

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
18 May 2007 (18.05.2007)

PCT

(10) International Publication Number
WO 2007/056766 A2

(51) International Patent Classification:

G06F 15/173 (2006.01)

(21) International Application Number:

PCT/US2006/060721

(22) International Filing Date:

9 November 2006 (09.11.2006)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

11/269,551 9 November 2005 (09.11.2005) US

11/269,638 9 November 2005 (09.11.2005) US

11/269,637 9 November 2005 (09.11.2005) US

(71) Applicant (for all designated States except US): **COM-PUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, New York 11749 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **HARVEY, Richard Hans** [AU/AU]; 4 Odette Court, Ringwood East, Victoria 3135 (AU). **GARDINER, Benjamin Michael** [AU/AU]; 75 Abbeygate Street, Oakleigh, Victoria 3166 (AU).

(74) Agents: **TOERING, Rick A.** et al.; **PILLSBURY WINTHROP SHAW PITTMAN LLP**, P.O. Box 10500, Mclean, Virginia 22102 (US).

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

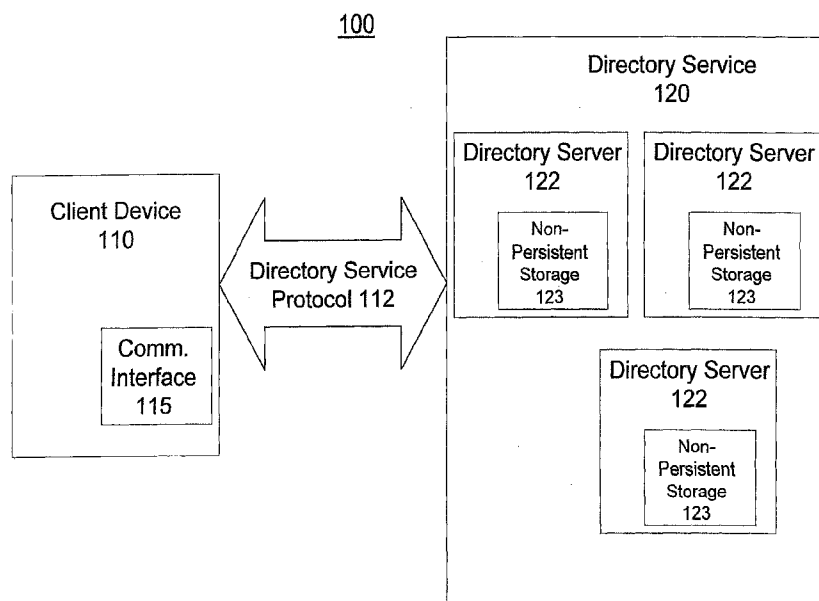
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR EFFICIENT DIRECTORY PERFORMANCE USING NON-PERSISTENT STORAGE



(57) Abstract: A system and method are provided for processing directory service operations. The system includes a client device communicatively coupled to one or more directory servers. Each directory server includes a communications interface, a storage mechanism, and an evaluation module. According to an exemplary embodiment, the storage mechanism is a non-persistent storage mechanism which increases read and write performance. When a directory server receives a directory service operation, it determines whether it is capable of processing the operation, and if so, processes it with respect to its non-persistent storage mechanism.

WO 2007/056766 A2

SYSTEM AND METHOD FOR EFFICIENT DIRECTORY PERFORMANCE USING NON-PERSISTENT STORAGE

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to the following applications, all of which were filed on 09 November 2005, and all of which are incorporated herein by reference: "System and Method for Routing Directory Service Operations in a Directory Service Network," U.S. 11/269,551; "System and Method for Providing a Directory Service Network," U.S. 11/269,638; and "System and Method for Efficient Directory Performance Using Non-Persistent Storage," U.S. 11/269,637.

[0002] This application is also related to the following co-pending applications, which are incorporated herein by reference: "Method and System for Configuring a Supplemental Directory," Attorney Docket No. 063170.8171; "Method and System for Providing Enhanced Read Performance for a Supplemental Directory," Attorney Docket No. 063170.8172; "Method and System for Improving Write Performance in a Supplemental Directory," Attorney Docket No. 063170.8173; "Method and System for Automatic Registration of Attribute Types," Attorney Docket No. 063170.8203; "System and Method for Writing Data to a Directory," Attorney Docket No. 063170.3119; and "Method and System for Providing a Directory Overlay," Attorney Docket No. 063170.8174.

[0003] This application is also related to the following applications, which were filed on May 20, 2005, and are incorporated herein by reference: "Method and Apparatus for Enhancing Directory Performance," U.S. Patent Application Serial No. 11/134,047; "Method and Apparatus of Optimizing Directory Performance," U.S. Patent Application Serial No. 11/134,143; "Method and Apparatus for Handling Directory Operations," U.S. Patent Application Serial No. 11/134,251; "Method and Apparatus for Loading Data into an Alternate Evaluator for Directory Operations," U.S. Patent Application Serial No. 11/134,043; "Structure of an Alternate Evaluator for Directory Operations," U.S. Patent Application Serial No. 11/134,237; and "Method of Selecting a Processor for Query Evaluation," U.S. Patent Application Serial No. 11/134,070.

FIELD OF THE INVENTION

[0004] The invention relates to a system and method for processing directory service operations using non-persistent storage.

BACKGROUND OF THE INVENTION

[0005] Directory services are commonly used to manage identities, policies, security information, and network resources. A directory service typically organizes electronic content and runs on a directory server. The directory service is the interface to a directory which holds information about objects that are to be managed by the directory service.

[0006] Directories differ from databases in several ways. A primary difference is that information in a directory is generally read more often than it is written. Still, information has to at times be written to directories. Additionally, data in a directory is accessed using one or more directory service protocols. A client requesting from a directory need not know which physical device stores the data. There is an ongoing need to increase directory read and write performance without sacrificing consistency and reliability.

[0007] Past attempts to increase read and write performance have resulted in increased complexity of operations and increased cost. Previous attempts at increasing read performance have included the use of various caching techniques. However, in a mixed read/write environment, read performance still suffers in order to maintain consistent information. Caching as well as turning off transaction logging has been used in attempts to increase write performance. However, these techniques do not ensure reliability because information may be lost in the event of a power failure or other service interruption.

[0008] The inventor has determined that it would be desirable to provide a system and method to increase both read and write performance of existing systems without sacrificing consistency and reliability.

SUMMARY OF THE INVENTION

[0009] A system and method are provided for efficiently evaluating directory service operations. The system of the present invention provides a directory server that uses non-persistent storage in order to expedite the process of handling directory service operations. The directory server may include a communications interface, a non-persistent storage mechanism, an evaluation module, and a directory service knowledge module.

[0010] The communications interface may be configured to receive one or more directory service operations from one or more client devices and to provide responses to the client device. The evaluation module may be configured to determine whether the directory server is equipped to perform a received directory service operation and to perform the operation. The non-persistent storage mechanism may be configured to store data that may be accessed by the client devices.

[0011] According to some embodiments, the directory server may be part of a directory service network. As such, the directory service knowledge module may be configured have knowledge of the directory service network and the capabilities of each directory server in the directory service network.

[0012] According to some embodiments of the invention, the server may additionally interface to a persistent storage mechanism. As such, certain predetermined types of data may be stored persistently.

[0013] In operation, a directory server may receive a directory service operation from a client device. The directory server may then determine whether it is equipped to evaluate and perform the operation. If so, the operation is evaluated with respect to the server's non-persistent storage device. If not, the server may forward the operation to another server, or may send a message to the client indicating that it is unable to perform the operation.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 illustrates client/server directory service system, in accordance with various embodiments of the invention.

[0015] FIG. 2 illustrates a directory server, in accordance with various embodiments of the invention.

[0016] FIG. 3 illustrates a process for evaluating a directory service operation, in accordance with various embodiments of the invention.

[0017] FIG. 4 illustrates a directory service network, in accordance with various embodiments of the invention.

[0018] FIG. 5 illustrates a directory server knowledge module, in accordance with various embodiments of the invention.

[0019] FIG. 6 illustrates a process for mutually updating directory server, in accordance with various embodiments of the invention.

[0020] FIG. 7 illustrates a directory service network, in accordance with various embodiments of the invention.

[0021] FIG. 8 depicts a plurality of directory service operations, in accordance with various embodiments of the invention.

[0022] FIG. 9 depicts a process for selecting a server to respond to a directory service operation, in accordance with various embodiments of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0023] In accordance with various disclosed embodiments, a system and method are provided for efficient directory service performance. Persistent writes to disk or other

permanent storage need not be a mandatory part of a directory server. For example, transient data such as a session object created when a user logs into a system, need not be maintained once the user logs out. In another example, tasks created as a part of a workflow application need not be maintained once the task is complete.

[0024] In both of the above examples, the data is created as required and it is not necessary that the data pre-exist before being used. Furthermore, the data may be removed once its usage is complete, and the data need not survive a machine failure because the data may be regenerated as needed. Leveraging these concepts, the present invention uses non-persistent information storage to expedite the data read and write processes.

[0025] Figure 1 depicts a client/server system 100 for providing efficient directory service access, according to some embodiments of the invention. System 100 may include client 110 communicatively coupled over a network to directory service network 120. Client 110 and directory service network 120 may communicate using one or more directory service protocols 112. Directory service network 120 may include one or more directory servers 122.

[0026] Client 110 may be a person, an application, a directory, and/or any network accessible electronic device. For example, client device may be a desktop computer, a laptop computer, a portable digital assistant, a wireless telephone, and/or other electronic devices. Client 110 may include a directory services communication interface 115 enabling communication with directory service network 120.

[0027] Directory service protocols 112 may include any known or proprietary directory service protocols such as, for example, lightweight directory access protocol (LDAP), X.500, directory service markup language (DSML), and/or other directory service protocols. Directory service protocols 112 enable a user at client device 110 to access one or more directory entries stored on one or more of directory servers 122.

[0028] Directory services 112 enable clients to locate directory entries and attributes associated with the entries. Each directory server 122 may store directory entries in a hierarchical, tree-like structure. Each directory entry may comprise a collection of attributes having a distinguished name. Additionally, each entry's attributes may have a type and one or more values. The distinguished name may be constructed by taking the name of the entry, known as the relative distinguished name, and concatenating the names of its ancestors in the hierarchy.

[0029] Directory service protocols 112 define a plurality of operations for accessing data stored with the directory service network. For example, the protocol may define operations for searching the directory, adding an entry, updating an entry, deleting an entry, and/or other

operations. When client device 110 issues a directory service operation, the client need not be aware of which directory server 122 is responsible for evaluating the operation.

[0030] According to an exemplary embodiment of the invention, one or more directory servers 122 may include non-persistent storage mechanism 123. In a typical client/server directory service arrangement, a persistent storage mechanism, such as a database, file system, and/or other permanent storage device may be used for evaluating directory service operations. As used herein, persistent storage refers to permanent, static, long-term storage mechanisms. Persistent data is used herein to refer to data that is read from and/or written to a persistent storage mechanism. Each time a read or write directory operation is performed, the responding directory server consults the persistent storage device to either retrieve an object or write an object to the storage device. However, as described above, many directory service operation do not require permanent storage.

[0031] Referring now to Figure 2, a directory server 200 is depicted, in accordance with various embodiments of the invention. Directory server 200 may include a communications interface 210, a non-persistent storage mechanism 220, and evaluation module 230. Additionally, directory server 200 may include directory server knowledge module 240.

[0032] Communications interface 210 may be configured to receive directory service operations from one or more client devices. Additionally, communications interface 210 may be configured to forward directory service operations to other directory servers or to receive forwarded directory service operations when connected as part of a network.

[0033] Directory server 200 may include non-persistent storage mechanism 220. Non-persistent storage mechanism 220 may be a portion of memory associated with server 200, according to some embodiments of the invention. While depicted in Figure 2 as internal to directory server 200, non-persistent storage mechanism 220 may be an external storage mechanism. Non-persistent storage mechanism 220 may initially be empty, according to some embodiments of the invention. In other embodiments, non-persistent storage mechanism 220 may be initialized with information stored in other persistent or non-persistent storage mechanisms. Updates to the data, such as modifications, additions, and/or deletions, may be stored in non-persistent storage mechanism 220. Additionally, read operations may be evaluated with respect to the data stored in non-persistent storage mechanism 220.

[0034] According to some embodiments of the invention, non-persistent storage mechanism 220 may be an alternate evaluator. The alternate evaluator may be configured to cache an entire directory store. The directory store may be a persistent storage mechanism.

Both read and write operations may be evaluated in relation to the alternate evaluator. Data may be read from the alternate evaluator and operations that add or modify data may be written to the alternate evaluator.

[0035] Evaluation module 230 may be configured to determine whether directory server 200 is equipped to perform a received directory service operation. For example, directory server 200 may not be equipped to perform a read operation if its accessible storage mechanism(s) is empty. According to some embodiments of the invention, evaluation module 230 may consider only the contents of non-persistent storage mechanism 220 in determining whether directory server 200 may perform a received directory service operation. In other embodiments, evaluation module 230 may consider non-persistent storage mechanism 220 in addition to other storage mechanisms. Additionally, evaluation module 230 may be configured to perform the requested operation. Directory server knowledge module 240 may be configured to query other directory servers within a directory service network to resolve an operation that evaluation module 230 has determined may not be performed by directory server 200.

[0036] Figure 3 depicts a process 300 for receiving and evaluating a directory service operation, according to various embodiments of the invention. As depicted at 302, a directory server may receive a directory service operation. The directory server may determine what type of operation it has received. For example, the operation may be an add, modify, search, remove, or other directory service operation.

[0037] The directory server may then determine whether the received operation depends on existing information, as depicted at 304. For example, to perform a search operation, the server must have access to an information storage mechanism that already includes data. For an add operation, the server need not have access to a storage mechanism that contains data.

[0038] If the operation does depend on existing information, the directory server may then determine whether it is able to process the operation, as depicted at 306. As described above, a directory server may be configured to evaluate a directory service operation solely with respect to its associated non-persistent storage mechanism or to additionally evaluate an operation with respect to other storage mechanisms. Thus, the directory server may determine whether it has access to the stored data. According to some embodiments of the invention, a non-persistent storage mechanism may be loaded or pre-configured with information from persistent or non-persistent storage mechanism, or other directory servers after which the directory server may commence evaluation of operations. In other embodiments, data may be loaded into the non-persistent storage mechanism via add

operations, after which more complex operations such as search, compare, read, list, modify, rename, delete, and/or other operations, may be performed.

[0039] If the directory server is unable to process the directory service operation, it may, according to some embodiments, query another server to process the request, as depicted at 308. Alternatively, the server may respond with a message indicating it is unable to complete the request, as depicted at 310.

[0040] As depicted at 312, the directory server may perform the directory service operation. According to some embodiments of the invention, the server may store the result of the directory service operation to its non-persistent storage mechanism. A directory server, according to some embodiments of the invention, may include or interface to a persistent storage mechanism in addition to a non-persistent storage mechanism. As such, certain predefined categories of data may be written to the persistent storage mechanism. For example, configuration data may be stored in the persistent storage mechanism and may be retrieved and/or updated as required. Additionally, certain types of data may be stored and/or pre-loaded in the persistent storage mechanism and may be used during evaluation of a particular operation.

[0041] Figure 4 depicts another example of a directory service network, in accordance with various embodiments of the invention. Directory service network 400 may include a plurality of directory servers 402-408. Directory servers 402-408 may be any type of directory server and may interface to a non-persistent and/or persistent storage mechanism. Directory servers 402-408 may be replicated servers, each providing the same type of information, according to some embodiments of the invention. In other embodiments of the invention, directory servers 402-408 may form a distributed network, wherein one or more of directory servers 402-408 store certain types or categories of data.

[0042] As depicted in Figure 4 by way of example only, directory server 402 includes an internal non-persistent storage mechanism, directory server 404 interfaces to an external non-persistent storage mechanism, directory server 406 includes a non-persistent storage mechanism and additionally interfaces to external persistent storage mechanism 407, and directory server 408 interfaces to external persistent storage mechanism 409. Other server configurations may be used. For example, each server may interface only to non-persistent storage mechanisms.

[0043] As described above, each directory server may include an evaluation module and a directory server knowledge module. Figure 5 depicts an exemplary directory server knowledge module, in accordance with various embodiments of the invention. Directory

server knowledge module 500 may include one or more sub-modules such, for example, server knowledge maintenance module 510, initialization module 520, and mutual updating module 530.

[0044] Server knowledge maintenance module 510 may be used to maintain a list of all servers in directory service network 400. The list may provide information such as, for example, a name associated with the directory server, a network address associated with the directory server, a location of the server, the server status, the type of data maintained by the server, and/or other information. Information maintenance module 510 may be configured to update information a directory server in the directory service network and to add new servers to the server list. According to some embodiments of the invention, each server maintains a server list. In other embodiments of the invention, a dedicated server may be configured to maintain the server list.

[0045] Mutual updating module 530 may be configured to enable one or more servers in a directory service network to share received directory service updates with one or more other servers in the network.

[0046] Figure 6 depicts a process 600 for mutually updating a directory server in a directory server network. As depicted at 602, a directory server may receive a directory service update operation. The directory server may then determine whether the update operation should be applied locally to its directory store, as depicted at 603. For example, the receiving directory server may determine whether the update operation refers to the type of data stored in its directory store. If it is determined that the operation should applied locally, the operation is performed as depicted at 606.

[0047] The receiving server may determine whether there are other directory servers within the directory service network that should receive the update whether or not the operation was performed locally, as depicted at 604. As described above, a directory server may include a directory server knowledge module which may maintain a knowledge list. The knowledge list may include information about other servers in the network. Determining whether another directory server should receive the update may include, for example, determining whether the other servers store the same type of information, whether the other servers interface to a persistent or non-persistent storage device, and/or other criteria.

[0048] If there are no additional servers in the receiving server's knowledge list which should receive the update, the receiving server may simply wait to receive the next operation, as depicted at 610. If there are additional servers in the network directory server list, the receiving server may forward the update to the additional server, as depicted at 608.

According to some embodiments of the invention, replication to other servers in the directory server network occurs after confirming the operation with the requesting client device. In other embodiments, replication may occur before confirming the update with the requesting client device. For example, replication to servers which interface to a persistent storage mechanism may be performed after confirming the operation with the client device while replication to servers which interface to a non-persistent storage mechanism may be performed before confirming the operation.

[0049] Mutually updating the additional servers may be performed using various synchronization or replication techniques. Table 1 illustrates an exemplary list of techniques that may be used to perform the mutual updating process. Other techniques may be used.

Table 1

Technique	Description
Write-Behind (Asynchronous) Replication	Update to another server occurs outside the update transaction
Write-Through (Synchronous) Replication	Update to another server occurs inside the update transaction
Replay-Based Replication	Every Update to one server is applied to another server
State-Based Replication	Final state of one or more changes is conveyed to another server
Meshed Replication	Server is capable of updating another server directly
Cascading Replication	Mutual update propagates through the network from one server to another
Tiered Replication	Mutual update propagates through the network in a hierarchy
Multi-Master Replication	Any server may receive client update operations and replicate them to another server
Preferred-Master Replication	System has an order of preference when it needs to replicate to another server
Master-Slave Replication	Slave servers do not receive client update operation and are only updated via a master server
Full Replication	Each replicated server receives a complete copy of information from another server
Selective Replication	Subset of information is replicated from one server to another
On-Demand Replication	Replication to another server occurs at the time of the update
Batch Replication	Replication to another server occurs at predetermined time or event

[0050] Initialization sub-module 520 may be configured to enable a directory server to receive information stored by a second directory server in the network, prior to bringing the server online. In a replicated directory server network, a new directory server may be initialized by any other directory server in the directory service network. In a distributed directory service network, a new directory server may serve as a replica for one or more other servers in the network. As such, the newly added server may be initialized by the one or more server in the network for which it is a replica.

[0051] According to various embodiments of the invention, a routing function may be used to increase the performance of a directory service network. Figure 7 depicts a system 700 for improving the performance of a directory service network. System 700 may include router 710 communicatively coupled to one or more directory servers 720. Additional routers may be included.

[0052] Router 710 may include a server knowledge list 712. Server knowledge list 712 may provide a list of servers in the network, and an indication of each server's status and/or other information. Status information may include, for example, an indication of the server's availability, a time associated with the last operation performed by the server, and/or other status information. A server may be unavailable, for example, if the server is offline, recovering from another operation, busy, or otherwise marked unavailable.

[0053] Server knowledge list 712 may also include information identifying the type or category of information managed by each server. In a replicated environment, multiple servers may serve the same information, whereas in a distributed environment, each server may serve a specific subset of information. In either environment, a client making a request need not be aware of the server allocation.

[0054] According to some embodiments of the invention, router 710 may be a directory server adapted to perform one or more routing functions. For example the router may be adapted to perform functions such as load sharing, fail-over, fail-back, and/or other routing functions. If router 710 is configured for fail-over and directs an operation to a server that fails, the operation can be redirected to another server without interrupting the client application. If router 710 is configured for fail-back and a new server is brought online, the new server is configured and used transparently without the need for client input.

[0055] As depicted in Figure 7, each directory server 720 may include a non-persistent information store. One or more persistent storage devices 730 may also be provided. Persistent storage device 730 may interface to any one or more of directory servers 720.

[0056] Load-sharing between the one or more directory servers 720 may be based on availability and other pre-determined criteria. For example, router 710 may be configured to route directory service operations to one of directory servers 720 in a round robin fashion wherein each server in turn receives a request. In some embodiments, the directory server having the least number of queued requests may be chosen by router 710. Other methods of choosing a directory server may be used such as, for example, choosing randomly, defining directory server precedence, operation complexity, operation type, security classification, and/or other methods.

[0057] Router 710 may be configured to respond to a plurality of events. As depicted in Figure 8, router 710 may be configured to respond to the receipt of a client directory operation request, as depicted at 802, receipt of a disconnect indication, as depicted at 804, receipt of a connection indication, as depicted at 806, and/or other events.

[0058] A router receiving a directory operation request may be adapted to forward the request to a directory service that can handle the request. This process, is known as navigation or chaining, and may be based on a unique attribute provided in the directory operation request. For example, deciding where to forward the request may be based on the distinguished name in the request. As described above, the distinguished name is a collection of attributes that make up a directory entry. Using the distinguished name, the router may locate a directory service that manages a namespace that includes the directory name of the target entry.

[0059] Figure 9 depicts a process 900 for determining a server which is capable of responding to a directory service operation and forwarding a directory operation request to the determined server. As depicted at 902, a router in a directory service network may receive a directory operation request. Upon receipt of the request, the router may consult its server knowledge list to determine one or more directory servers that may service the request, as depicted at 904. If no servers are available, the router may return an error response to the client.

[0060] The directory operation request may include, among other things, a distinguished name associated with the operation. As such, the router may determine which one or more servers manage the provided distinguished name. Other criteria may be used, as would be apparent.

[0061] As described above, one or more servers may be capable of responding to a directory operation request. As depicted at 906, a determination is made as to whether more than one server may service the request. The router may consult its knowledge list to determine whether more than one server manages the information identified in the directory operation request. If only one server is identified in the knowledge list, the request is forwarded to the identified server, as depicted at 908. If more than one server may respond to the request, the router may choose a server, as depicted at 910.

[0062] The router may determine which of the two or more directory servers to route a directory operation request to based on availability. A server may be unavailable if it is offline, recovering from another operation, busy, or otherwise marked unavailable. In addition to availability, a variety of methods to determine whether a directory operation

request should be routed to a server. As described above, this evaluation may be based on one or more of round robin, least busy, multicast, random, preferred server rankings, operation type, operation complexity, security, or any other suitable determination method. For example, if available servers are evaluated using least busy and preferred, the router first determines which server is least busy and then of those which is highest in the preference list. Once a server have been chosen, the directory operation request may be forwarded to the chosen server.

[0063] A router in a directory service network may additionally be configured to receive server disconnect indications. According to some embodiments of the invention, a router may receive a disconnect indication due to a server or network failure. Upon receipt of a disconnect indication, the router may mark the server as unavailable in its server availability list. The router may additionally store the time the disconnect indication was received. As such, the router may then send a connection request to the failed server after a pre-defined time-out period has elapsed if the server does not automatically reconnect.

[0064] The router may also receive connection indications from one or more directory servers joining the directory service network. When a new server comes online, it may be added to the router's available server list. Furthermore, the router may receive a connection indication when a previously failed server comes back online. The router may then unmark its failure in the availability list.

[0065] While the invention has been described with reference to the certain illustrated embodiments, the words that have been used herein are words of description, rather than words of limitation. Changes may be made, within the purview of the associated claims, without departing from the scope and spirit of the invention in its aspects. Although the invention has been described herein with reference to particular structures, acts, and materials, the invention is not to be limited to the particulars disclosed, but rather can be embodied in a wide variety of forms, some of which may be quite different from those of the disclosed embodiments, and extends to all equivalent structures, acts, and, materials, such as are within the scope of the associated claims.

WHAT IS CLAIMED IS:

1. A method of evaluating directory service operation comprising:
receiving, at a first directory server, a directory service operation; and
enabling the first directory server to process the directory service operation, with
reference to a non-persistent storage mechanism.
2. The method of claim 1, wherein data resident in the non-persistent storage mechanism
is used in the step of evaluation.
3. The method of claim 1, wherein only data resident in the non-persistent storage
mechanism is used in the step of evaluation.
4. The method of claim 1, wherein processing the directory service operation comprises
referencing the non-persistent storage mechanism and a secondary storage mechanism.
5. The method of claim 4, wherein the secondary storage mechanism is a persistent
storage mechanism.
6. The method of claim 1 wherein the first directory server is part of a directory service
network comprising a plurality of directory servers.
7. The method of claim 1, wherein when the first directory server is unable to process
the directory service operation, the method further comprises:
forwarding the directory service operation to one or more additional directory servers
in the directory service network.
8. The method of claim 1 further comprising:
determining whether the directory service operation is an update operation;
if the directory service operation is an update operation, determining whether the
operation is to be performed on a predefined type of data; and
if the directory service operation is to be performed on a predefined type of data,
updating a persistent storage mechanism with the result of the directory service operation.
9. The method of claim 1, wherein the directory service operation is received using at
least one of the following protocols: X.500 or lightweight directory access protocol.
10. A directory server for processing directory service operations comprising:
a communications interface configured to receive one or more directory service
operations from one or more client devices;
a non-persistent storage mechanism configured to temporarily store data for
responding to the directory service operations received by the communications interface; and
an evaluation module configured to process the directory service operation with
reference to the non-persistent storage mechanism.

11. The directory server of claim 10, wherein the non-persistent storage mechanism is an internal memory resident table.
12. The directory server of claim 10, wherein the non-persistent storage mechanism is an external storage device.
13. The directory server of claim 10, wherein the non-persistent storage mechanism is pre-loaded with data from a secondary storage mechanism.
14. The directory server of claim 10, wherein the non-persistent storage mechanism comprises an alternate evaluator configured to cache a directory store.
15. A directory system for evaluating directory service operations comprising:
 - one or more client devices, each of the client devices including a communications interface; and
 - a directory service network comprising a plurality of directory servers, wherein at least one of the directory servers comprises:
 - a communications interface configured to receive one or more directory service operations from the one or more client devices;
 - a non-persistent storage mechanism configured to temporarily store data for responding to the directory service operations received by the communications interface; and
 - an evaluation module configured to process the directory service operations with reference to a non-persistent storage mechanism.
16. The system of claim 15, wherein the one or more client devices and the directory service network communicate using a directory service protocol.
17. The system of claim 16, wherein the directory service protocol is at least one of X.500, LDAP, and DSML.

100

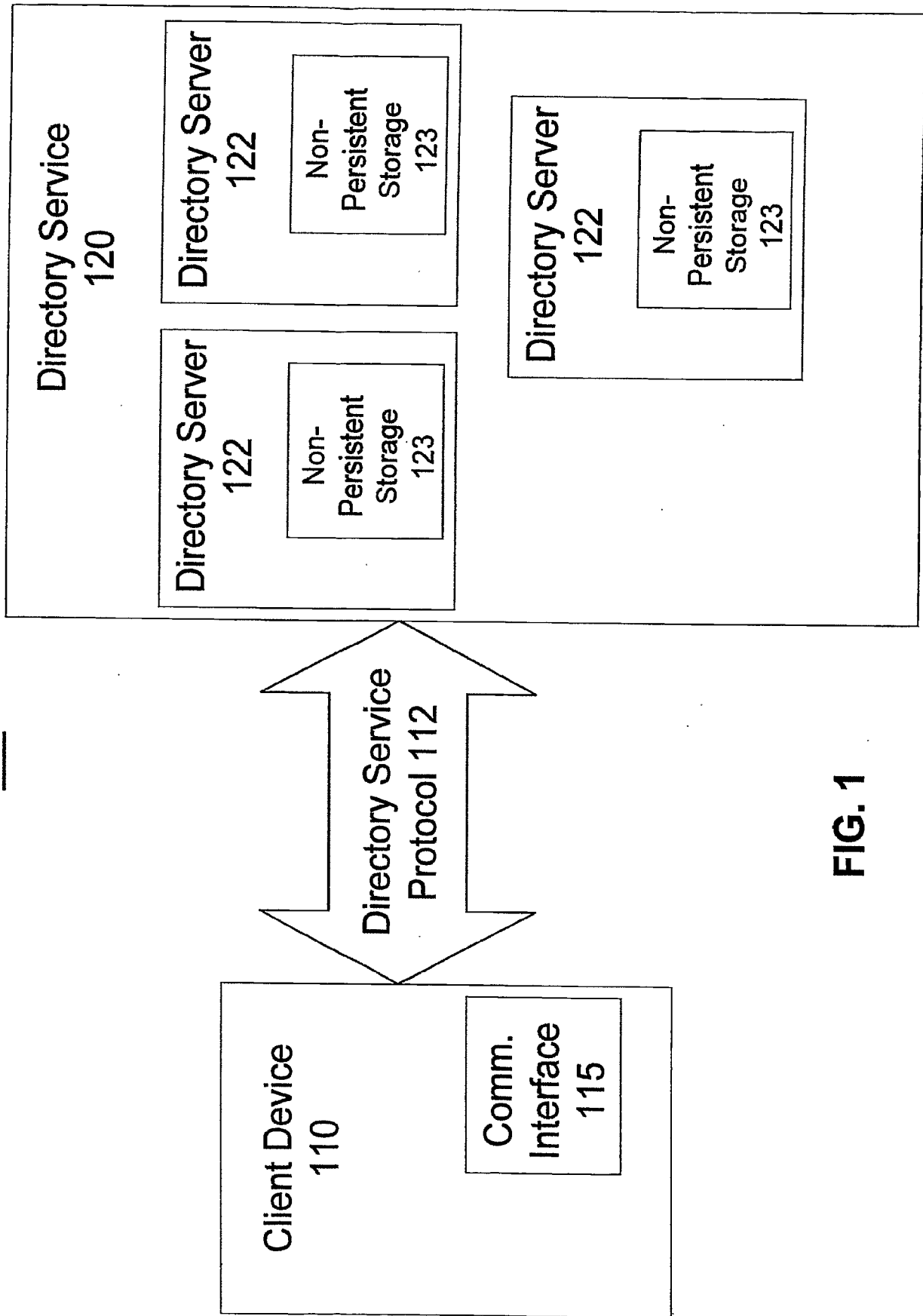


FIG. 1

200

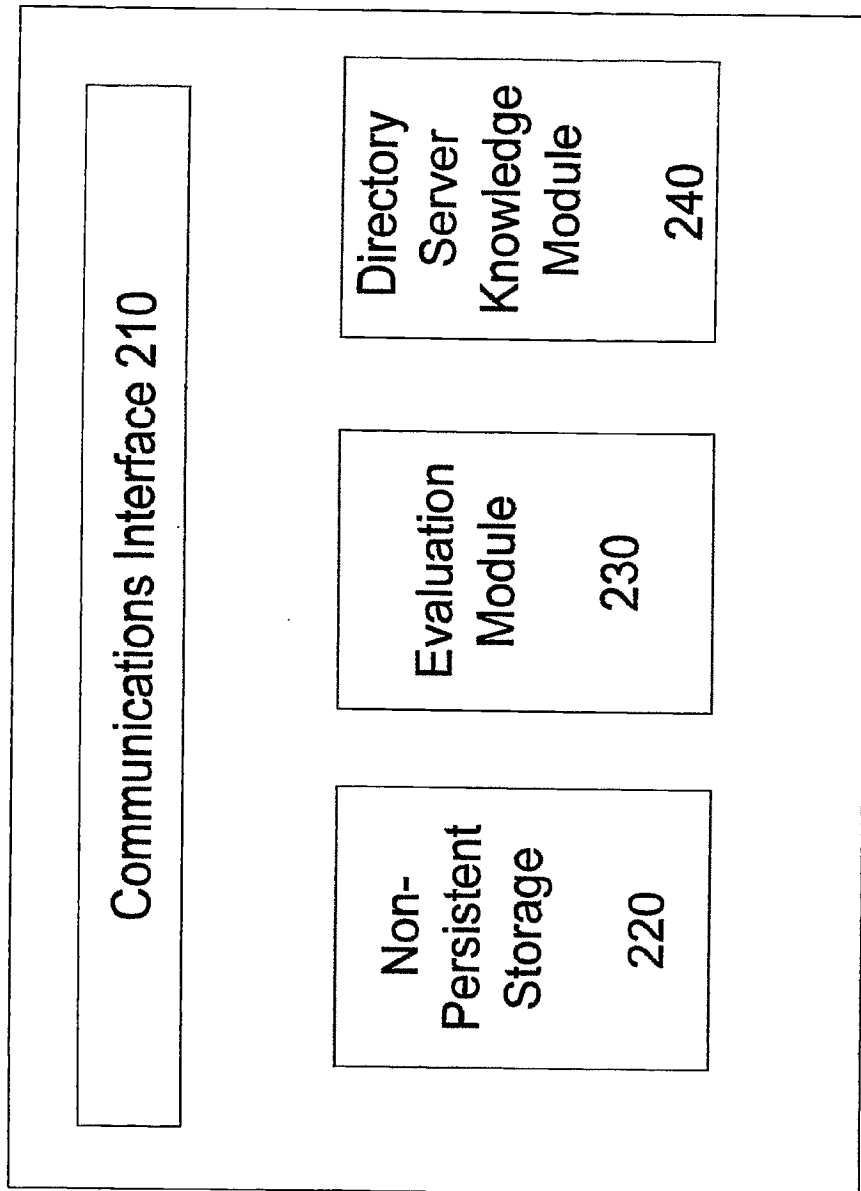


FIG. 2

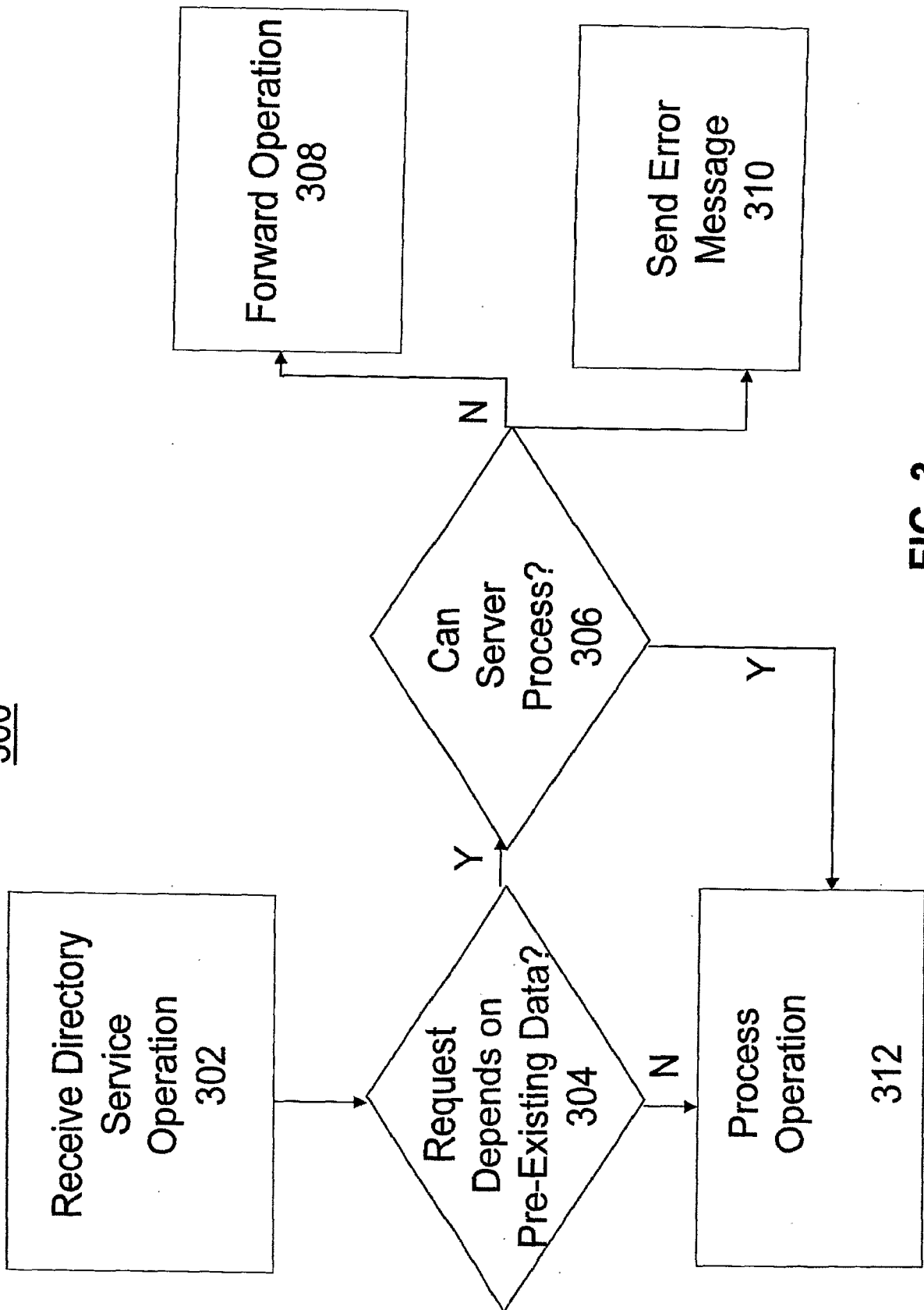
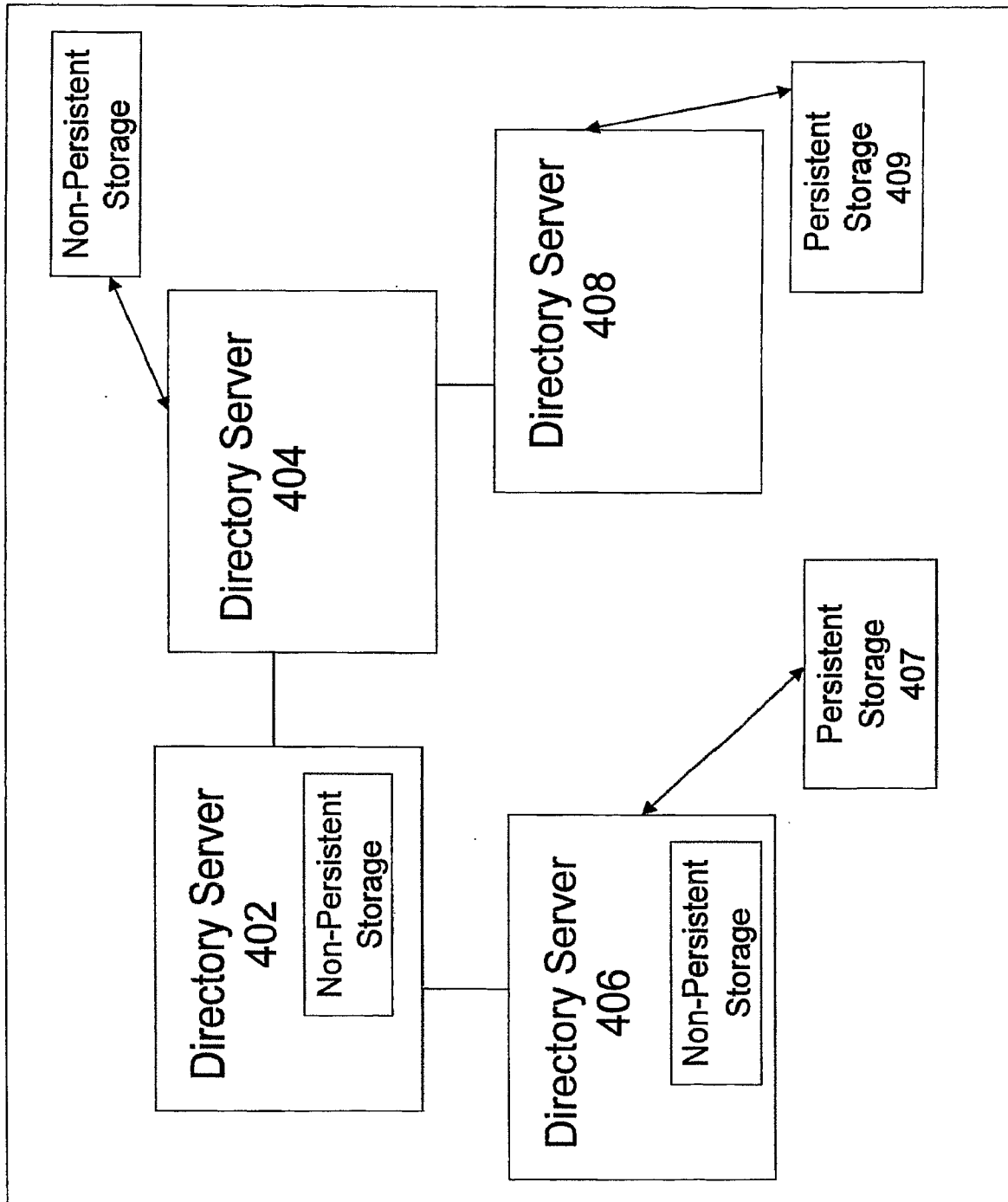
300

FIG. 3

FIG. 4



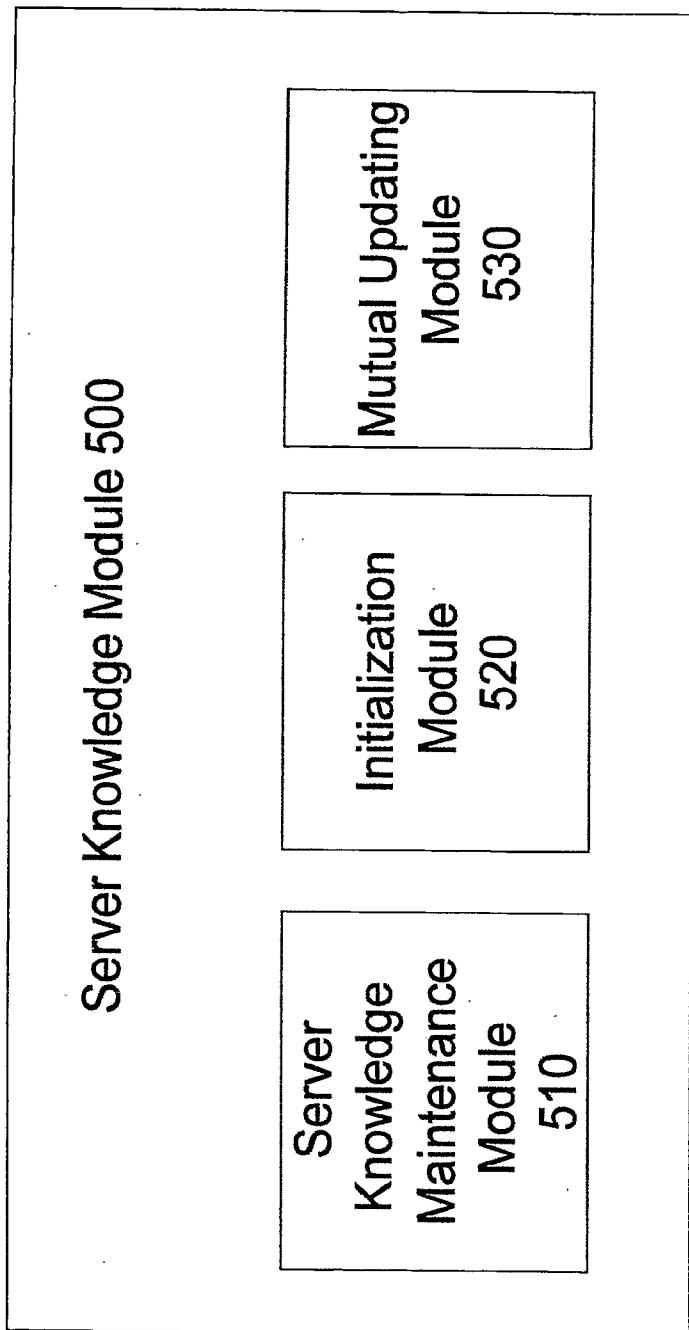


FIG. 5

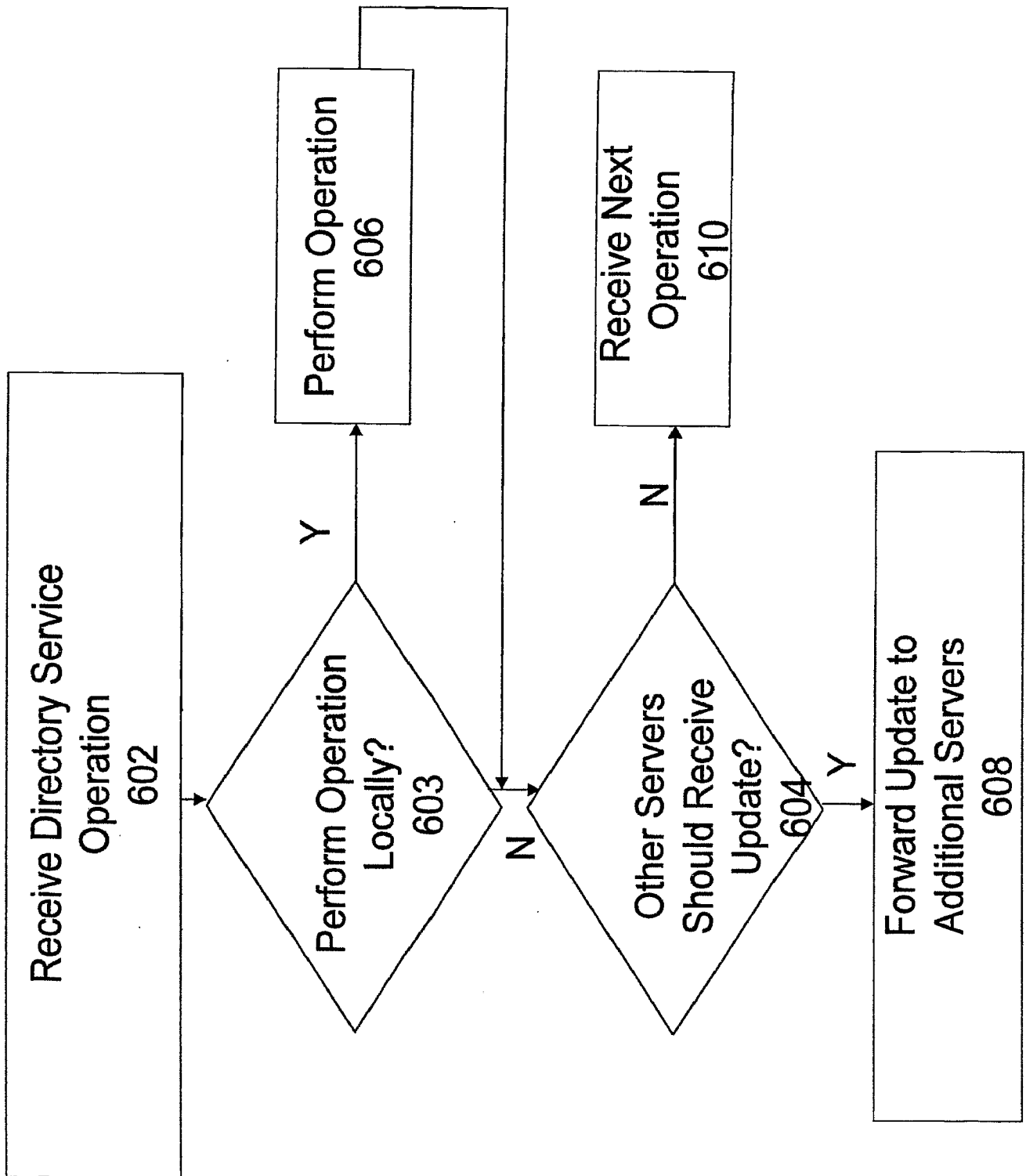


FIG. 6

700

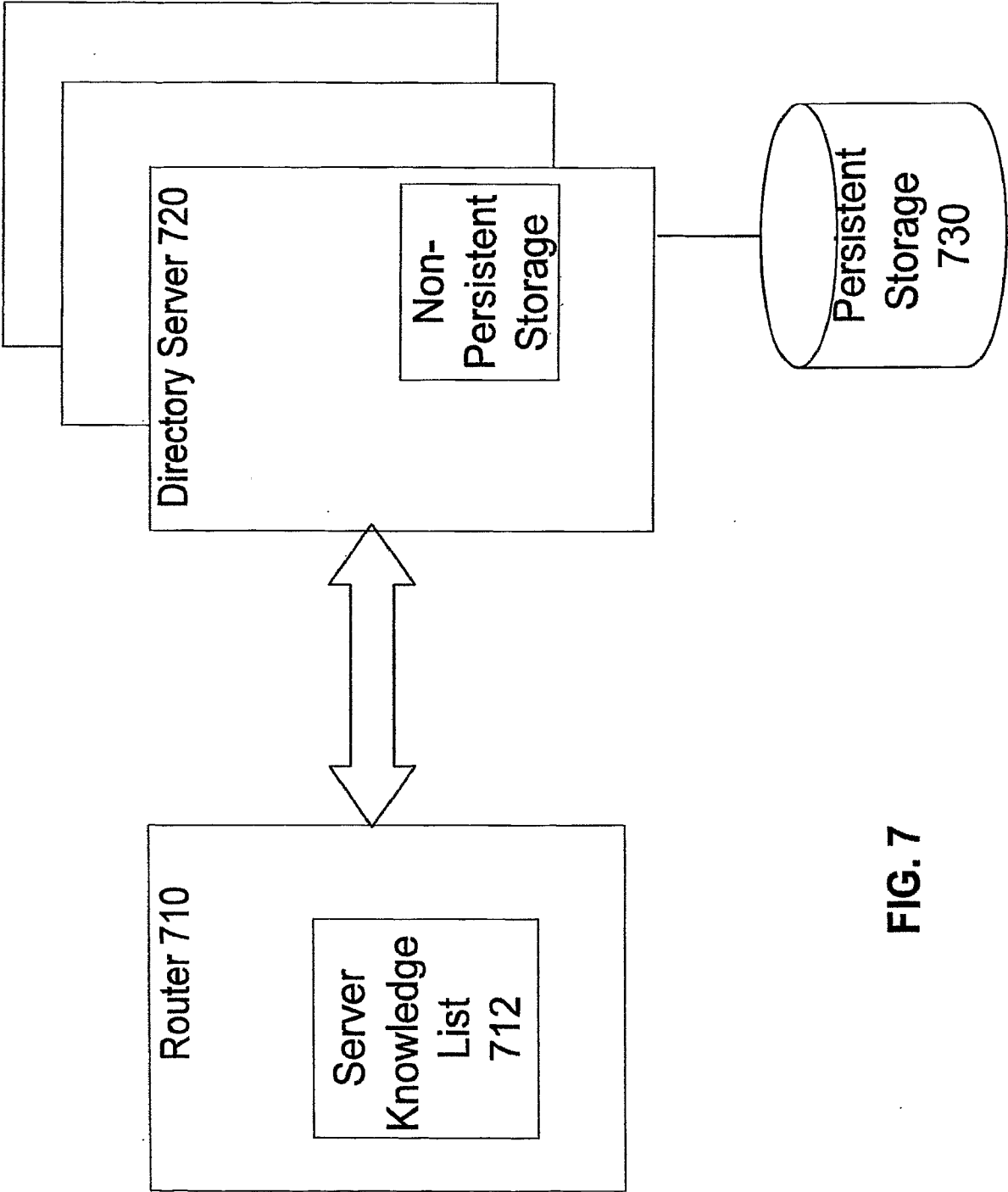


FIG. 7

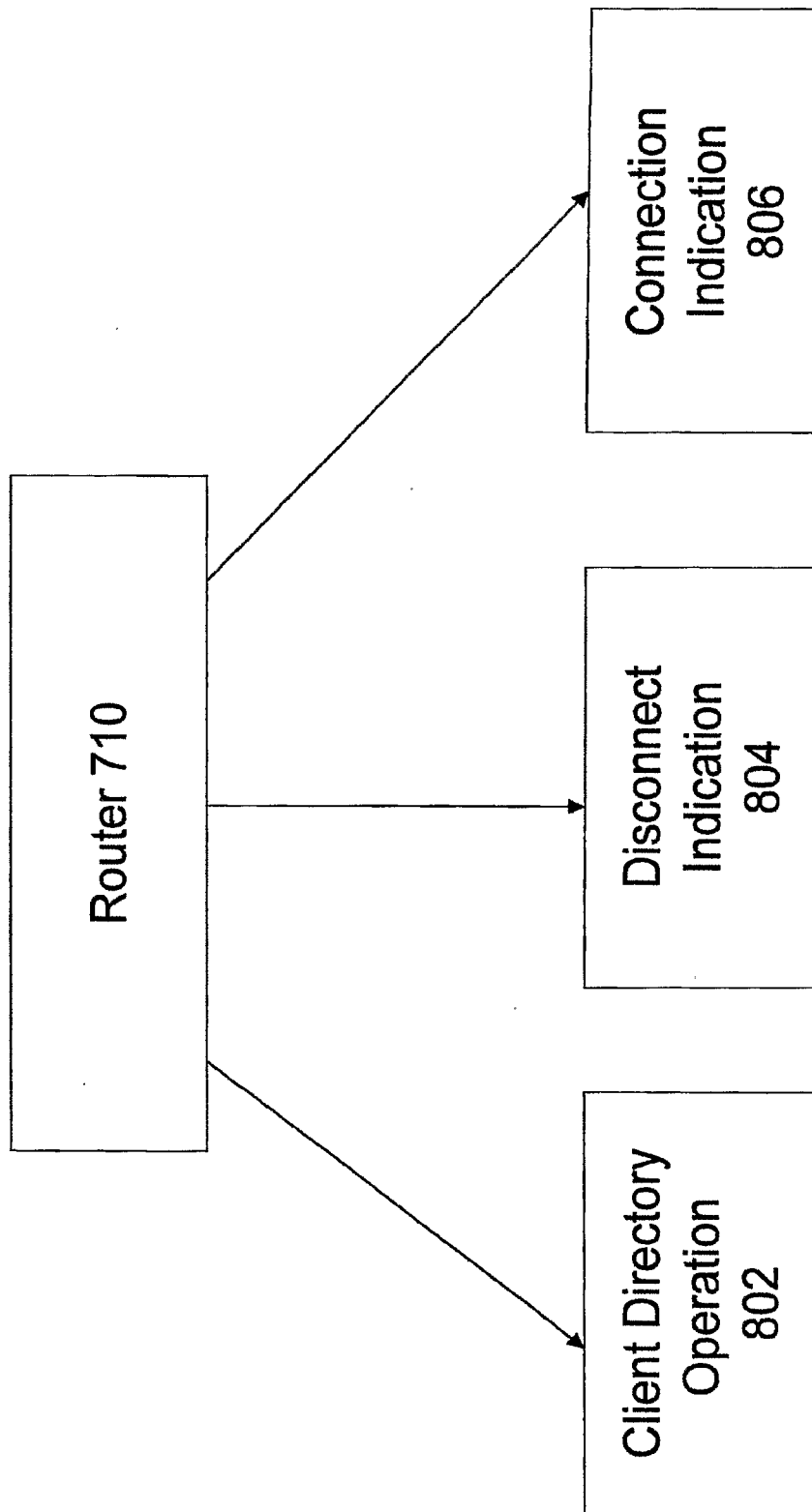
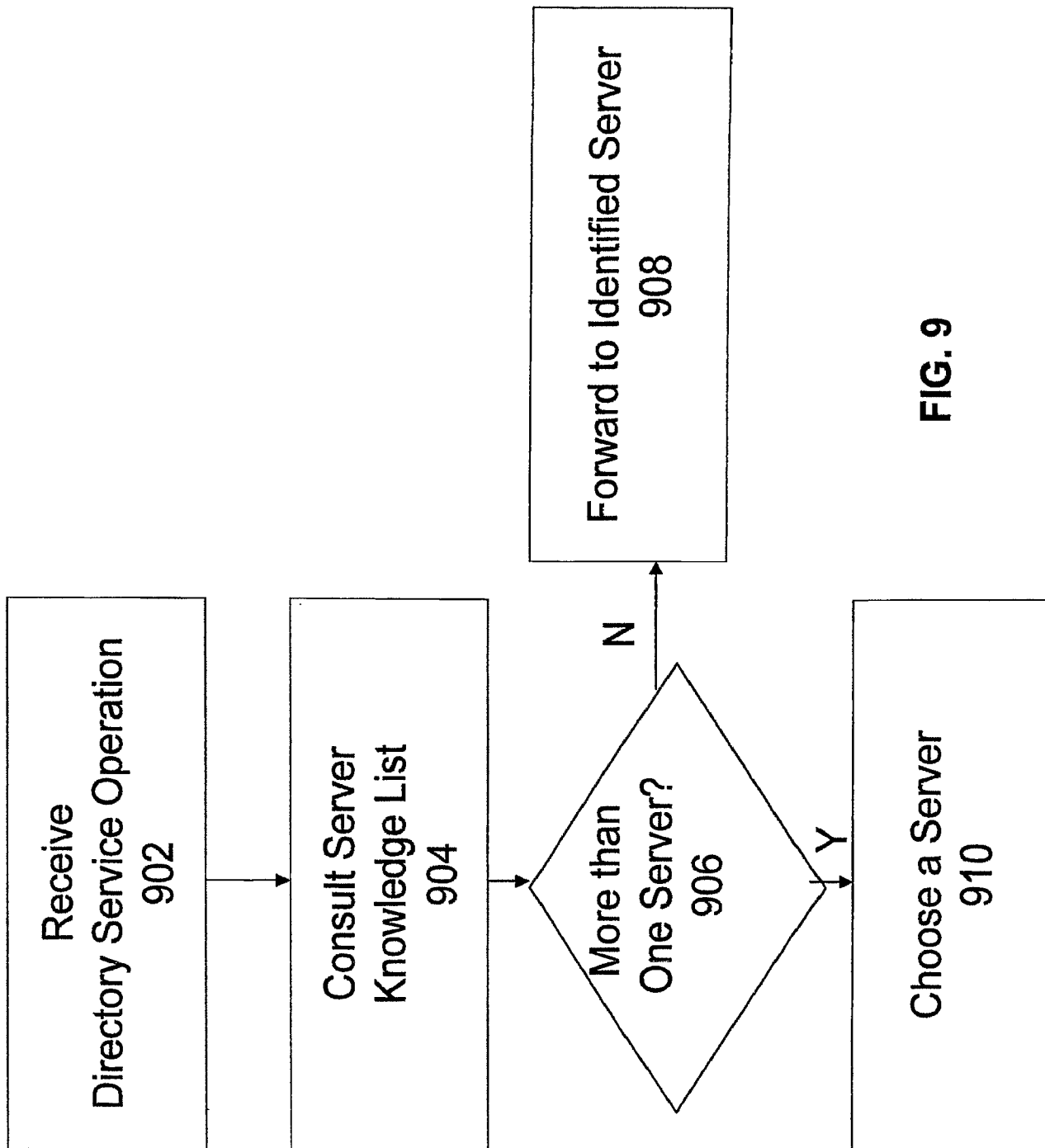


FIG. 8

**FIG. 9**