selected from the environment. This step simply selects a preferred subset of the data and/or methods defined for the environment in step **502** for processing. For example, preferred databases and data objects relevant to a particular

geographical region may be selected as a subset of a larger geographical environment defined in step **502**. This subset is declared the data for a job.

[0078] In step **506**, a job is defined (set up) by identifying i) various phases such as, for example, interpretation, design, modeling, drilling, and updating; ii) the methods/algorithms to process each data subset; iii) a list of rules that will define the stakeholders that will work with each data subset; and iv) the order (or workflow) that will be followed in the integrated process.

[0079] In step **508**, the job is executed for each phase defined in step **506**. For example, a job may be executed in a particular order according to the phases defined in step **506**. In addition, the methods described herein and the methods described in the Application are performed. For example, the method described in reference to **FIG. 4** and the methods described in the Application may yield a different preferred method and controlling stakeholder, respectively, depending upon which phase is currently being executed or performed. In other words, a selected operation may have a preferred method during the interpretation phase, which is different than the preferred method for the same operation during the design phase. Likewise, data controlled by a particular stakeholder during the interpretation phase may be controlled by a different stakeholder during the design phase. Results of this step may include data being updated, changed or created. Because the workflow **500** utilizes an integrated platform, the data is broadcast according to the rules described in reference to **FIG. 3B**.

[0080] In step **510**, the results of the job are reported, archived and finalized before the method **500** ends.

[0081] While the present invention has been described in connection with presently preferred embodiments, it will be understood by those skilled in the art that it is not intended to limit the invention to those embodiments. It is therefore, contemplated that various

alternative embodiments and modifications may be made to the disclosed embodiments without departing from the spirit and scope of the invention defined by the appended claims and equivalents thereof.

CLAIMS:

1. A computer implemented method for processing data in a collaborative environment, comprising:

indexing multiple methods in a shared asset according to an intended operation for each method;

identifying a preferred method in the shared asset for each intended operation;

selecting an operation; and

processing data using the preferred method for the selected operation.

2. The method of claim 1, wherein the intended operation for each method is the same.

3. The method of claim 2, wherein the preferred method is identified using a constraint for each method.

4. The method of claim 1, wherein the intended operation for each method is not the same.

5. The method of claim 4, wherein the preferred method is identified using a predetermined criteria for the multiple methods.

6. The method of claim 1, wherein the data is processed by an application in the shared asset using the preferred method for the selected operation.

7. The method of claim 1, wherein the data is processed by an application using a dynamically-bound instance of the preferred method for the selected operation in the application.

8. The method of claim 1, further comprising:

compiling additional methods in the shared asset; and

repeating the steps of claim 1.

9. The method of claim 1, wherein the multiple methods comprise calculations and subroutines.

10. The method of claim 9, wherein the calculations comprise interpretation calculations and drilling calculations.

11. A program carrier device for carrying computer executable instructions for processing data in a collaborative environment, the instructions being executable to implement:

indexing multiple methods according to an intended operation for each method;

identifying a preferred method for each intended operation;

selecting an operation; and

processing data using the preferred method for the selected operation.

12. The program carrier device of claim 11, wherein the intended operation for each method is the same.

13. The program carrier device of claim 12, wherein the preferred method is identified using a constraint for each method.

14. The program carrier device of claim 11, wherein the intended operation for each method is not the same.

15. The program carrier device of claim 14, wherein the preferred method is identified using a predetermined criteria for the multiple methods.

16. The program carrier device of claim 11, wherein the data is processed by an application in a shared asset using the preferred method for the selected operation.

17. The program carrier device of claim 11, wherein the data is processed by an application using a dynamically-bound instance of the preferred method for the selected operation in the application.

18. The program carrier device of claim 11, further comprising:

compiling additional methods in a shared asset; and

repeating the steps of claim 11.

19. The program carrier device of claim 11, wherein the multiple methods comprise calculations and subroutines.

20. The program carrier device of claim 19, wherein the calculations comprise interpretation calculations and drilling calculations,

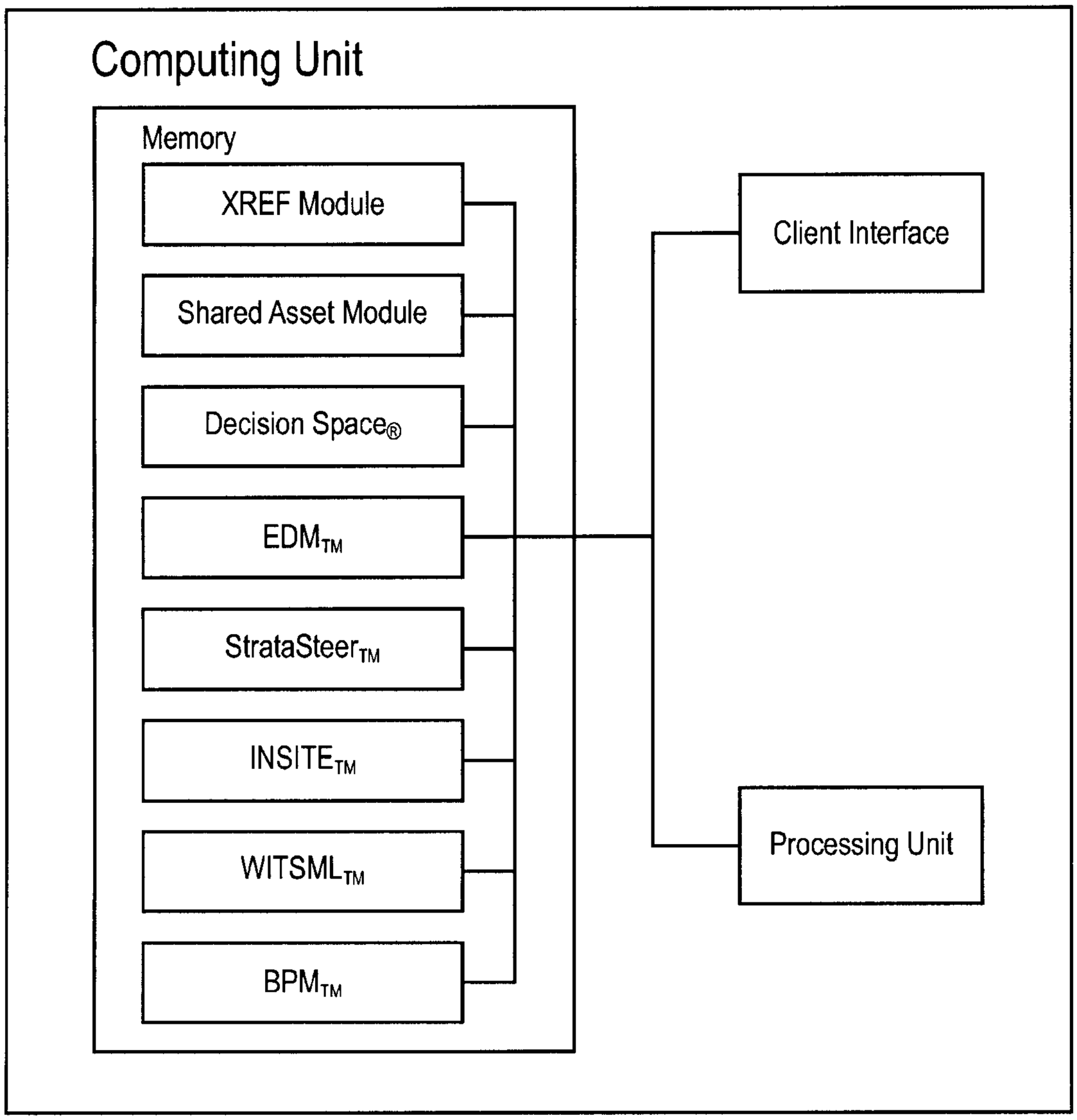


FIG. 1

CA 3000734 2018-04-10

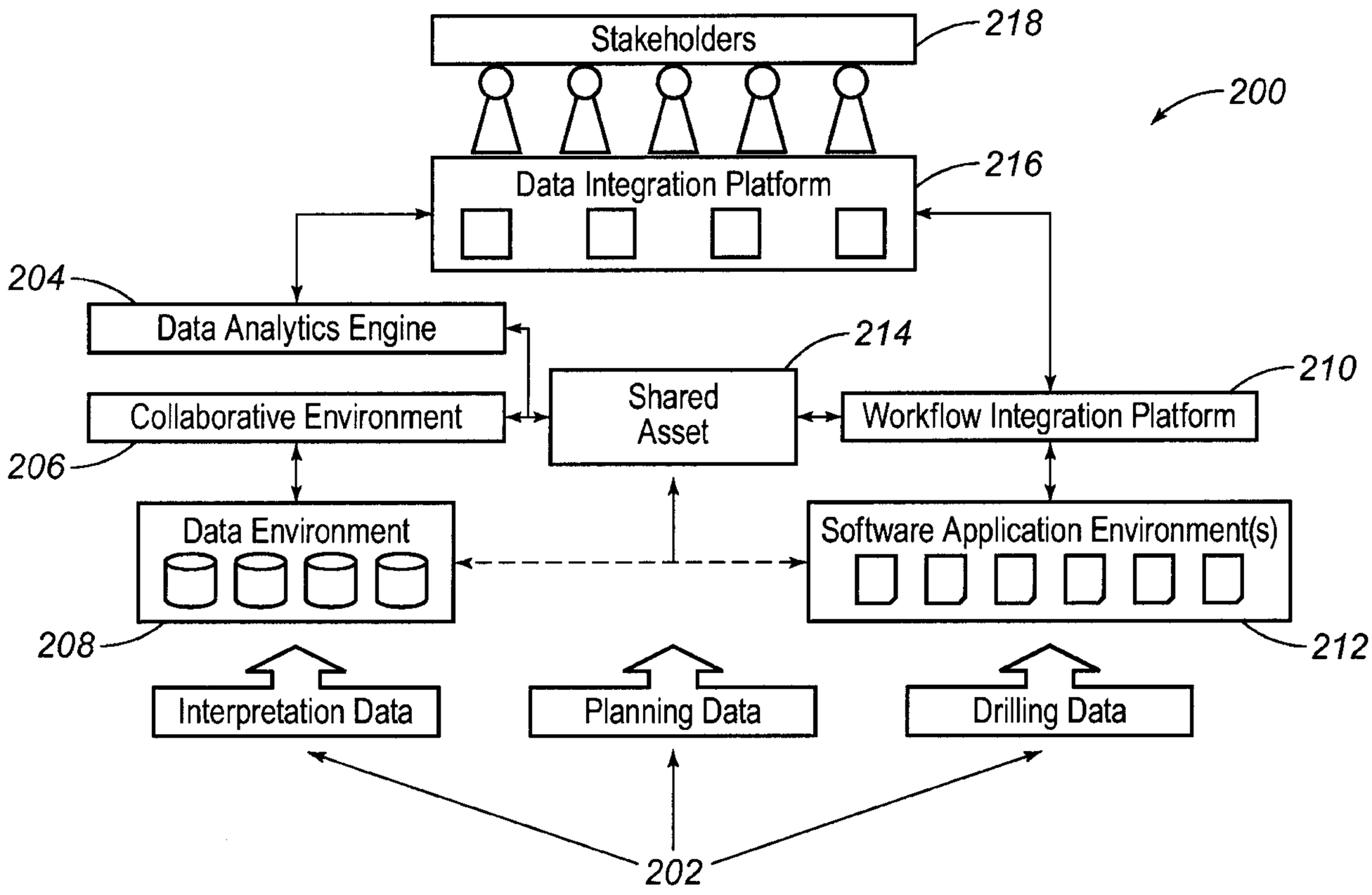

Stakeholders
218
200
Data Integration Platform
216
204
Data Analytics Engine
214
210
Collaborative Environment
Shared Asset
Workflow Integration Platform
206
Data Environment
Software Application Environment(s)
208
212
Interpretation Data
Planning Data
Drilling Data
202

FIG. 2

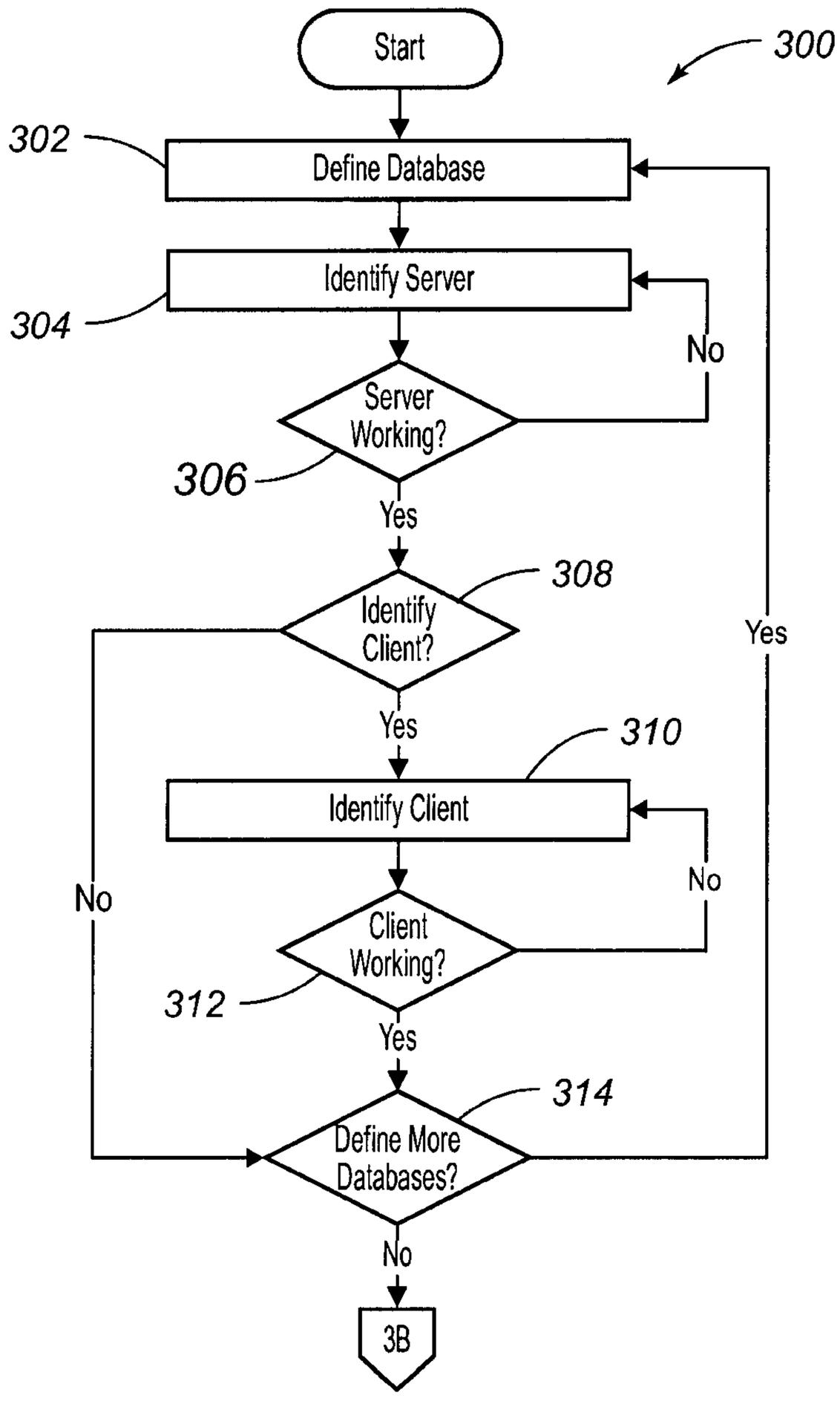


FIG. 3A

CA 3000734 2018-04-10

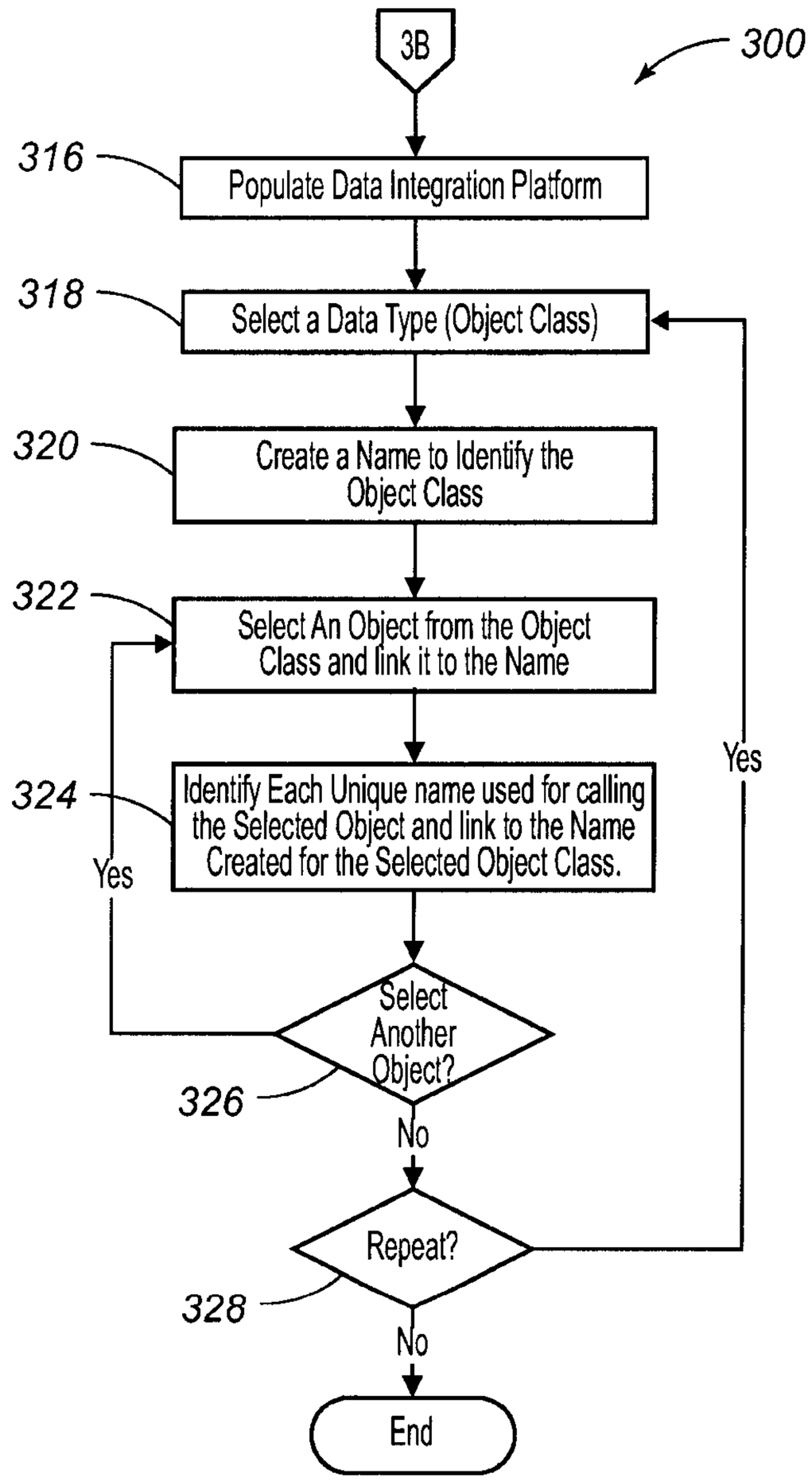


FIG. 3B

CA 3000734 2018-04-10

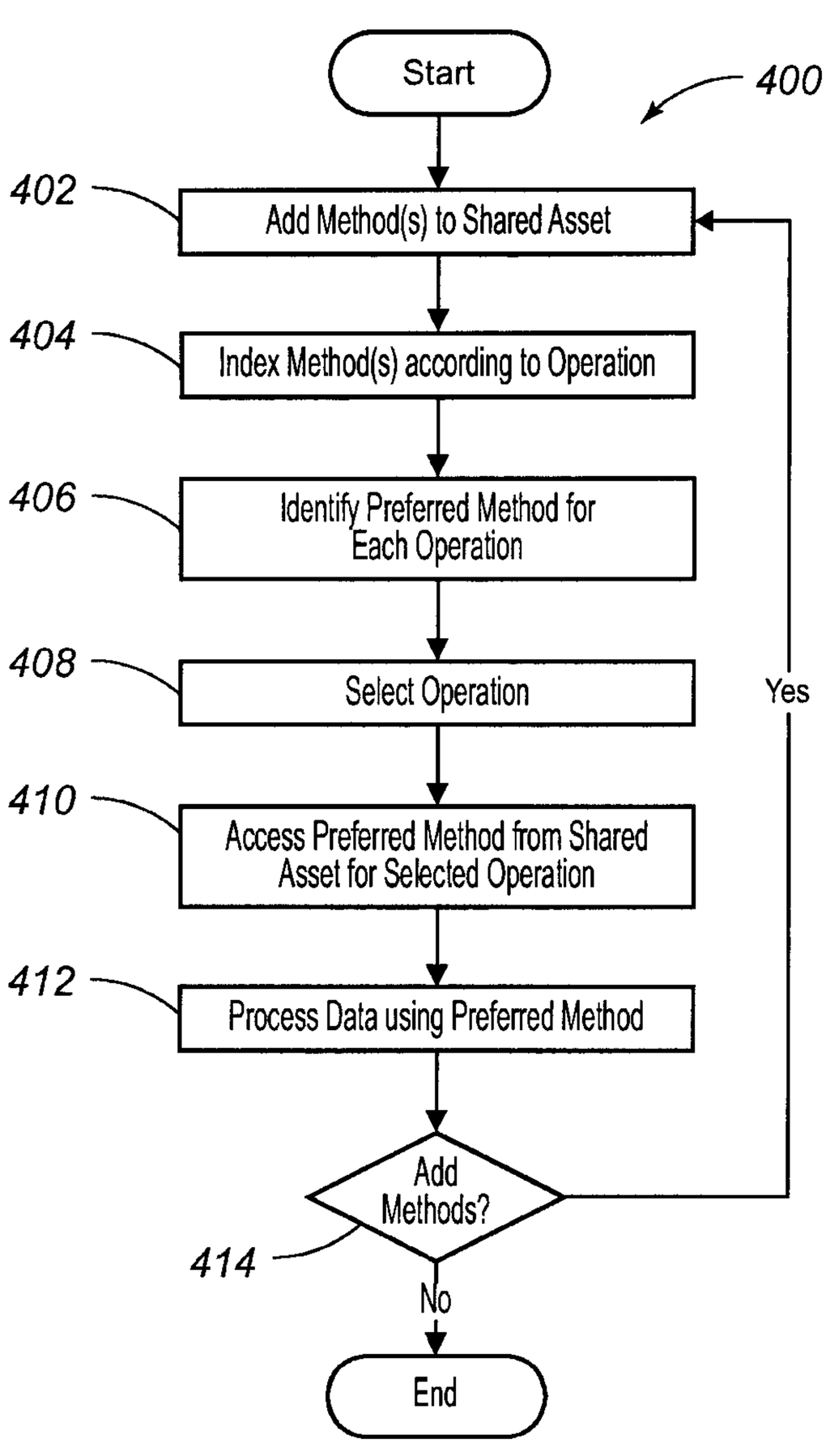


FIG. 4

CA 3000734 2018-04-10

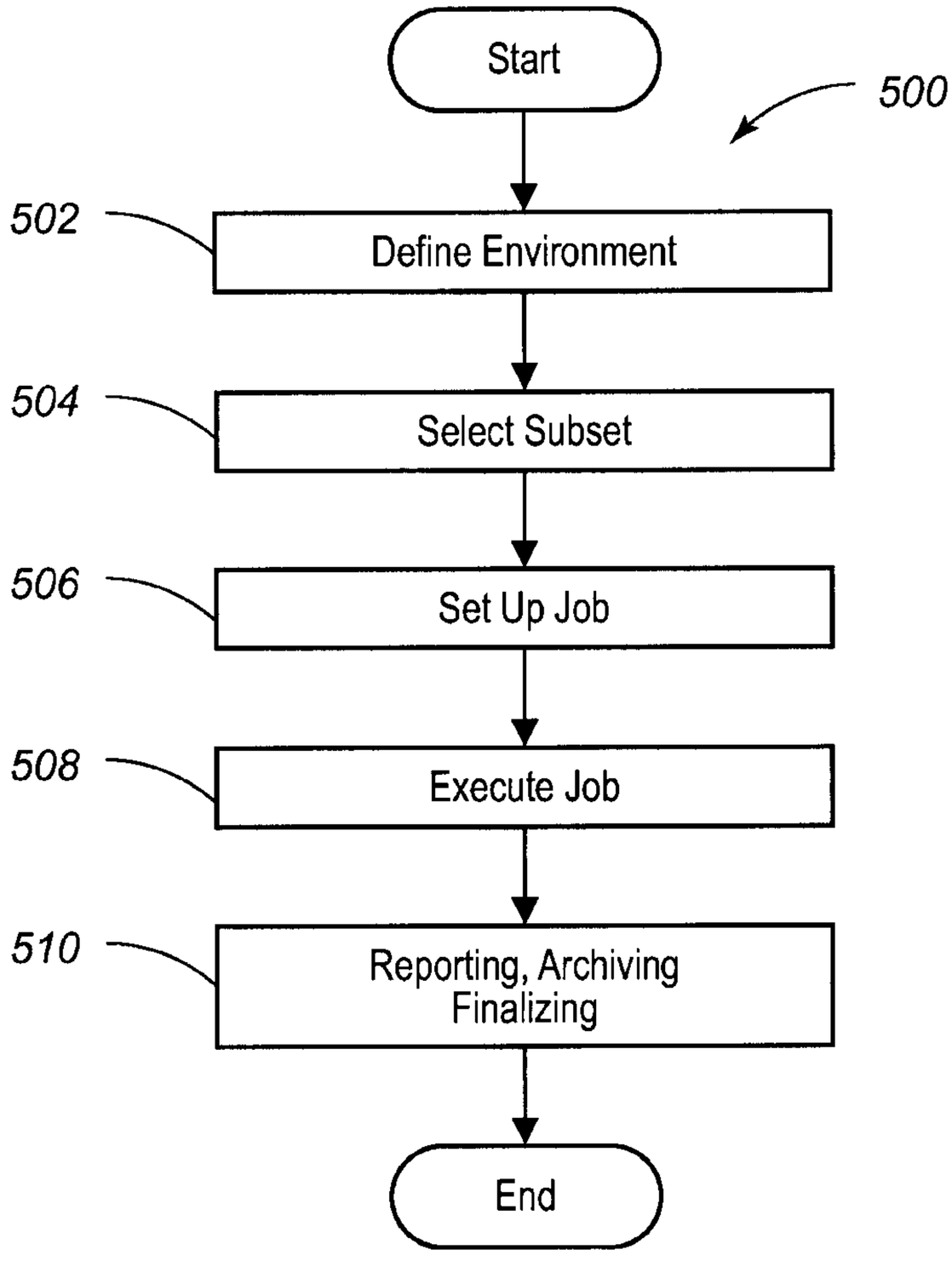

Start
500
502
Define Environment
504
Select Subset
506
Set Up Job
508
Execute Job
510
Reporting, Archiving
Finalizing
End

FIG. 5

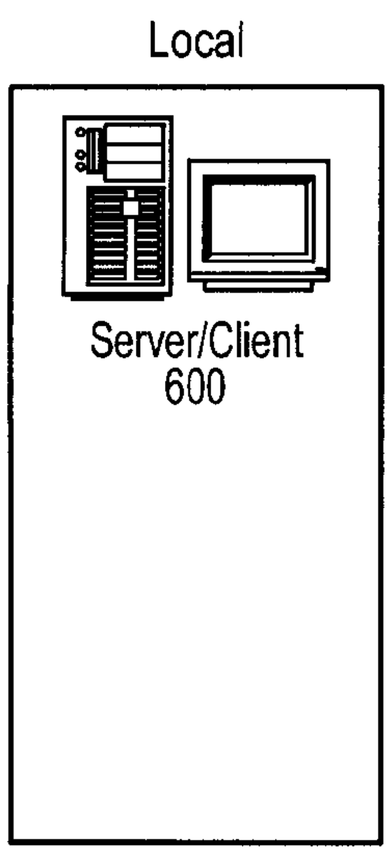


FIG. 6A

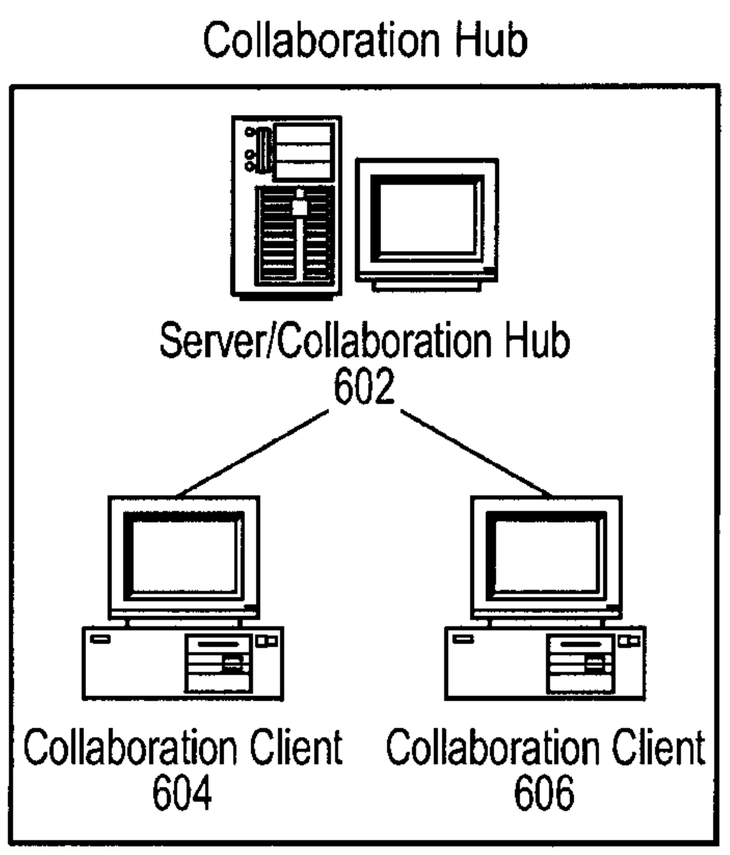


FIG. 6B

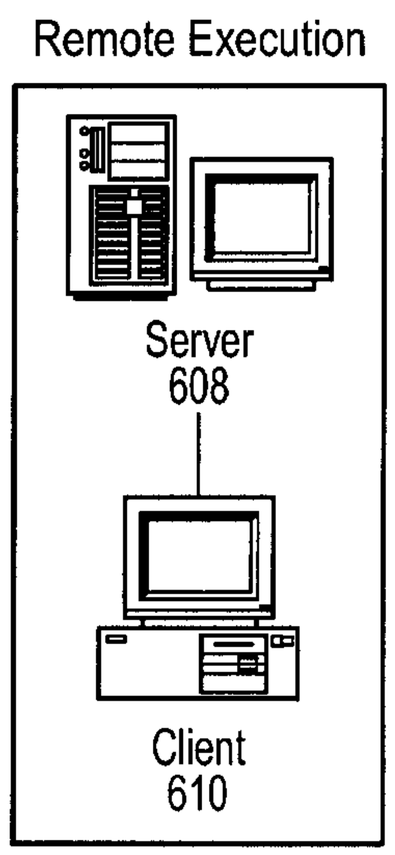


FIG. 6C

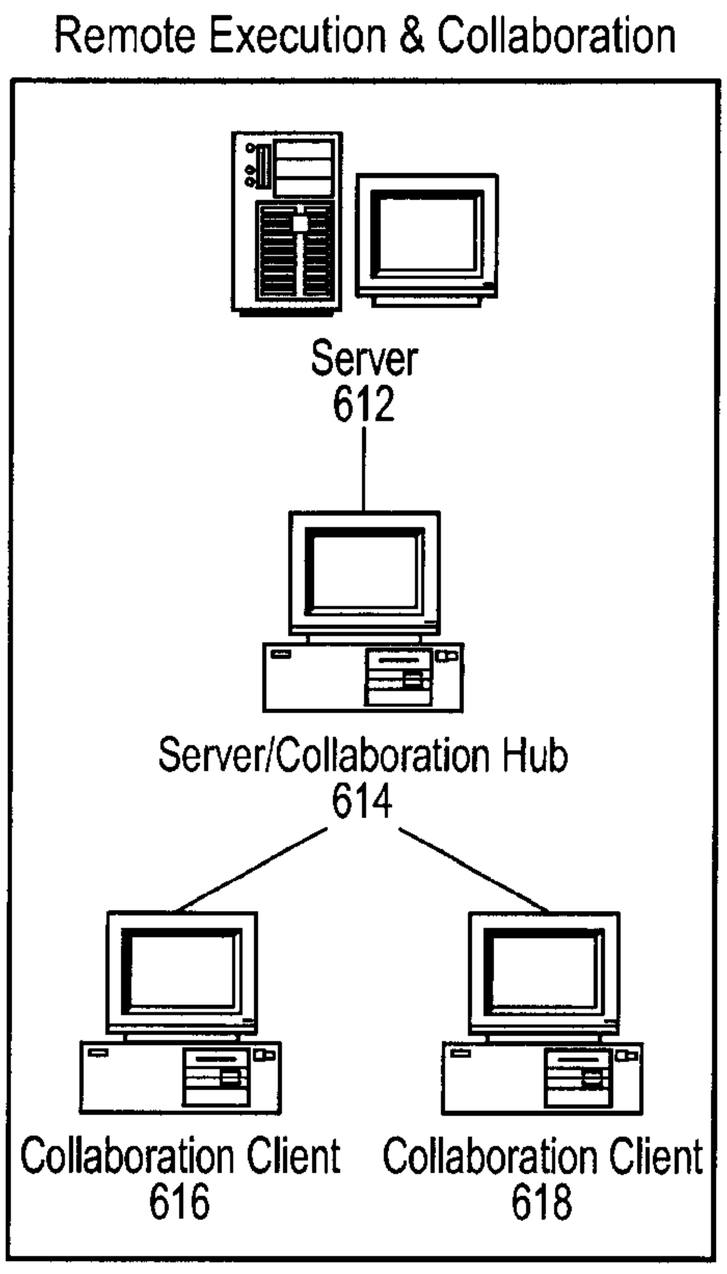


FIG. 6D

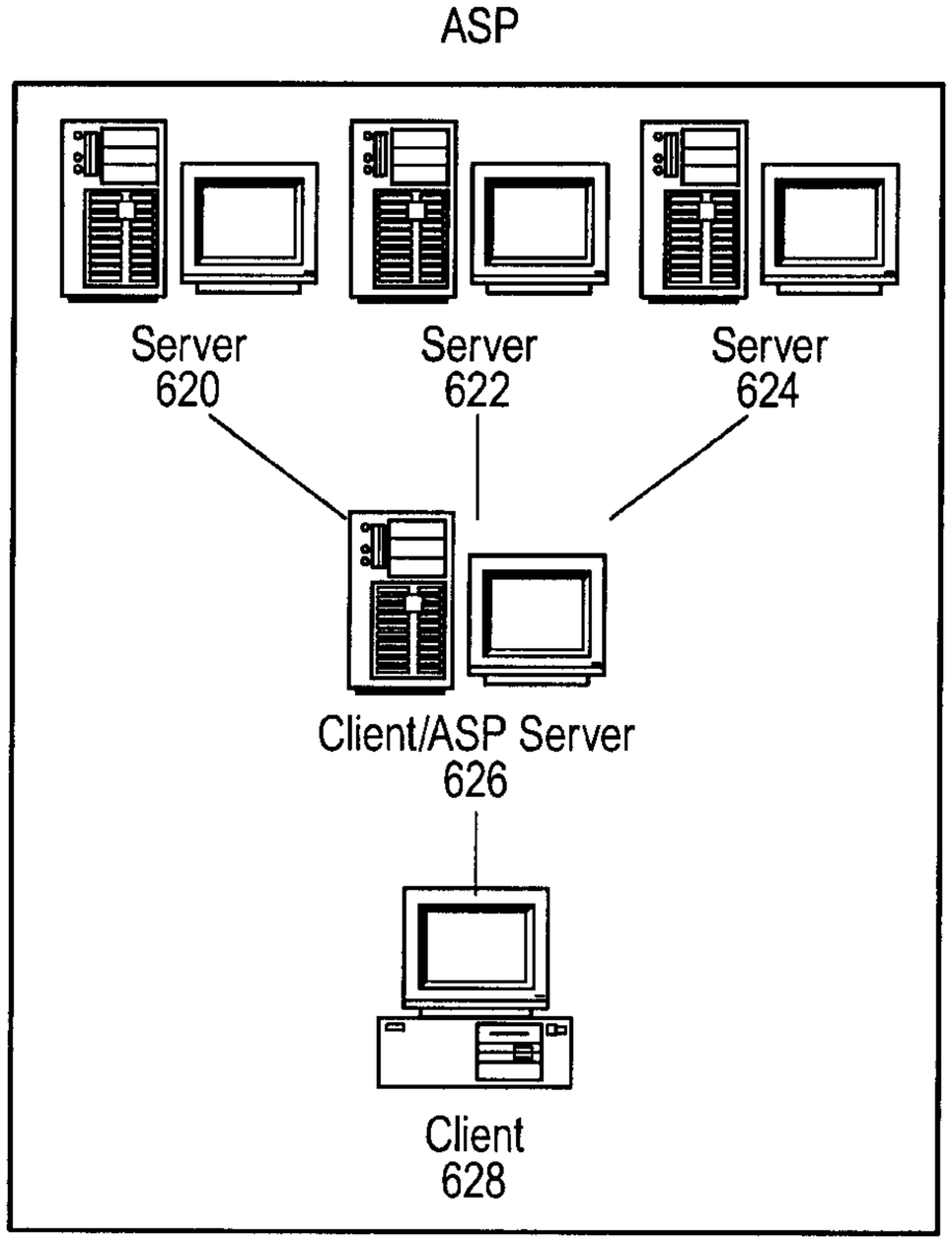

ASP
Server
620
Server
622
Server
624
Client/ASP Server
626
Client
628

FIG. 6E

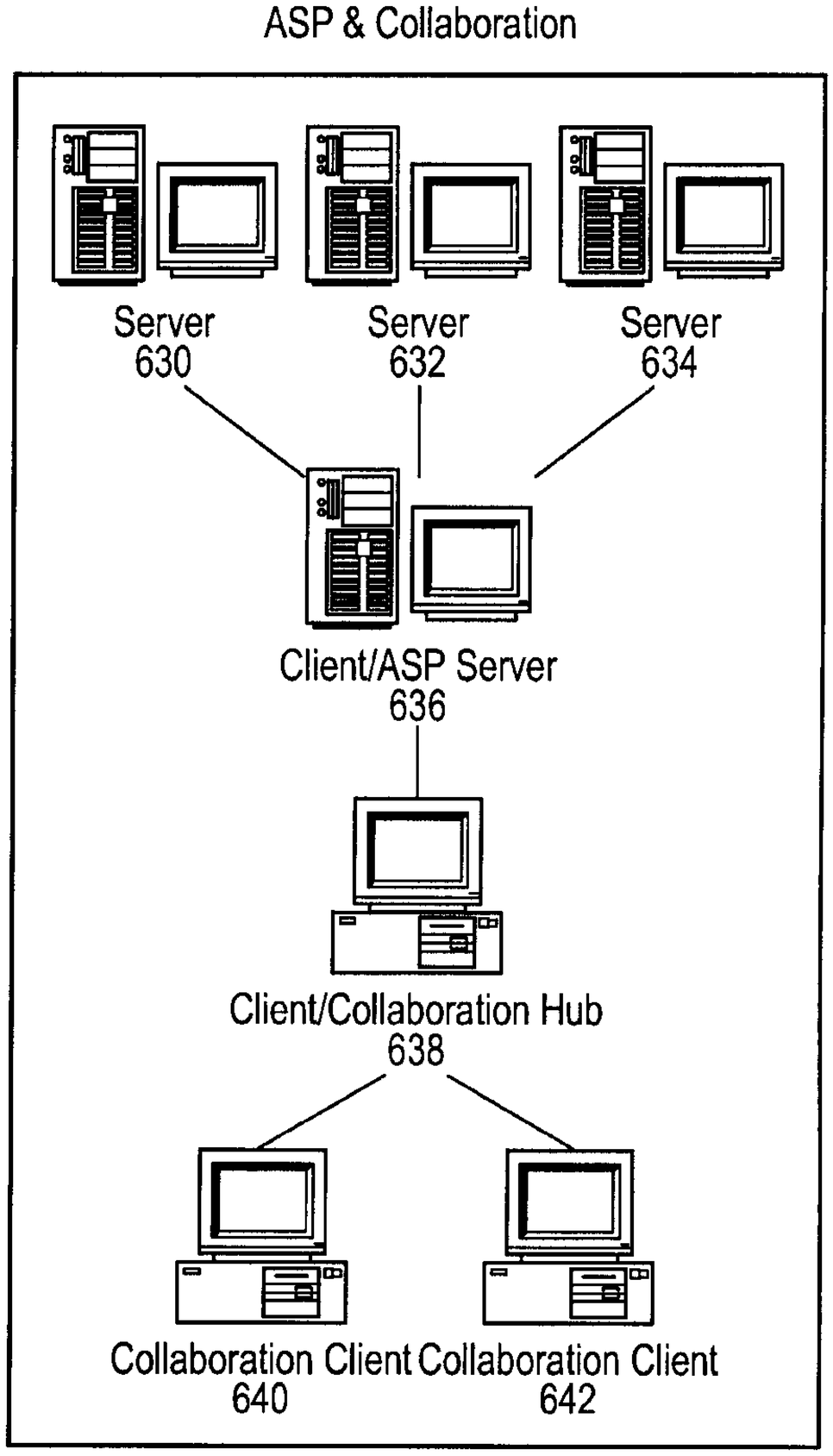
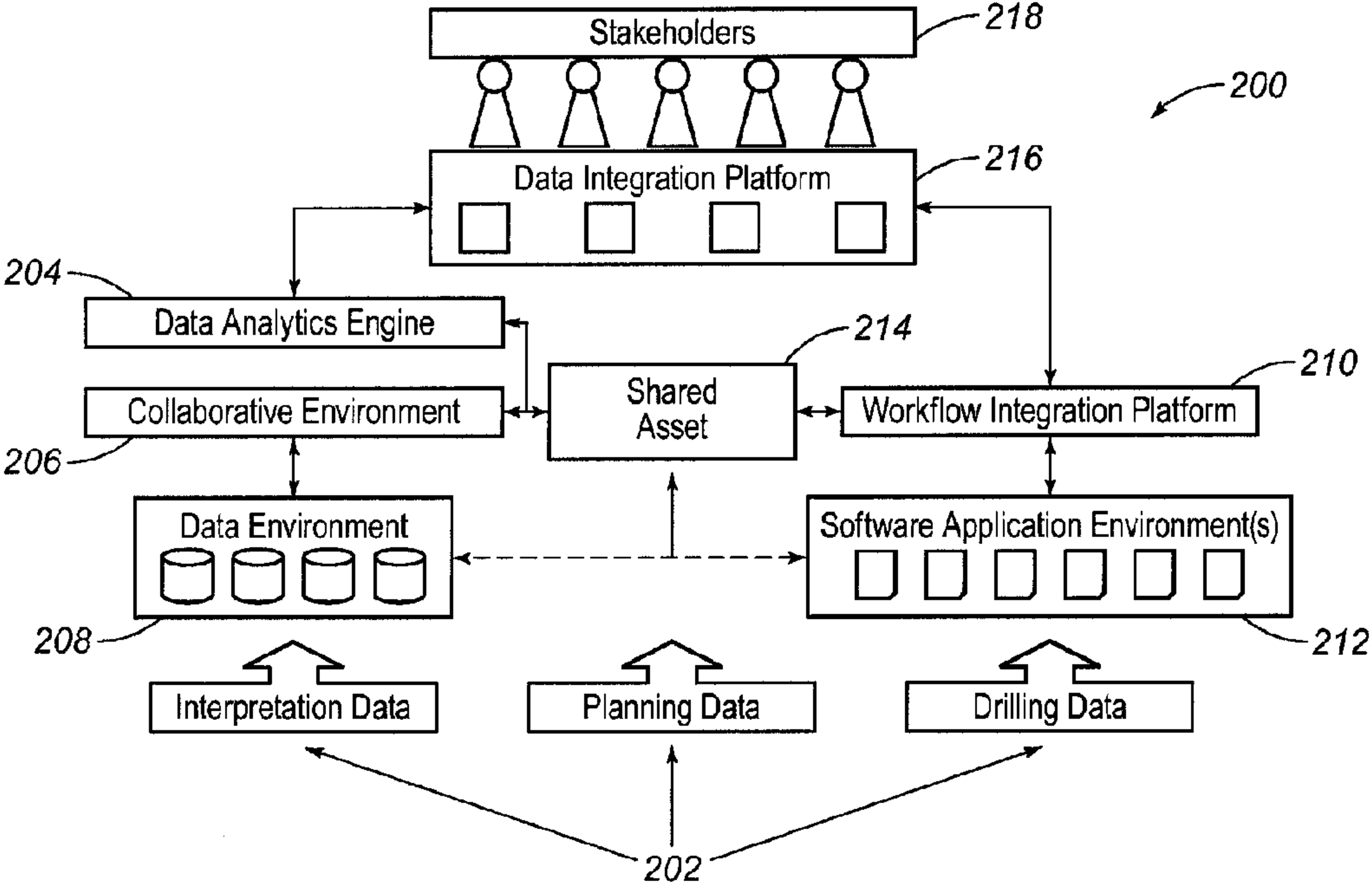

ASP & Collaboration
Server
630
Server
632
Server
634
Client/ASP Server
636
Client/Collaboration Hub
638
Collaboration Client
640
Collaboration Client
642

FIG. 6F


Stakeholders
218
200
Data Integration Platform
216
204
Data Analytics Engine
214
210
Collaborative Environment
Shared Asset
Workflow Integration Platform
206
Data Environment
Software Application Environment(s)
208
212
Interpretation Data
Planning Data
Drilling Data
202