

PATENT APPLICATION

CLOUD DATA PERSISTENCE ENGINE

Inventor(s): Albert J. McGowan, a citizen of the United States, residing at:
4201 East Camelback Road #101
Phoenix, Ariz., 85018

Richard L. Carls, a citizen of the United States, residing at:
19793 West 84th Way
Scottsdale, Ariz., 85255

Assignee: Unicorn Media, Inc.
24 West 5th Street
Tempe, AZ 85281

Entity: Small

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000

CLOUD DATA PERSISTENCE ENGINE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to Australian Patent Application Serial No. 2010202782, filed July 1, 2010, entitled "CLOUD DATA PERSISTENCE ENGINE," which is
5 incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] This disclosure relates in general to cloud-based computer processing and, but not by way of limitation, to distribution, storage, and retrieval of data on a cloud networking arrangement.

10 [0003] In a typical data hosting architecture, multiple servers may be spread over a geographical area. The data stored on each of these servers may be the same (or substantially similar) data as that stored on each other server. A client attempting to access data hosted on such a system may communicate with a particular server, often the server closest to the client's physical location.

15 [0004] While such an arrangement may allow the client to quickly access all data stored on the server system, it may not be efficient for all data to be stored on each server. Clients in different locations (e.g., different cities, states, countries, and/or continents) may request data that varies significantly by location. For example, particular data may frequently be requested by clients in Europe, while the same data may be infrequently requested by clients in the Americas.

20 Therefore, if all of the data is maintained at each server, such a system may result in significant quantities of data being stored at servers where the data is infrequently, if at all, accessed. Accordingly, such a system may result in a substantial amount of data being stored at servers that wastes storage resources and is infrequently accessed.

25 BRIEF SUMMARY OF THE INVENTION

[0005] Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data
30 object stored locally, it may contact a second server to determine if it has the data object stored.

If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

5 **[0006]** In some embodiments, a cloud data persistence system for distribution of data is present. The system may include a first server which stores a first plurality of data objects and is communicatively coupled with a plurality of cache servers. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to one of a plurality of media assets. The plurality of media assets may be stored at a second location separate from
10 the first location. The system may include a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects. The second server may be one of the plurality of cache servers. The second server may be communicatively coupled with a first plurality of application servers. The system may include a third server that stores a third plurality of data
15 objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects. The third server may be one of the plurality of cache servers. The third server may be communicatively coupled with a second plurality of application servers. The system may include a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data object is
20 not present in the fourth plurality of data objects. The fourth server may be one of the first plurality of application server. The fourth server may be configured to receive a request from a first client for the data object. The system may also include a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects. The fifth server may be one of the
25 first plurality of application servers. The fifth server may be configured to receive a request from a second client for the data object. The system may further include a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects. The sixth server may be one of the second plurality of application servers. The sixth server may be configured to
30 receive a request from a third client for the data object. The system may include a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects. The seventh

server may be one of the second plurality of application servers. The seventh server may be configured to receive a request from a fourth client for the data object.

[0007] In some embodiments, a method for retrieving data objects using a cloud data persistence system configured for distribution of data is presented. The method may include receiving a first request from a client for a data object. The application center may be one of a plurality of application centers that communicate with a plurality of clients. The application center may be located at a first location. The data object may comprise a link to a media asset. The media asset may be stored at a second location separate from the first location. The method may further include determining the data object is not stored among a first plurality of data objects stored at the application center. The method may include transmitting a second request for the data object to a first cache server. The first cache server may be communicatively coupled with the plurality of application centers. The first cache server may be one of a plurality of cache servers. The method may include receiving the second request for the data object. The method may also include determining the data object is not stored among a second plurality of data objects stored at the cache server. The method may further include transmitting a third request for the data object to an origin server. The origin server may be communicatively coupled with the plurality of cache servers. The method may also include receiving the third request for the data object. The method may include locating the data object among a third plurality of data objects. The method may also include transmitting the data object linked to the third request to the cache server. The method may include receiving the data object linked to the third request. The method may include transmitting the data object linked to the second request. Also, the method may include receiving the data object linked to the second request. Further, the method may include transmitting the data object linked to the first request to the client.

[0008] In some embodiments, a cloud data persistence system for distribution of data is present. The system includes a first server which stores a first plurality of data objects and is communicatively coupled with a second server and a third server. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to a media asset of a plurality of media assets. The plurality of media assets may be located at a second location. The second location may be distinct from the first location. The system may include the second server, which may store a second plurality of data objects and may be configured to request a data object from the first server if the data object is not present in the

second plurality of data objects. The second server may be communicatively coupled with a fourth server and fifth server. The system may include the fourth server that stores a third plurality of data objects and may be configured to request the data object from the second server if the data object is not present in the third plurality of data objects. The fourth server may be configured to receive a request from a first client for the data object.

[0009] In some embodiments, the first plurality of data objects comprises all data objects present in the second plurality of data objects and the third plurality of data objects. In some embodiments, the third server stores a fourth plurality of data objects and is configured to request the data object from the first server if the data object is not present in the fourth plurality of data objects. The third server and the second server may be data persistence cache servers. The third server may be communicatively coupled with a sixth and seventh server. In some embodiments, the fifth server stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects. The fifth server and the fourth servers may be application servers. The fifth server may be configured to receive requests from a second client for the data object. The fifth server may be geographically separated from the fourth server. In some embodiments, the request from the first client for the data object comprises a universal resource indicator. In some embodiments, each of the first plurality of data objects comprise a universal resource locator and a title linked to a media asset of the plurality of media assets. Also, in some embodiments, data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on a frequency of requests for those data objects received from clients. In some embodiments, data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients. Further, in some embodiments, the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present disclosure is described in conjunction with the appended figures:

[0011] FIG. 1 illustrates a block diagram of a media servicing system.

[0012] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center
5 connected with application centers.

[0013] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.

[0014] FIG. 3 illustrates a block diagram of an embodiment of an application center.

[0015] FIG. 4A illustrates a block diagram of processes and objects utilized by a cloud-hosted
10 integrated multi-node pipelining system for media ingestion.

[0016] FIG. 4B illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery network.

[0017] FIG. 4C illustrates another block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery
15 network.

[0018] FIG. 5A illustrates a simplified block diagram of an embodiment of a cloud data persistence system.

[0019] FIG. 5B illustrates another simplified block diagram of an embodiment of a cloud data persistence system.

[0020] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to retrieve a data object from a cloud data persistence system.

[0021] FIG. 7 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to modify a data object on a cloud data persistence system.

[0022] FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management
25 and storage system.

[0023] FIG. 9 illustrates another simplified block diagram of an embodiment of an asset management and storage system.

[0024] FIG. 10 illustrates an embodiment of a method for retrieving data from an asset management and storage system.

5 [0025] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective
10 of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

[0026] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing
15 description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

[0027] As opposed to storing copies of a piece of data, such as a data object that contains
20 information linked to a media object, at each server on a system of multiple servers, it may be more efficient to store only data objects that are frequently and/or recently accessed by clients at particular servers. Such an arrangement, generally referred to as a cloud data persistence engine or system, may allow the data objects stored at each server to vary based on which data objects the server's clients are currently and/or have historically requested. Therefore, data objects
25 stored at a first server in a first location may include some of the same data objects and some different data objects than data objects stored at a second server in a second location (possibly located in a different city, country or on a different continent than the first server). Storing only data objects that are frequently and/or recently accessed by clients at each server may result in considerable efficiencies. For example, by storing only data objects that are frequently and/or
30 recently accessed by clients, the servers may require significantly less storage space because they do not need to store a potentially large amount of data objects that are infrequently accessed or

have not recently been accessed. A related benefit of this may be that by decreasing the amount of storage necessary at each server, the speed of access to the storage at the servers may be increased. For example, instead of relying on hard drives, solid state drives (which may cost significantly more per megabyte, but may offer improved access times) may be used for the storage at the servers.

[0028] While a server may infrequently receive a request for a particular data object from a client, the server may still need to be able to fulfill such a request. Therefore, if a server is contacted by a client for a data object that it does not have stored locally, the server may be able to contact another server, which may have the requested data object stored. Such a server may be configured to satisfy multiple servers with requests for such data objects.

[0029] While this second cache server may contain the data object requested by the client, it may also only store data objects that are frequently and/or recently (albeit less frequently and/or recently than the data stored at the first server) accessed by clients. If the second server does not have the data object requested by the client stored, the second server may have the ability to contact yet another server. This server, referred to as the data object origin server, may maintain a copy of all data objects stored on the system. Therefore, if the data object is present on the server system, the data object origin server is able to fulfill the request for the data. The data object origin server may then route the requested data to the client via the second server and the first server.

[0030] While the above arrangement may increase efficiencies when a client requests a data object, a client (perhaps the same or a different client) may request to modify (including, but not limited to changing, adding and/or deleting) a data object. In such instances, the modification to the data object (assuming the client has the appropriate access rights) may be routed to the data object origin server by a server receiving the request from the client. The modification may be applied to the relevant data at the data object origin server. The data object origin server may then send a notification to other servers, or only the cache servers that have the previous version of the data stored, that the previous version of the data is no longer valid. These servers may in turn send a notification to other associated servers, or only those servers that have the previous version of the data stored, that the previous version of the data object is no longer valid. This may prevent a client from receiving an out-of-date version of the data no matter which server is

accessed by the client. The modified data residing at the origin may then propagate out to the other servers as requests are made by clients to those servers for the data object.

[0031] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. **FIG. 1** is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user 140 (alternatively referred to as a user or a client) through a network such as the Internet 120. A media asset provided by a content provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on content delivery network (CDN) 150. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0032] The media servicing system further enables a content provider 130 or other entity to gather information regarding user behavior during media playback. For example, a content provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a content provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user 140 with a more desirable playback experience.

[0033] End user 140 can request a media asset to hear and/or see with a client program running on a personal computer or other electronic device connected to Internet 120. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the CDN 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator. During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed on a device operated by the end user 140 and/or indexing service from within the CHIMPS and/or CDN. The CHIMPS 150 can

subsequently provide the data and/or any analytics information derived from the data to the content provider 130.

[0034] FIG. 2A is a block diagram illustrating an embodiment of a kernel application 111-1 center connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0035] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more data object origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

[0036] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a content provider 130 for syndication. Such a service can allow a content provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a CDN and/or stored locally by storage array 230 and data object origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and data object origin server(s) 240.

[0037] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at

or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0038] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end users through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a content provider 130 through a content provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0039] FIG. 4A is a block diagram 400-1 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4A further indicates the physical systems in which my execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4A. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0040] Media can be ingested into the CHIMPS 110 when a content provider 130 uploads a media asset to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The

client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0041] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured to organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0042] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0043] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user who accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0044] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different

transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 425 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object updater 420. The data object updater 420, which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0045] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased.

[0046] FIG. 4B is a block diagram 400-2 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-2 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in place of media asset origin 455. Therefore, in block diagram 400-2, as opposed to the media assets being stored at media asset origin 455, they are stored at content data network 150, which may be operated and maintained by a third party.

[0047] FIG. 4C is a block diagram 400-3 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-3 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in conjunction with media asset origin 455. Here, in block diagram 400-3, media assets may reside at either media asset origin 455, content delivery network 150, or both.

[0048] Embodiments of such systems may include other systems that manage various requests from end users. One such system may be a cloud data persistence system. Referring to FIG. 5A, a simplified block diagram of an embodiment of a cloud data persistence system 500-1 is

illustrated. Such a system may allow for information, such as data objects, to be efficiently distributed to multiple of clients.

[0049] Some embodiments of cloud data persistence systems are used to distribute data objects that are linked to media assets. A data object may contain information pertinent to the data object it is linked to. For example, a data object may include details about a media asset and permissions for the media asset. More specifically, a data object may include fields containing: a title, keywords linked to the data object, the data object's (and/or associated media asset's) date of creation, the duration of the associated media asset, the file formats the associated media asset is available in, and what clients are permitted to access and/or edit the data object and/or the media asset. The data object may also contain a link to its associated media asset. Such a link may be in the form of a universal resource indicator, such as a universal resource locator. Other information may also be stored as a data object that is linked to a media asset. A media asset may represent many different forms of media, such as an audio file (in a variety of different formats), a video file, or a flash file, to name only a few examples.

[0050] Important to note, data objects may be stored separate from media assets. For example, a data object containing information linked to a particular media asset may not be stored in the same location as the media asset. For example, a media asset may reside on multiple servers that are part of a content delivery network, while the data object that contains information about the media asset is stored in some other location, possibly on some other network. Also, while the above discusses linked media assets, it should be understood that data objects may also be linked to other assets, files or data that do not constitute media. In such embodiments, the data objects may contain different fields of data.

[0051] Such a cloud data persistence system may be used to distribute data objects to one or several clients, but also may be used to distribute data objects to thousands of clients geographically disbursed worldwide. In the embodiment of cloud data persistence system 500-1, three levels of servers are present: a data object origin server 510, data persistence cache servers 520, and application centers 530. These application centers are in contact with one or more clients 540. It should be understood that each application center may include one or more servers. For example, an application center may be represented by application center 112 of FIG. 3. Alternatively, an application center may comprise a single server. While three levels of servers are present in the illustrated embodiment, it should be understood that greater or fewer

levels of servers may be present in other embodiments. Also, various servers may be integrated into the same server. For example, one server may serve as both an application center 530 and a data persistence cache server 520. It should also be understood that the terms of “application center,” “data persistence cache server,” and “data object origin server” are used to simplify the nomenclature used to describe the various levels of the cloud data persistence system 500-1, and these terms should not be interpreted as limiting the functionality of their respective identified servers.

[0052] Referring to cloud data persistence system 500-1, clients 540 make requests for data objects from application centers 530 via one or more networks. In the illustrated embodiment, only two clients (540-1 and 540-2) are present to maintain simplicity; many more clients (possibly representing many end users) may be possible. These clients may communicate with the same application center (530-1) or may communicate with different application centers (530-1 and 530-2). Further, because a client at one time interacts with a particular application center, at a different time, the same client may interact with a different application center. For example, if a particular application center is experiencing a heavy client load, possibly caused by a large number of clients attempting to access it, a client may be routed to another application center.

[0053] The client may make a request for a particular data object to the application center. This request may be made via a network, such as the Internet. Other networks may also be possible, such as a private intranet. The request for the data object may involve the client sending a universal resource indicator (URI) to an application center. A URI may take a variety of different formats. In particular, the URI may be in the form of a universal resource locator (URL). However, it should be understood that other forms of URIs may also be possible.

[0054] If client 540-1 transmits a URI to application center 530-1, application center 530-1 may have the data object linked to the URI stored locally. If this is the case, application center 530-1 may respond by sending the data object identified by the URI back to client 540-1. However, application center 530-1 may have only a subset of the data objects linked to valid URIs stored locally. This may be due to space constraints at application center 530-1. For example, if clients infrequently request particular data objects, it may be inefficient to distribute those data objects to each of application centers 530.

[0055] If the data object requested by client 540-1 is not present at application center 530-1, application center 530-1 may contact another server to retrieve it. Application center 530-1 may

be located in close proximity to data persistence cache server 520-1, or may be geographically separated from data persistence cache server 520-1. Application center 530-1 may communicate with data persistence cache server 520-1 directly (such as a direct cable connection) or may communicate using a network. This network may be a private network, such as a local area network or dedicated connection between the servers, or may use a public network, such as the Internet. Such a request from application center 530-1 to data persistence cache server 520-1 may include the URI received by application center 530-1 from client 540-1. In some embodiments, data persistence cache servers supply multiple application centers with requested data objects. In cloud data persistence system 500-1, data persistence cache server 520-1 serves two application centers (530-1 and 530-2), while data persistence cache server 520-2 serves two different application centers (530-3 and 530-4). It should be understood that a data persistence cache may serve a varying number of application centers. For example, a data persistence cache may serve 20 application centers.

[0056] It should also be recognized that the connections between application centers 530 and data persistence cache servers 520 are dynamic. If a particular data persistence cache is offline, busy, or otherwise unavailable, an application center may be able to route requests to a different data persistence cache server. For example, referring to cloud data persistence system 500-1, if data persistence cache 520 became unavailable, application center 530-1 may be able to route a request for the data object to data persistence cache server 520-2.

[0057] While, if application centers 530 and data persistence caches 520 communicate via a network (e.g., the Internet), the servers may be located anywhere a network connection is available worldwide. In some embodiments, a data persistence cache is located near a group of application centers. For example, if a region, such as New England, contains two application centers for each state, a single data persistence cache may be located in or near New England to serve these twelve scattered application centers. Having the data persistence cache located near a group of application centers may decrease communication latency when an application center and the data persistence cache communicate.

[0058] Upon receiving a request for a data object from application center 530-1, data persistence cache server 520-1 may determine if it has the data object stored. If it does, data persistence cache server 520-1 may send the data object to application center 530-1, which may, in turn, send the data object to client 540-1. Upon the application center receiving the data

object, it may store the data object to be able to satisfy requests from the same or a different client for the same data object at a future time. Application center 530-1 may make a determination as to whether the data object should be stored or not. For example, the determination may be based on the frequency it is receiving requests for the data object from clients 540. If the data object is frequently being requested by clients 540, application center 530-1 may store the data object. This may only occur if the data object is being requested more frequently than some other data objects. For example, only the 10,000 most frequently requested data objects may be stored by the application center. In some embodiments, application center 530-1 may store the most recently requested data objects. Therefore, this data object would be stored at application center 530-1, and possibly some other data object would no longer be stored at application center 530-1. In some embodiments, different application centers use different mechanisms to determine what data objects to store. Also, in some embodiments, some data objects are always stored at application centers 530. This may occur if a particular data object has been labeled a “high priority” (or some equivalent thereof).

[0059] However, if data persistence cache server 520-1 determines that it does not have the data object requested by client 540-1 stored, data persistence cache server 520-1 may contact another server. Data persistence cache server 520-1 may contact another server called a data object origin server 510. Data object origin server 510 may supply data persistence cache servers 520 with various data objects. In cloud data persistence system 500-1, data object origin server 510 is shown as communicating with two data persistence cache servers 520. However, it should be understood that data object origin server 510 may communicate with one, three, or some other number of data persistence cache servers. For example, data object origin server 510 may communicate with more, such as 20 data persistence cache servers.

[0060] Data object origin server 510 may be physically located with one or more data persistence cache servers. Data object origin server 510 may communicate directly, such as via a direct connection, with data persistence cache servers. For example, in cloud data persistence system 500-1, data object origin server 510 communicates directly with the data persistence cache server 520-2. Data object origin server 510 may also communicate via any of the previously mentioned networks with a data persistence cache server. Again, referring to cloud data persistence system 500-1, data object origin server 510 communicates with data persistence cache server 520-1 via a network, possibly the Internet.

[0061] While application centers 530 and data persistence cache servers 520 may store only a subset of the various data objects that clients 540 may request, data object origin server 510 may store all data objects that exist on system 500-1. In other words, if a data object requested by a client is not present on data object origin server 510, the data object may not exist on system 500-1.

[0062] Just as a data persistence cache server may be physically located near a group of application centers, data object origin server 510 may be located near the group of data persistence cache servers 520. Returning to the previous regional example, if a data persistence cache server is located in New England, another may be located near New York City, another near Washington DC, and perhaps yet another near Chicago. If one data object origin server is servicing these data persistence cache servers, it may be located in a central location to serve them all. In some embodiments, the data object origin server may be co-located with one or more data persistence cache servers. Alternatively, a data object origin server may be located at a separate physical location from any other database servers.

[0063] A request from the data persistence cache server 520-1 to data object origin server 510 may comprise a URI. This may be the same URI that was sent from client 540-1 to application center 530-1, then from application center 530-1 to data persistence cache server 520-1. In some embodiments, the URI transmitted from data persistence cache 520-1 may be in a different format than the other URIs. Upon receiving the URI, data object origin server 510 may locate the data object the URI is linked to. The data object origin 510 may then transmit the data object to data persistence cache server 520-1. The data persistence cache server 520-1 may make determination as to whether to store the data objects as previously described in relation to application center 530-1. For example, the determination of whether to store the data object may be based on the frequency with which application centers 530 request the data object from data persistence cache server 520-1. In some embodiments, data persistence cache server 520-1 stores the most recently requested data objects. Data persistence cache server 520-1 may then transmit the data object to application center 530-1. Application center 530-1 may then transmit the data object to client 540-1.

[0064] If the data object was first found at application center 530-1, the latency between client 540-1's request and client 540-1 receiving the data object may be the shortest. If instead the data object is first found at data persistence cache server 520-1, the latency may be longer. Finally, if

the data object is only found at data object origin 510, the latency may be the longest. Regardless of where the data object is found, client 540-1 may be unaware of whether application center 530-1, a data persistence cache server 520-1, or the data object origin server 510-1, had the data object stored.

5 **[0065]** While only one data object origin server 510 is present in cloud data persistence system 500-1, more than one data object origin server is possible. For example, a data object origin server may serve a particular region, such as a country. Other data object origin servers may be present in other countries. These data object origin servers may each maintain identical or substantially similar data sets. Therefore, a data object found on one data object origin server
10 would also be found on other data object origin servers. While not illustrated in FIG. 5, it should be understood that data object origin server 510 may be in communication with a kernel server and/or a data object updater server, such as illustrated in FIGs. 2A, 2B, and 4. Such an arrangement may allow multiple different data object origin servers to have the same data objects stored.

15 **[0066]** While the above deals with the communication flow in cloud data persistence system 500-1 when a client requests a data object, a client (or, of course, an end user acting through the client) may also modify a data object. Depending on the data object, a request for a particular data object may be much more frequent than a request to modify a data object. By way of example only, consider a video clip posted on an online video sharing website. If the video is
20 popular, data object information, such as the video owner's name and the video's title may be sent to millions of different clients. However, the end user who created the video may only infrequently modify (via a client) data linked to the video.

[0067] In cloud data persistence system 500-1, the request to modify a data object, as opposed to a request to retrieve a data object, may be treated differently. If client 540-1 makes a request
25 to application center 530-1 to modify a data object, application center 530-1 (assuming the end user and/or client has the proper permissions to modify the data object) may route the request to Kernel application center 111. Kernel application center 111 may then update data object origin server 510, and any other data object origin servers requiring updating. Application center 530-1 may not modify or delete the previous version of the data object if it is stored at application
30 center 530. The data object may then be updated at data object origin server 510. While the data object may now be updated at data object origin server 510, other application centers and data

persistence cache servers, such as application center 530-3 and data persistence cache 520-1, may still have out-of-date versions of the data object. Therefore, upon receiving the modifications to the data object, kernel application center 111 may notify data persistence cache servers 520 to no longer use their current version of the data object. This may involve data persistence cache servers 520 deleting their out-of-date version of the data object. In turn, data persistence cache servers 520 may notify application centers 530 to similarly no longer use their out-of-date versions of the data object. This may again involve the out-of-date data object being deleted from application centers 530. Application centers 530 and data persistence cache servers 520 may not receive a new version of the data object along with the notification. Rather, application centers 530 and data persistence cache servers 520 may only receive the new version of the data object if a client requests the data object and it is routed through the data persistence cache server and the application center.

[0068] If kernel application center 111 knows what particular data objects are stored at data persistence cache servers 520, a notification may only be sent to the data persistence cache server known to store the data object. Similarly, if data persistence cache servers 520 know what data objects are stored at application centers 530, only those application centers that have out-of-date versions of the data object may be sent the notification. In some embodiments, whenever a modification of a data object is received at kernel application center 111, all data persistence cache servers 520 are notified, which in turn notify all application centers 530. In such embodiments, therefore, kernel application center 111 does not need to be aware of what particular data objects are stored at data persistence cache servers 520, and data persistence cache servers 520 do not need to be aware of what data objects are stored at application centers 530. In some embodiments, Kernel application center 111 notifies both data persistence cache servers 520 and application centers 530.

[0069] Referring to **FIG. 5B**, a simplified block diagram of an embodiment of a cloud data persistence system 500-2 is illustrated. Such a system may be substantially similar to system 500-1. However, system 500-2 illustrates how various components of system 500-2 may be geographically separated. It should be understood that each component of system 500-2 may be located at a different physical location. In some embodiments, client 540-1 is at location 551 while client 540-2 is at location 552. Locations 551 and 552 may be close together, possibly only several miles if both client 540-1 and client 540-2 are using the same application center.

Clients 540-1 and 540-2 may be a great distance apart if they are using different application centers.

[0070] Application centers 530-1, 530-2, 530-3, and 530-4 are shown as different locations of 553, 554, 555, and 556, respectively. Application centers that use the same data persistence cache server may be closer together than application centers that do not use the same data persistence cache servers. Further, as those with skill in the art will recognize, while an application server may not be physically closer to a data persistence cache server, due to the state of the network, communication between two physically further apart servers may be quicker. In such a case, the further servers may be in communication as opposed to the closer servers.

10 [0071] Data persistence cache server 520-1 is at another location 557. Location 557 may be near or in between location 553 and 554. Data persistence cache server 520-2 is at the same location 555 as application center 530-3. In some embodiments, data persistence cache server 520-2 is at some other location from application center 530-3. Finally, data object origin server 510 may be present at location 558. This location may be at the kernel application center 111, or
15 may be separate.

[0072] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method 600 for responding to a client request to retrieve a data object. Such a method may use a cloud data persistence system, such as cloud data persistence system 500-1 of FIG. 5A. Alternatively, method 600 may rely on some other cloud data persistence system. The embodiment of FIG. 6
20 presupposes that the data object requested is not present on either the application center contacted or the cache server contacted. As described in relation to FIG. 5, if one of those servers has the data object stored, the method would differ.

[0073] At block 605, a request may be sent from a client to an application center. The client may be client 540-1 or 540-2 of FIG. 5, or may represent some other client. Similarly, the
25 application center of FIG. 6 may represent any of application centers 530 of FIG. 5, or may represent some other application center. The request sent by the client at block 605 of FIG. 6 may contain a URI, possibly a URL, indicating the data object the client is requesting to retrieve. The request may be received either directly or via a network (such as the Internet), by the application center at block 610.

[0074] At block 615, the application center may determine if the data object requested by the client is present. This may involve the application center searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the application center may search an index of all the data objects it has stored. In this embodiment, the application center does not have the data object stored locally. Therefore, at block 620, a request is sent from the application center to a data persistence cache server. This request may contain the same URI received in the request from the client at block 610. The cache server contacted may be a data persistence cache server 520 of FIG. 5A or may represent some other data persistence cache server. The request by the application center may be received by the data persistence cache server either directly or via a network (such as the Internet) at block 625.

[0075] At block 630, the data persistence cache server may determine if the data object requested by the application center is present. This may involve the data persistence cache server searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the data persistence cache server may search an index of all the data objects it has stored. In this embodiment, the data persistence cache server does not have the data object stored locally. Therefore, at block 635, a request is sent from the data persistence cache server to an data object origin server. This request may contain the same URI received in the request from the client at block 610 and the application center at block 625. The data object origin server contacted may be the data object origin server 510 of FIG. 5A or may represent some other data object origin server. The request by the data persistence cache server may be received by the object origin server either directly or via a network (such as the Internet) at block 640.

[0076] At block 645, the data object origin server may locate the data object requested. Assuming that the URI indicating the data object initially requested by the client is valid, the object origin server should have the data object stored. If the data object is not present at the data object origin server, the data object may be assumed to not exist or that the URI submitted by the client is incorrect. To retrieve the data object, the data object origin server may search one or more local (or remote) hard drives or solid state drives for the data object. In some embodiments, the data object origin server uses an index to locate and retrieve the data object. Once the data object has been located, at block 650 it may be transmitted to the data persistence cache server.

[0077] At block 655, the data persistence cache server may receive the data object. The data received from the data object origin server may also include the URI sent in the request at block 635. At block 660, the data persistence cache server may determine whether the data object should be stored locally. If the data persistence cache server stores the most recent data objects requested, the data persistence cache server may always store the data object when it is received from the data object origin server. If the data persistence cache server stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested more frequently. As those with skill in the art will understand, there are other factors that may be used to determine whether the data object is stored locally at the data persistence cache server. As another example, the decision as to whether to store the data object may be based partially on frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored or not, it may be transmitted to the application center at block 665.

[0078] At block 670, the application center may receive the data object. This may be the same application center that received the request at block 610. The data received from the data persistence cache server may also include the URI sent in the request by the application center at block 620. At block 675, the application center may determine whether the data object should be stored locally. The factors used to consider whether to store the data object may be the same or different from the factors used at the data persistence cache server. Again, if the application center stores the most recent data objects requested, the application center may always store data objects received from the data persistence cache server. If the application center stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested by clients more frequently. As those with skill in the art will understand, there are other factors that may be used to determine whether the data object is stored locally at the application center. As another example, the decision as to whether to store the data object may be based partially on the frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored by the application center or not, it may be transmitted to the client at block 680. Further, to be clear, while it is possible that the same data objects may be stored at both the application and the data persistence cache server, it is likely that at least some different data objects would be stored at the application center than the data persistence cache server. Also, the amount of storage available at the application center and the data persistence cache server may vary. For example, the data persistence cache server may

have 50% more storage space than an application center; therefore, it may store roughly 50% more data objects.

[0079] At block 685, the client may receive the data object it requested at block 605. The client may be unaware as to whether the data object was present at the application center, the data persistence cache server, or the data origin cache server. To the client, it may appear as if the client is only interacting with the application center.

[0080] While FIG. 6 illustrates an embodiment of a method of a client requesting a data object, assuming the associated end user and/or client has the proper permissions, the end user and/or client may also request to modify a data object. FIG. 7 illustrates a swim diagram of a simplified embodiment of a method 700 of a client modifying a data object. Method 700 may be conducted using a cloud data persistence system, such as cloud data persistence system 700 of FIG. 7. Alternatively, some other cloud data persistence system may be used. A modification may involve changing data within a data object, adding data to a data object, or deleting the data object, to name only a few examples. It should be understood that in some embodiments, modifications to data objects are much less frequent than requests to retrieve a data object. At block 705, the client may transmit the request to modify the data object to an application center. The client may be the same client as detailed in FIG. 6 or client 540-1 or 540-2 of FIG. 5. Alternatively, the client of FIG. 7 may represent some other client. Similarly, the application center of FIG. 7 may represent the application center of FIG. 6 or any of application centers 530 of FIG. 5. The application center of FIG. 7 may also represent some other application center.

[0081] At block 710, the application center may receive the modification request from the client. The request may include a URI (such as a URL) indicating the data object the client is attempting to modify. The request may also include the changes, additional information, or an indication to delete the data object. If the application center determines it has the data object stored, at block 712, the application center may delete or block access to the data object so that other clients do not receive an out-of-date version of the data object. At block 715, a modification request may be transmitted from the application center to the data object origin server. This transmission may be routed from the application to the data object origin server (e.g., not through the data persistence cache server) or may be relayed by the data persistence cache server.

[0082] At block 720, the kernel application center may receive the modification request. The kernel application center may be the kernel application center 111 of FIG. 5A and FIG. 5B.

Kernel application center of FIG. 7 may also represent a data object origin server. After

receiving the modification request at block 720, the kernel application center may modify the

5 data object at one or more data object origin servers at block 725. Data object origin servers may always have a stored version of the data object because they maintain a stored version of all valid data objects. If the data object origin server does not have a version of the data object requested to modify, it may be assumed that the data object does not exist or the URI sent by the client at block 705 was incorrect. The modification at block 725 may involve the data object being
10 changed, having data added to it, or the data object being deleted. At block 730, the modified data object may be stored at one or more data object origin servers. Obviously, if the data object is being deleted, this step may not occur.

[0083] At block 735, a notification may be sent by the kernel application center that indicates that the data object has been modified. Of note, this modification may not contain the modified

15 data object, but rather may contain only information that indicates that previous versions of the data object are no longer valid. This notification may be transmitted to one or more data persistence cache servers at block 735. While the swim diagram illustrates the notification being sent to only one data persistence cache server, this is for simplicity only. The notification transmitted by the data object origin server may be sent to all data persistence cache servers (or
20 those data persistence cache servers that have the previous version of the data object stored) in communication with the data object origin server. The notification may only be sent to those data persistence cache servers that have an out-of-date version of the data object if the kernel application center is aware of which data persistence cache servers have the data object stored. This may be possible if the data persistence cache server informs the kernel application center of
25 its stored data objects. Alternatively, all data persistence caches may receive the notification whether they have the previous version of the data object stored or not. If a data persistence cache server does not have the previous version of the data object stored, the data persistence cache server may ignore the notification.

[0084] At block 740, the notification may be received by the data persistence cache server.

30 The data persistence cache server may be the data persistence cache server of FIG. 6 and/or one of the data persistence cache servers 520 of FIG. 5. The data persistence cache server of FIG. 7 may represent some other data persistence cache server. At block 745, the previous version of

the data object may be cleared (e.g., deleted or otherwise removed) from the data persistence cache server. At block 750, a notification that the previous version of the data object is out-of-date may be transmitted to one or more application centers at block 750.

[0085] While the swim diagram illustrates the notification being sent to only one application center, this is for simplicity only. The notification transmitted by the data persistence cache server may be sent to all application centers (or those application centers that have the previous version of the data object stored) in communication with the data persistence cache server. The notification may only be sent to those application centers that have the previous version of the data object if the data persistence cache server is aware of which application centers have the previous version of the data object stored. This may be possible if the application center informs the data persistence cache server (or the kernel application center and/or data object origin server) of its stored data objects. Alternatively, all application centers may receive the notification whether they have the previous version of the data object stored or not. If an application center does not have the previous version of the data object stored, the application center may ignore a received notification. The notification sent by the data persistence cache server at block 740 may represent the same or a similar notification as sent by the kernel application center at block 735. For example, the notification may include a URI (possibly a URL) indicating the data object that has been modified. At block 760, the previous version of the data object may be cleared (e.g., deleted or otherwise demarked) from the application center.

[0086] While the previous description and figures relate to how a client may retrieve and modify a data object, the storage of data objects and other data and/or files may be more sophisticated than one server (e.g., an application center, a data persistence cache server, or a data object origin server) having one or more storage devices. For example, each application center of FIG. 5A may represent a cluster of servers, each connected to one or more storage devices, such as hard drives or solid state drives. FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management and storage system 800 showing how various storage devices may be used to store data and accessed. While such a system may be used to store data objects, such as those data objects described previously, such a system may be used to store other forms of electronic data, such as media assets. A media asset may be a media file, such as a movie file, audio file, or image file.

[0087] In system 800, multiple different storage devices 810 may be present. These storage systems may be servers having storage devices such as hard drives and/or solid state drives. Storage devices 810-1 and 810-2 may be located in the same location or may be physically in different locations. Each of storage devices 810 may be linked to one or more storage servers 820. These storage servers 820 may not be physical servers, but rather logical abstractions of servers. Alternatively, they may be physically separate servers. Various storage drives on storage servers 820 may be mapped to different storage devices. For example, storage server 820-1 is linked to one storage drive. The storage drive is present on storage device 810-1. Storage server 820-1 and storage device 810-1 may be directly connected, or may be connected via a network, such as the Internet or a private corporate intranet. In system 800, storage server 820-2 has two storage drives. A first storage drive of storage server 820-2 maps to storage device 810-1. Therefore, in the embodiment of the system 800, storage device 810-1 is mapped to two different storage servers. If, for example, storage device 810-1 has storage totaling 1 TB, this gross storage space may be divided among the two drives mapped to it. For example, the storage drive of storage server 820-1 mapped to storage device 810-1 may be 300 GB. And, the storage drive of storage device 820-2 mapped to storage device 810-1 may be 700 GB. Storage server 820-2 may also have another storage drive. The storage drive is mapped to storage device 810-2. In this embodiment, storage device 810-2 is only mapped to storage server 820-2. Therefore, if storage device 810-2 has total storage of 2 TB, this entire space may be mapped to the linked storage drive of storage server 820-2. Or, as another example, only 800 GB of storage device 810-2 may be mapped to storage server 820-2, with the remaining space being free space.

[0088] Also, different storage devices may require communication, such as requests for media assets, to use varying protocols. For example, storage device 810-1 may be configured to accept file requests via FTP. However, storage device 810-2 may use MICROSOFT WINDOWS's file copy. It should also be understood that other protocols may also be used. Therefore, depending on what protocol a storage device accepts, the storage server links to that storage device may be configured to communicate in the proper protocol. This may be especially useful if the entity operating the storage servers 820 does not control one or more of storage devices 810. For example, storage device 810-2 may be maintained by some other entity, such as a client of the entity operating system 800.

[0089] Other logical abstractions that may be present are storage systems 830. Each storage system, such as storage system 830-1, may contain one or more than one storage servers 820. In

system 800, storage system 830-1 includes two storage servers: 820-1 and 820-2. Therefore, a request for a particular piece of data, such as a media asset, received at storage system 830-1, may not indicate the storage server or the storage device where the media asset is located.

However, upon receiving the request (possibly in the form of a URL) at storage system 830-1 for the media asset, it may be able to determine the appropriate storage server to contact. In turn, this storage server may be able to determine the appropriate storage drive (linked to a storage device) that contains the media asset. The logical abstractions of the storage system 830 and storage servers 820 may reside on the same physical servers as storage devices 810. In some embodiments, storage server 820 and storage system 830 are present on different servers than storage device 810. Also, storage systems 830 may be physically separate servers from storage servers 820.

[0090] Yet another logical abstraction may be present as storage cloud 840. Storage cloud 840 may include one or more storage systems. Storage cloud 840 may be viewed as the entire storage system. In system 800, storage cloud 840 includes storage systems: 830-1 and 830-2.

Therefore, storage cloud 840 may contain all storage systems 830 in system 800. In turn, storage systems 830 may contain all storage servers 820 in system 800. Further, storage server 820 may be linked to all storage devices 810 in system 800.

[0091] FIG. 9 illustrates another simplified block diagram of an embodiment of a system 900 showing how various storage devices may be accessed. System 900 of FIG. 9 may represent the same system as system 800 of FIG. 8, or may represent a different system. In the embodiment illustrated, system 900 contains four storage devices: 910-1, 910-2, 910-3, and 910-4. These storage devices may represent any of storage devices 810 of FIG. 8. Also, these storage devices may represent different storage devices. As will be understood to those with skill in the art, fewer or more storage devices may also be possible. In the illustrated embodiment, storage device 910-1 and storage device 910-2 are illustrated as being operated by first entity 915. First entity 915 may be the entity that operates other portions of system 900, or is a client of the entity that operates other portions of system 900. These storage devices, 910-1 and 910-2, may be co-located with one or more storage servers 920, or may be geographically separated. Further, storage device 910-1 and storage device 910-2 may be present at the same location, or may be geographically separated. Storage device 910-3 may be operated by a second entity 916. In some embodiments, the second entity 916 may be a client of the first entity. In system 900, storage device 910-3 may communicate with other systems 917 of the second entity 916.

Therefore, while storage servers 920 may be able to access storage device 910-3, client system 917 may also access and/or modify data stored on storage device 910-3. Storage device 910-4 is illustrated as part of a third entity 918. This entity may be an entity that sells or leases storage on their system. While FIG. 9 illustrates four storage devices operated by various entities, those skilled in the art will recognize that this arrangement is merely for exemplary purposes. Many other arrangements of storage devices, operated by one or more entities may be possible.

[0092] Whether operated by three different entities, as illustrated, or by one entity, storage devices 910 may communicate with multiple storage servers 920. System 900 includes four storage servers: 920-1, 920-2, 920-3, and 920-4. The storage servers 920, which may be logical abstractions, may communicate with the various storage devices 910 via a direct connection (such as 920-1 to storage device 910-1) or via a network (such as 920-2 to storage device 910-3). The network may be a network such as the Internet or a private corporate intranet. One storage server may be mapped to one storage device, for example, in FIG. 9, storage server 920-1 only has one storage drive mapped to storage device 910-1. A storage device may also be mapped to multiple storage servers, for example, in FIG. 9, storage servers 920-2 and storage server 920-4 are both mapped to storage device 910-3. Also, one storage server may be mapped to multiple storage devices. For example, storage server 920-4 is mapped to storage device 910-3 and storage device and 910-4.

[0093] Also, storage servers 920 may communicate with storage devices 910 via a variety of different protocols. For example, the format of requests to some storage devices may be FTP, while others may be MICROSOFT WINDOWS file copy. Other protocols may also be possible. Therefore, any request for data or a media asset that is received by storage servers 920 does not need to indicate the appropriate protocol to retrieve the file or media asset from the appropriate storage device.

[0094] As in FIG. 8, one or more than one storage servers may be mapped to a storage system 930. Storage systems 930 may represent the same storage systems as storage systems 830 of FIG. 8. Alternatively, storage systems 930 may represent some other storage systems. In system 900 of FIG. 9, storage servers 920-1 and storage server 920-2 are mapped to storage system 930-1. Storage system 930-1 may communicate directly with the storage server (such as the connection between storage system 930-1 and storage server 920-1) or may use a network (such as the connection between storage system 930-1 and storage server 920-2). In system 900,

storage system 930-2 may communicate with two storage servers: 920-3 and 920-4. While in system 900 storage systems 930 are shown as communicating with two storage servers each, those with skill in the art will recognize that a storage system may also communicate with one or more than two storage servers.

5 [0095] Storage systems 930 may make up storage cloud 940. Storage cloud 940 may represent the same storage cloud as storage cloud 840 of FIG. 8. Alternatively, storage cloud 940 may represent some other storage cloud. Therefore, storage cloud 940 may be viewed as including storage systems 930, storage servers 920, and storage devices 910.

[0096] Other systems may interact with various components of storage cloud 940. For
10 example, a content data network 960 may communicate with storage system 930-1. If the client requests the media asset from content data network 960 (CDN) that the CDN does not have stored (a cache miss), the content data network 960 may request the media asset from storage system 930-1. Storage system 930-1 may then contact the appropriate storage server 920, which may, in turn, retrieve the media asset via the correct protocol from the appropriate storage
15 device. From the viewpoint of CDN 960, it may appear that the media asset is stored at storage system 930-1. While CDN 960 is shown only interacting with storage system 930-1, it should be understood that this is for simplicity only. Multiple CDN's may communicate with storage system 930-1. Further, CDN's may also communicate with storage system 930-2.

[0097] While CDN 960 may communicate with storage system 930-1, file replication service
20 950 may communicate directly with storage servers 920. File replication service 950 may be responsible for distributing various media assets to storage server 920 that need to be stored for possible future retrieval. While file replication service 950 is shown as connected to only storage server 920-1, it should be understood that file replication service 950 may be in communication with multiple storage servers 920. Besides file replication service 950, other services may
25 interact directly with storage server 920.

[0098] FIG. 10 illustrates a simplified embodiment of a method for retrieving data (such as a media asset) from a storage cloud system. Such a method may be used in conjunction with a storage cloud system such as system 800 of FIG. 8, or system 900 of FIG. 9. The method 1000 may also be used in conjunction with some other storage cloud system. At block 1010, a request
30 may be received for a piece of data, such as a media asset. Such a request may be received from an entity such as a content data network or some other entity accessing the storage cloud.

[0099] At block 1015, if the storage cloud contains more than one storage system, the appropriate storage system that contains the data requested may be identified. At block 1020, a request for the data may be sent to the appropriate storage system. At block 1025, the request may be received at the appropriate storage system. At block 1030, the storage system may
5 determine the appropriate storage server that stores the data requested. Such a determination may involve an index being searched by the storage system to determine which storage server to contact. If the storage system only has one link to a storage server, such a determination may not be necessary. At block 1035, the request for the data may be sent to the appropriate storage server. This request may be received by the storage server at block 1040. At block 1045, a
10 storage server may determine the appropriate storage device that stores the requested data. This step may not be necessary if only one storage device is linked to the storage server. At block 1050, the storage server may send a request for the data to the appropriate storage device. This request may be formatted using the appropriate protocol for the storage device receiving the request.

[0100] At block 1055, the request may be received at the appropriate storage device in the appropriate protocol. The storage device may then retrieve the appropriate data. At block 1060, the requested data may be transmitted from the storage device and received at the storage server at block 1065. The storage server may then transmit the requested data at block 1070, the requested data is then received by the storage system at block 1075. At block 1080, the
20 requested data may be transmitted from the storage system to the requesting party, in this case the CDN.

[0101] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated
25 that, in alternative embodiments, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should
30 not be interpreted to limit the scope of the invention.

[0102] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention.

[0103] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0104] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

WHAT IS CLAIMED IS:

1 1. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:
3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a plurality of cache servers, wherein:
5 the first server is located at a first location;
6 each of the first plurality of data objects comprise a link to one of a
7 plurality of media assets; and
8 the plurality of media assets are stored at a second location separate from
9 the first location;
10 a second server that stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein:
13 the second server is one of the plurality of cache servers; and
14 the second server is communicatively coupled with a first plurality of
15 application servers;
16 a third server that stores a third plurality of data objects and is configured to
17 request the data object from the first server if the data object is not present in the third plurality
18 of data objects, wherein:
19 the third server is one of the plurality of cache servers; and
20 the third server is communicatively coupled with a second plurality of
21 application servers; and
22 a fourth server that stores a fourth plurality of data objects and is configured to
23 request the data object from the second server if the data object is not present in the fourth
24 plurality of data objects, wherein:
25 the fourth server is one of the first plurality of application server; and
26 the fourth server is configured to receive a request from a first client for
27 the data object;
28 a fifth server that stores a fifth plurality of data objects and is configured to
29 request the data object from the second server if the data object is not present in the fifth plurality
30 of data objects, wherein:
31 the fifth server is one of the first plurality of application servers; and

the fifth server is configured to receive a request from a second client for the data object;

a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects, wherein:

the sixth server is one of the second plurality of application servers; and

the sixth server is configured to receive a request from a third client for the data object; and

a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects, wherein:

the seventh server is one of the second plurality of application servers; and

the seventh server is configured to receive a request from a fourth client for the data object.

2. The cloud data persistence system of claim 1, wherein the second location comprises a content delivery network.

3. The cloud data persistence system of claim 1, wherein:
data objects in the second plurality of data objects are at least partially determined based on when requests for those data objects were last received at the second server;
data objects in the third plurality of data objects are at least partially determined based on when requests for those data objects were last received at the third server;
data objects in the fourth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fourth server;
data objects in the fifth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fifth server;
data objects in the sixth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the sixth server; and
data objects in the seventh plurality of data objects are at least partially determined based on when requests for those data objects were last received at the seventh server.

1 4. The cloud data persistence system of claim 1, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth
4 plurality of data objects, and the seventh plurality of data objects.

1 5. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:

3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a second server and a third server, wherein:

5 the first server is located at a first location;

6 each of the first plurality of data objects comprises a link to a media asset
7 of a plurality of media assets;

8 the plurality of media assets is located at a second location; and

9 the second location is distinct from the first location;

10 the second server stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein the second server is communicatively coupled with a fourth server and
13 fifth server; and

14 the fourth server stores a third plurality of data objects and is configured to
15 request the data object from the second server if the data object is not present in the third
16 plurality of data objects, wherein the fourth server is configured to receive a request from a first
17 client for the data object.

1 6. The cloud data persistence system of claim 5 wherein the first plurality of
2 data objects comprises all data objects present in the second plurality of data objects and the third
3 plurality of data objects.

1 7. The cloud data persistence system of claim 5, wherein the third server
2 stores a fourth plurality of data objects and is configured to request the data object from the first
3 server if the data object is not present in the fourth plurality of data objects, wherein:

4 the third server and the second server are data persistence cache servers;

5 and

6 the third server is communicatively coupled with a sixth and seventh
7 server.

1 8. The cloud data persistence system of claim 5, wherein the fifth server
2 stores a fifth plurality of data objects and is configured to request the data object from the second
3 server if the data object is not present in the fifth plurality of data objects, wherein:

4 the fifth server and the fourth servers are application servers; and
5 the fifth server is configured to receive requests from a second client for
6 the data object; and

7 the fifth server is geographically separated from the fourth server.

1 9. The cloud data persistence system of claim 5 wherein the request from the
2 first client for the data object comprises a universal resource indicator.

1 10. The cloud data persistence system of claim 5 wherein each of the first
2 plurality of data objects comprise a universal resource locator and a title linked to a media asset
3 of the plurality of media assets.

1 11. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on a frequency of requests for those data objects received from clients.

1 12. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on when requests for those data objects were last received from clients.

1 13. The cloud data persistence system of claim 5, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

1 14. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data, the method comprising:

3 receiving, by an application center, a first request from a client for a data object,
4 wherein:

5 the application center is one of a plurality of application centers that
6 communicate with a plurality of clients;
7 the application center is located at a first location;
8 the data object comprises a link to a media asset; and
9 the media asset is stored at a second location separate from the first
10 location;
11 determining, by the application center, the data object is not stored among a first
12 plurality of data objects stored at the application center;
13 transmitting, by the application center, a second request for the data object to a
14 first cache server, wherein:
15 the first cache server is communicatively coupled with the plurality of
16 application centers; and
17 the first cache server is one of a plurality of cache servers;
18 receiving, by the cache server, the second request for the data object;
19 determining, by the cache server, the data object is not stored among a second
20 plurality of data objects stored at the cache server;
21 transmitting, by the cache server, a third request for the data object to an origin
22 server, wherein the origin server is communicatively coupled with the plurality of cache servers;
23 receiving, by the origin server, the third request for the data object;
24 locating, by the origin server, the data object among a third plurality of data
25 objects;
26 transmitting, by the origin server, the data object linked to the third request to the
27 cache server;
28 receiving, by the cache server, the data object linked to the third request;
29 transmitting, by the cache server, the data object linked to the second request;
30 receiving, by the application center, the data object linked to the second request;
31 and
32 transmitting, by the application center, the data object linked to the first request to
33 the client.

1 15. The method of claim 14 further comprising, after receiving the first data
2 object from the origin server, storing, by the cache server, the first data object.

1 16. The method of claim 14 further comprising, after receiving the first data
2 object from the cache server, storing, by the application center, the first data object.

1 17. The method of claim 14 wherein the cache server is located at a third
2 location.

1 18. The method of claim 14 wherein the first request for the data object
2 comprises a universal resource indicator.

1 19. The method of claim 14, wherein data objects in the first plurality of data
2 objects and the second plurality of data objects are at least partially determined based on when
3 requests for those data objects were last received from clients.

1 20. The method of claim 14, wherein the third plurality of data objects
2 comprises all data objects in the first and the second pluralities of data objects.

ABSTRACT OF THE DISCLOSURE

Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data object stored locally, it may contact a second server to determine if it has the data object stored. If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

63128000 v1

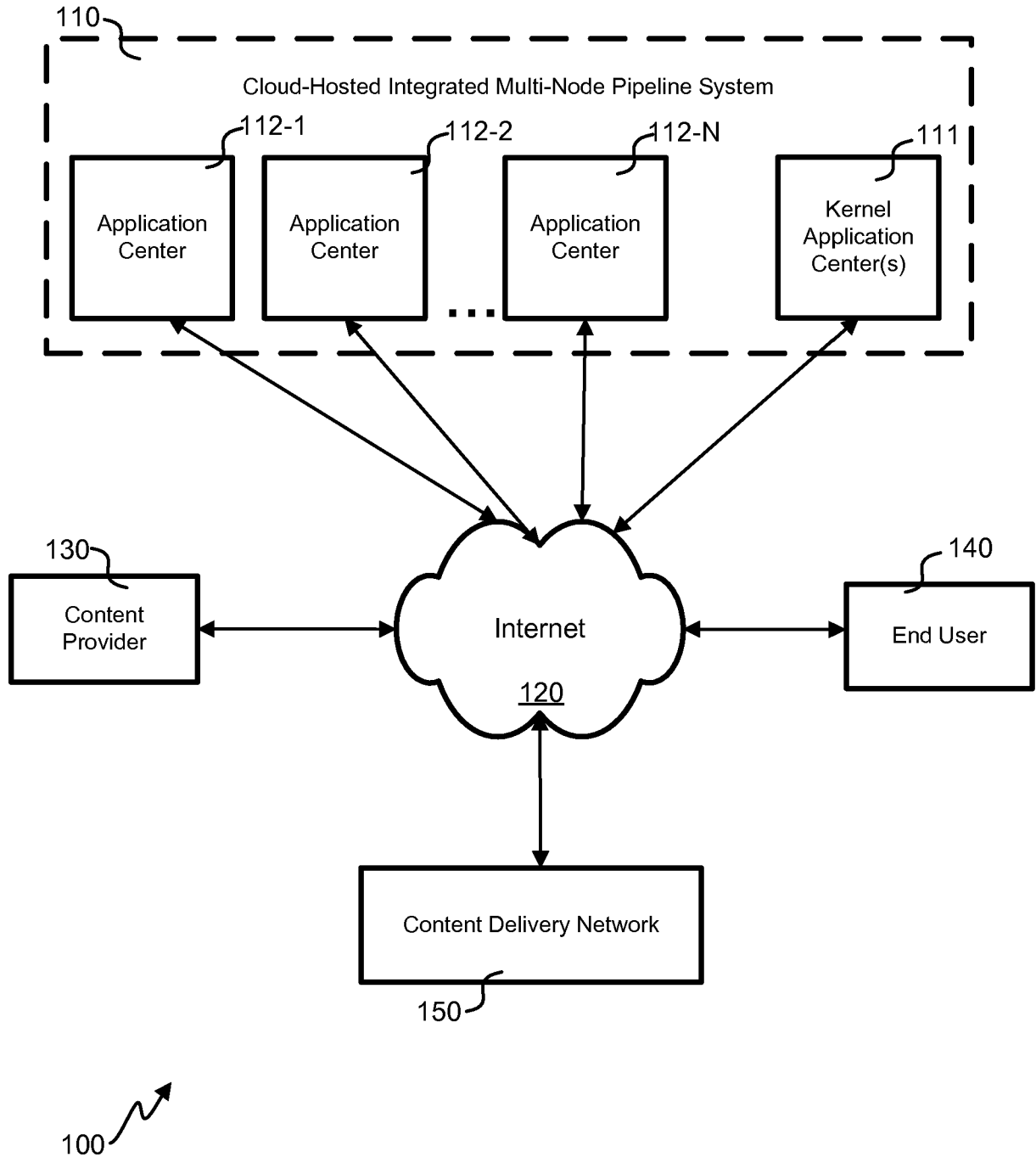


FIG. 1

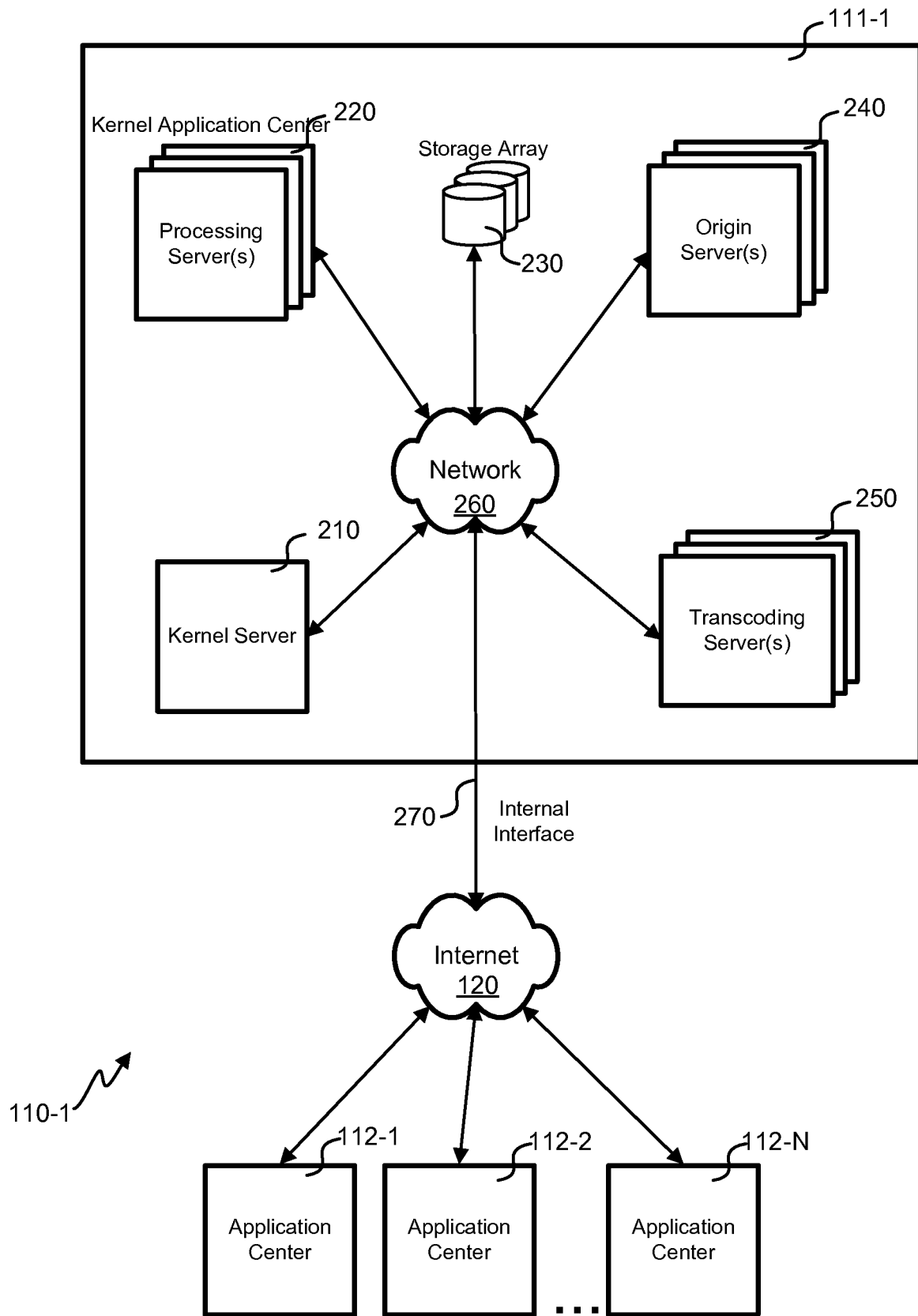


FIG. 2A

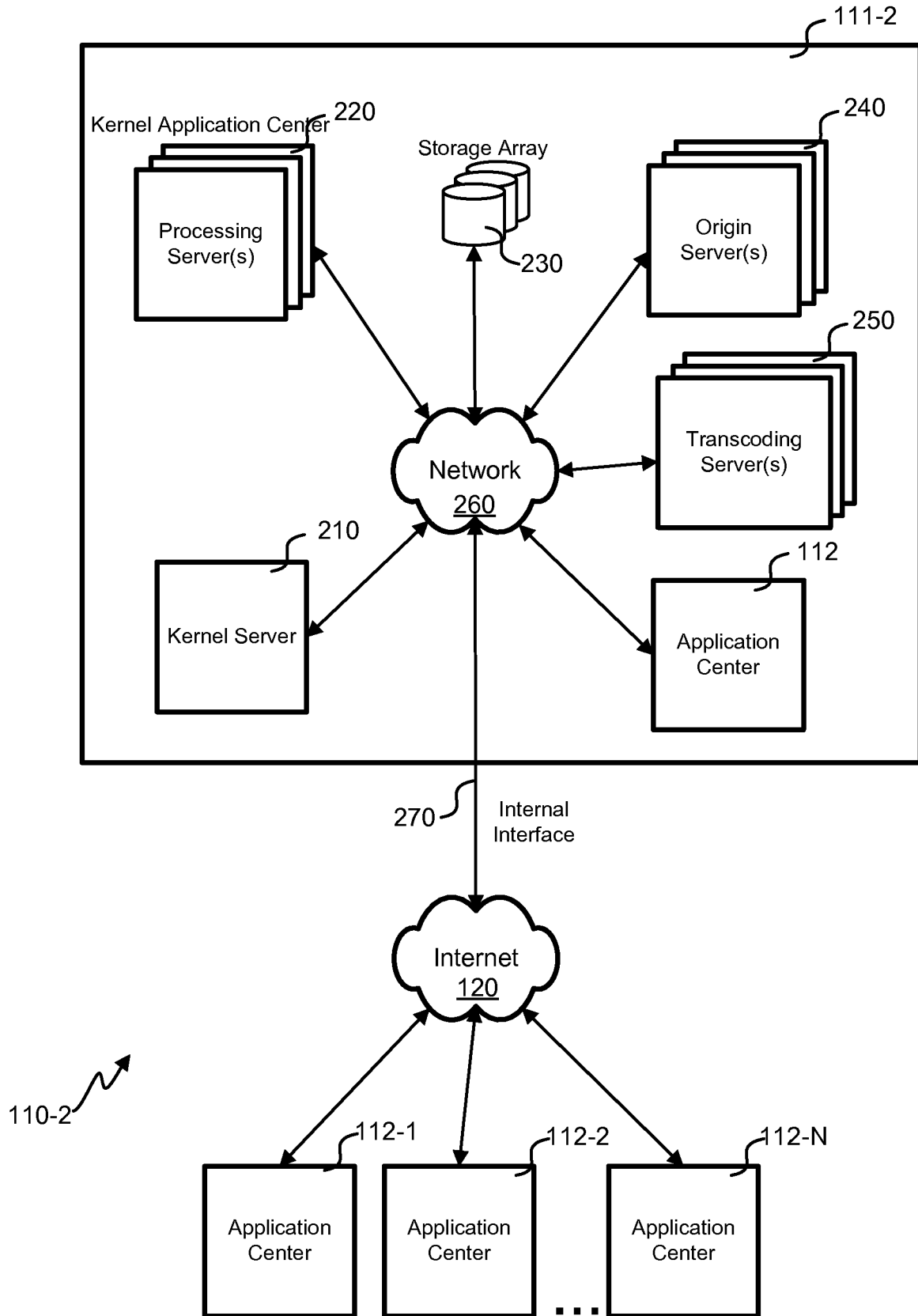


FIG. 2B

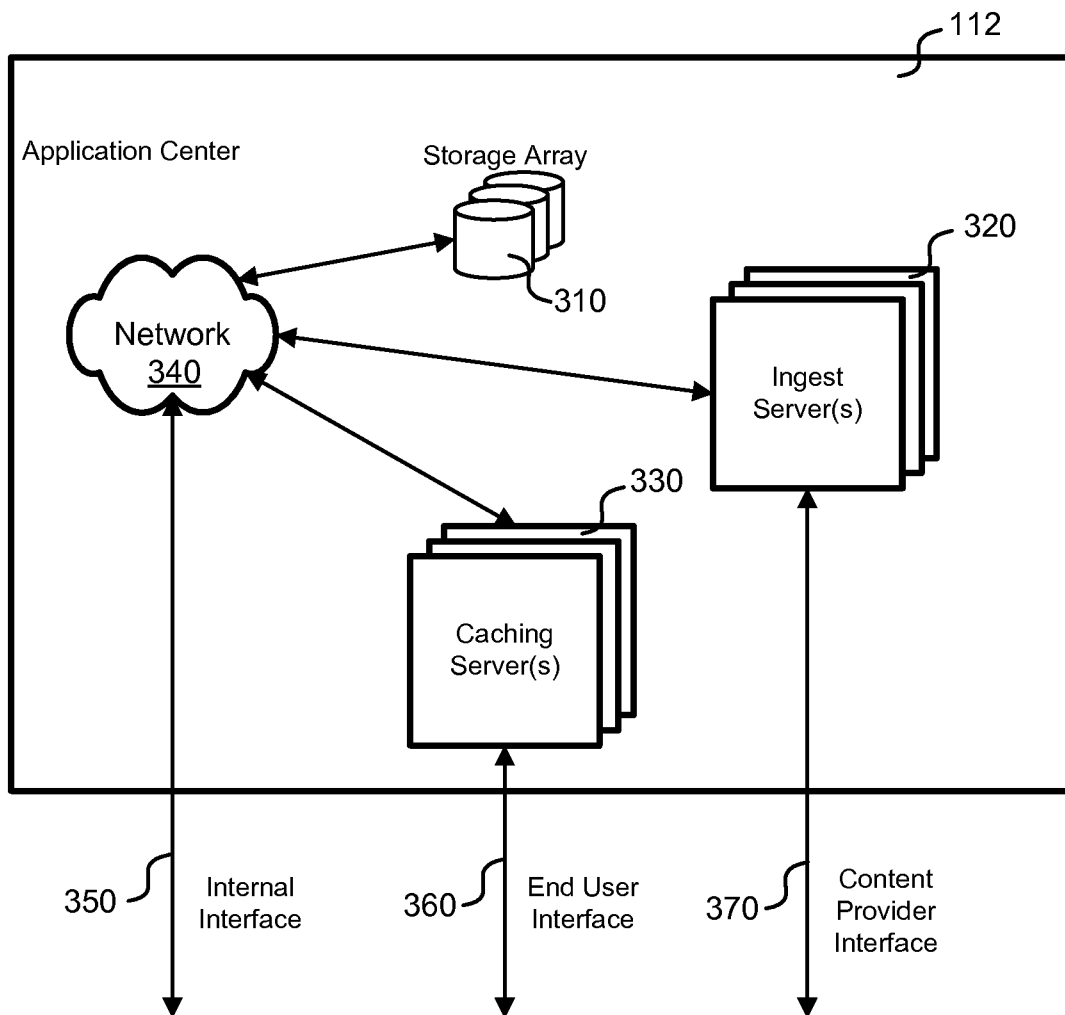
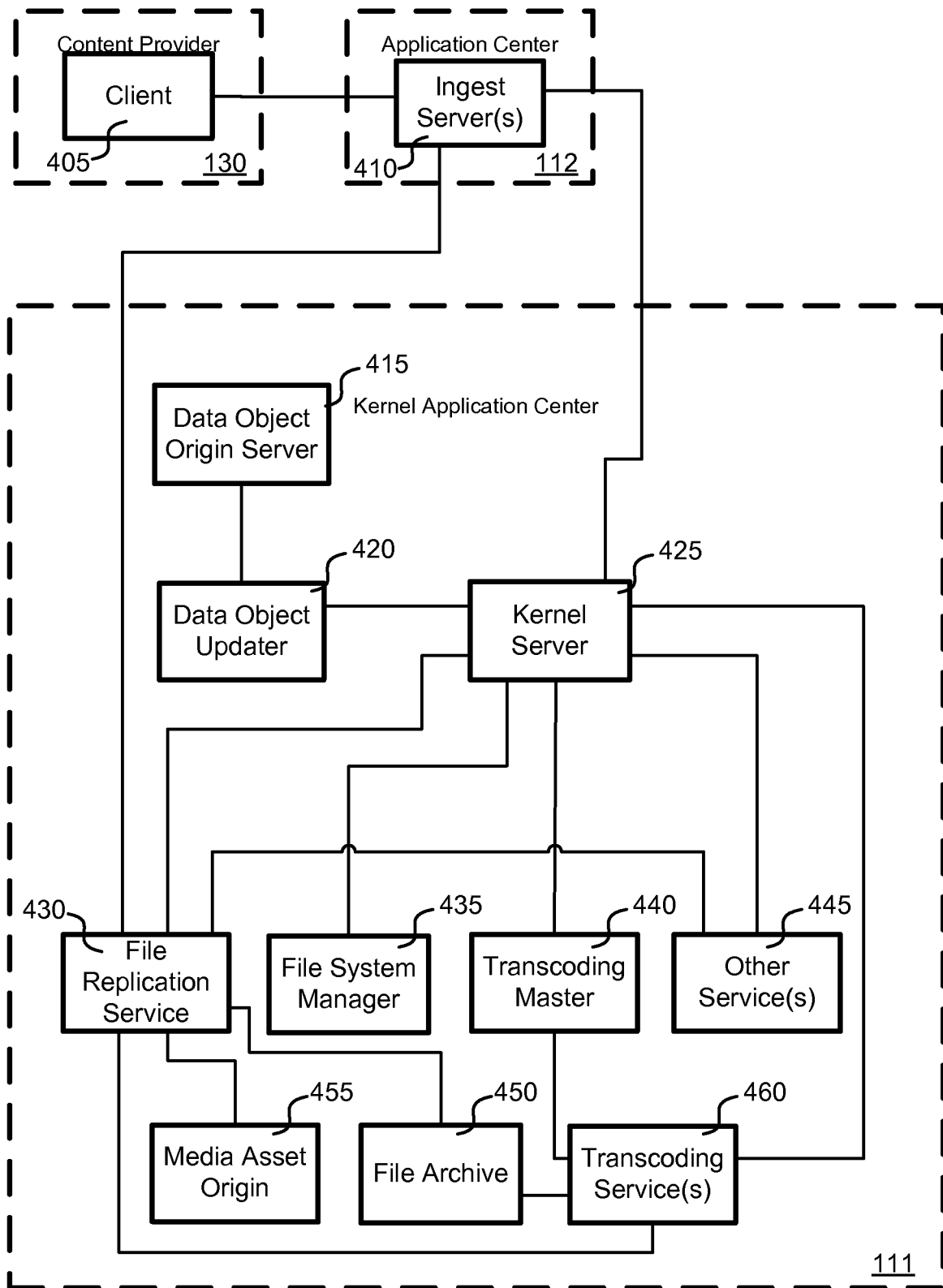


FIG. 3



400-1

FIG. 4A

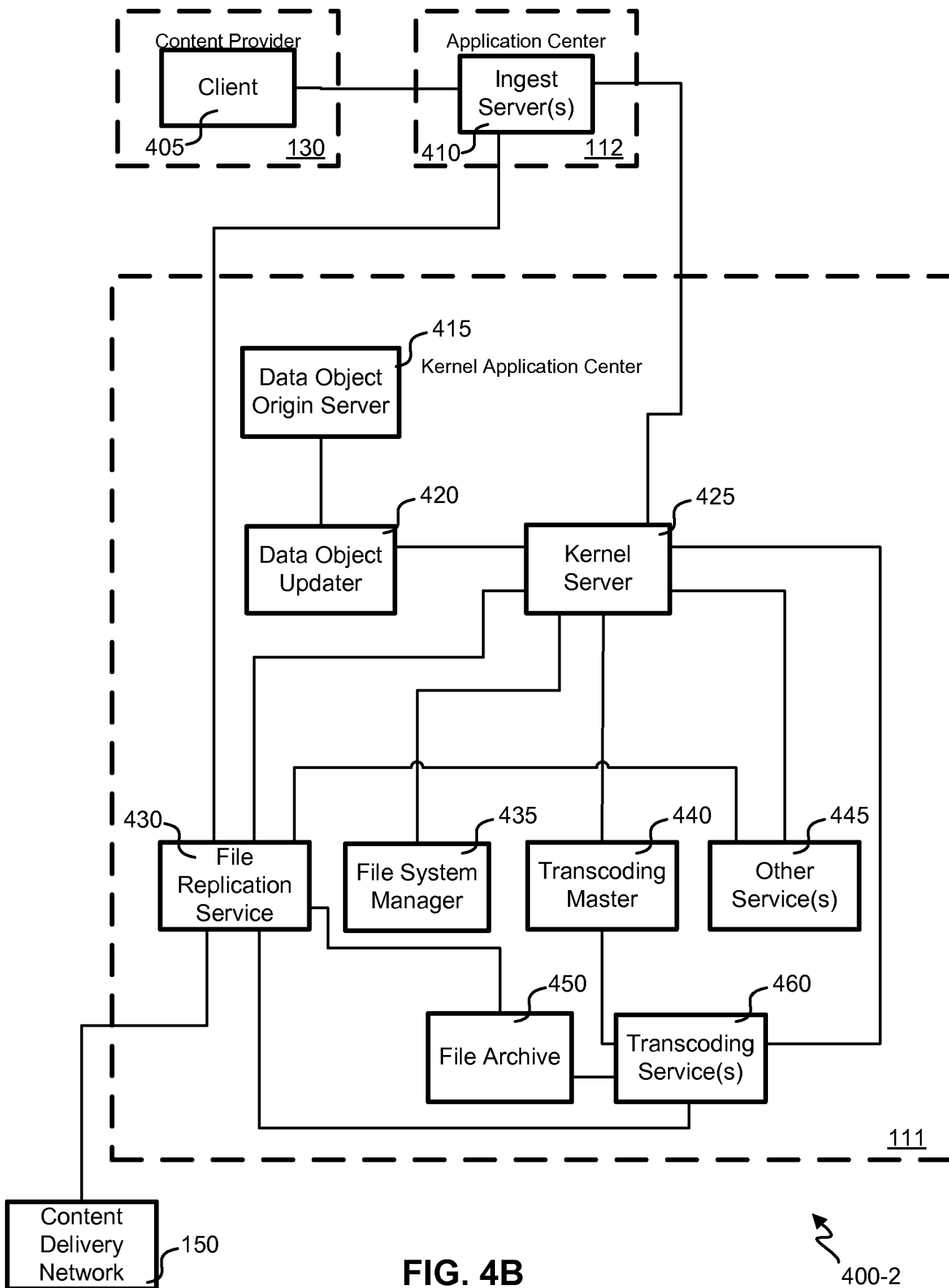


FIG. 4B

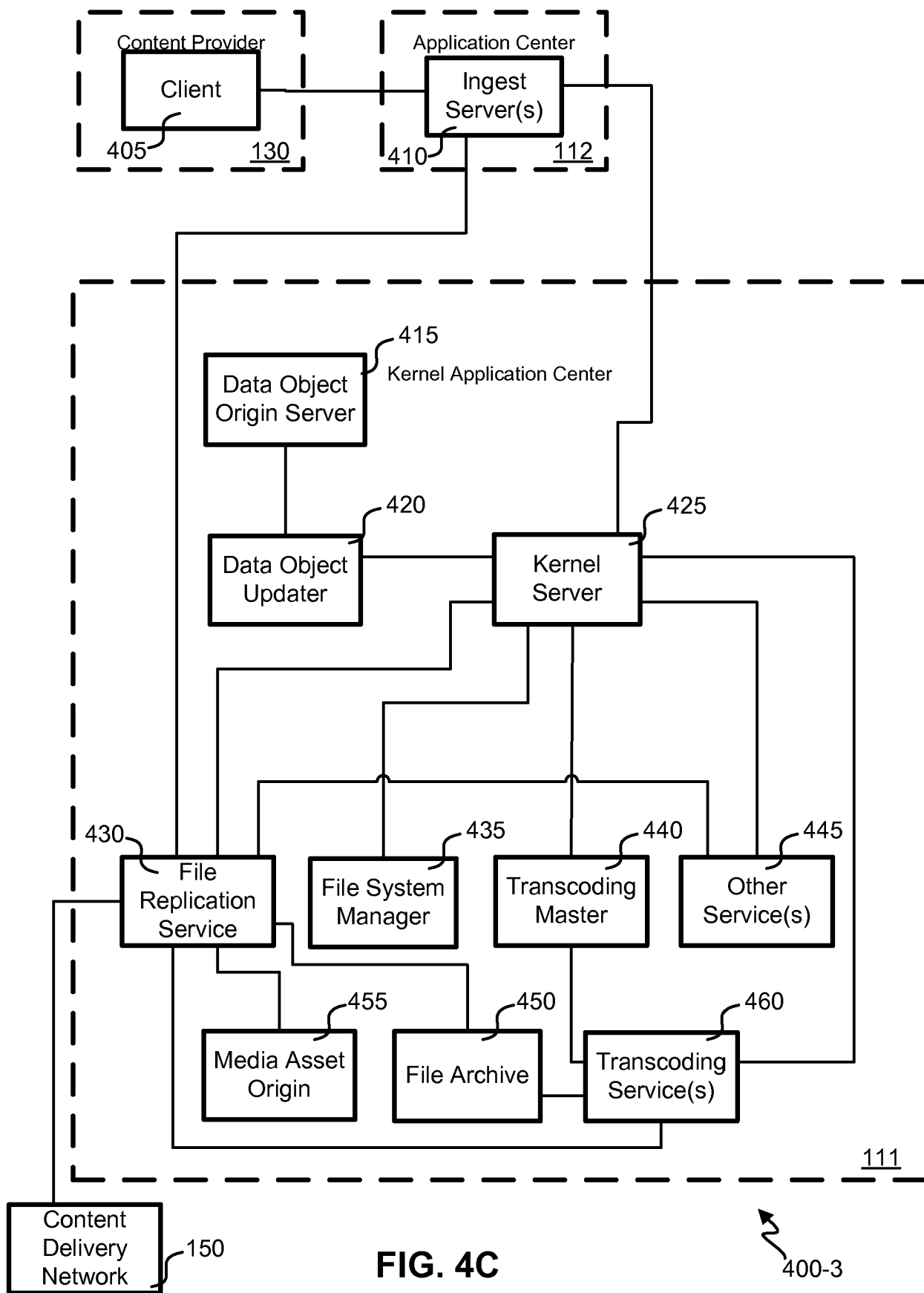


FIG. 4C

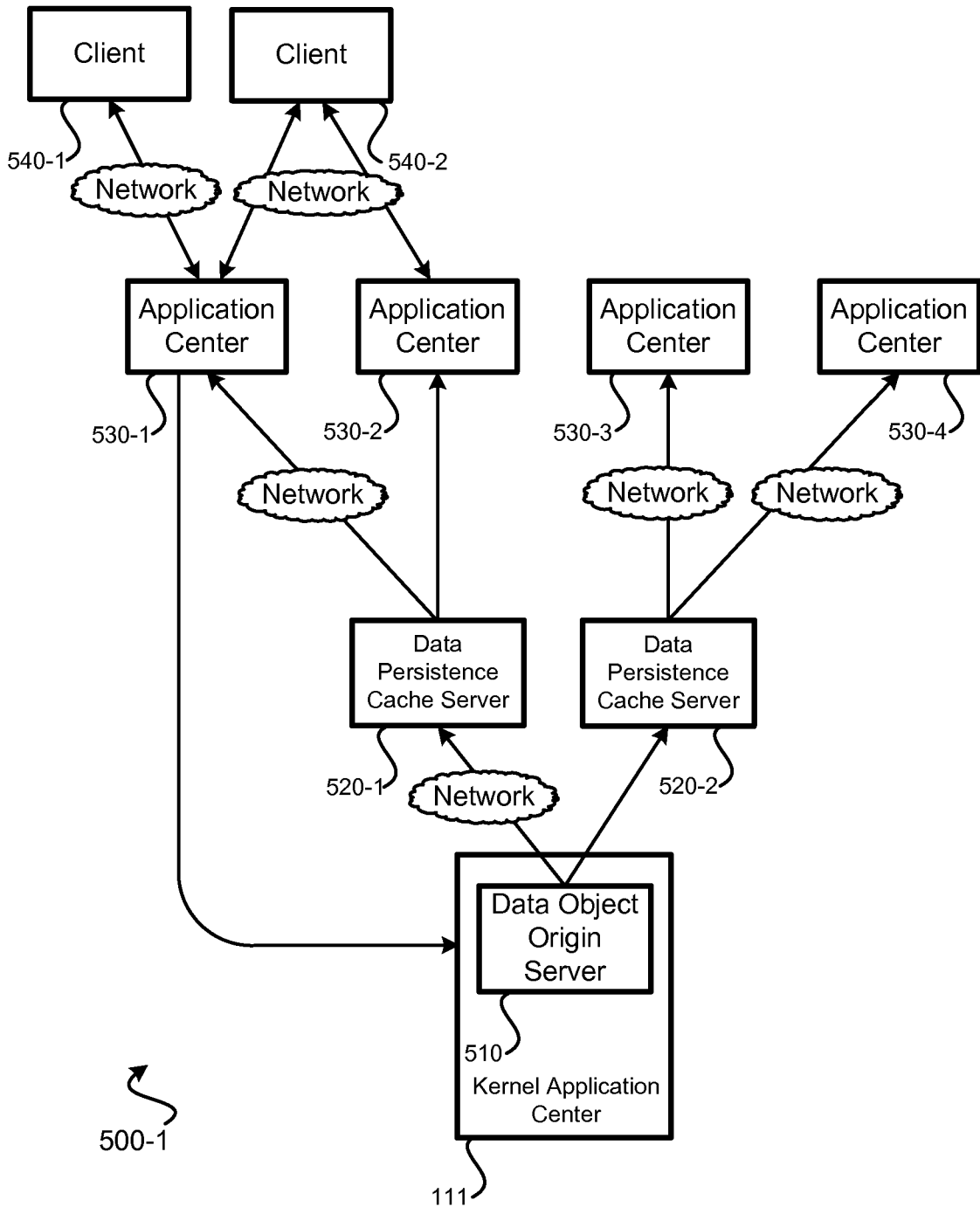


FIG. 5A

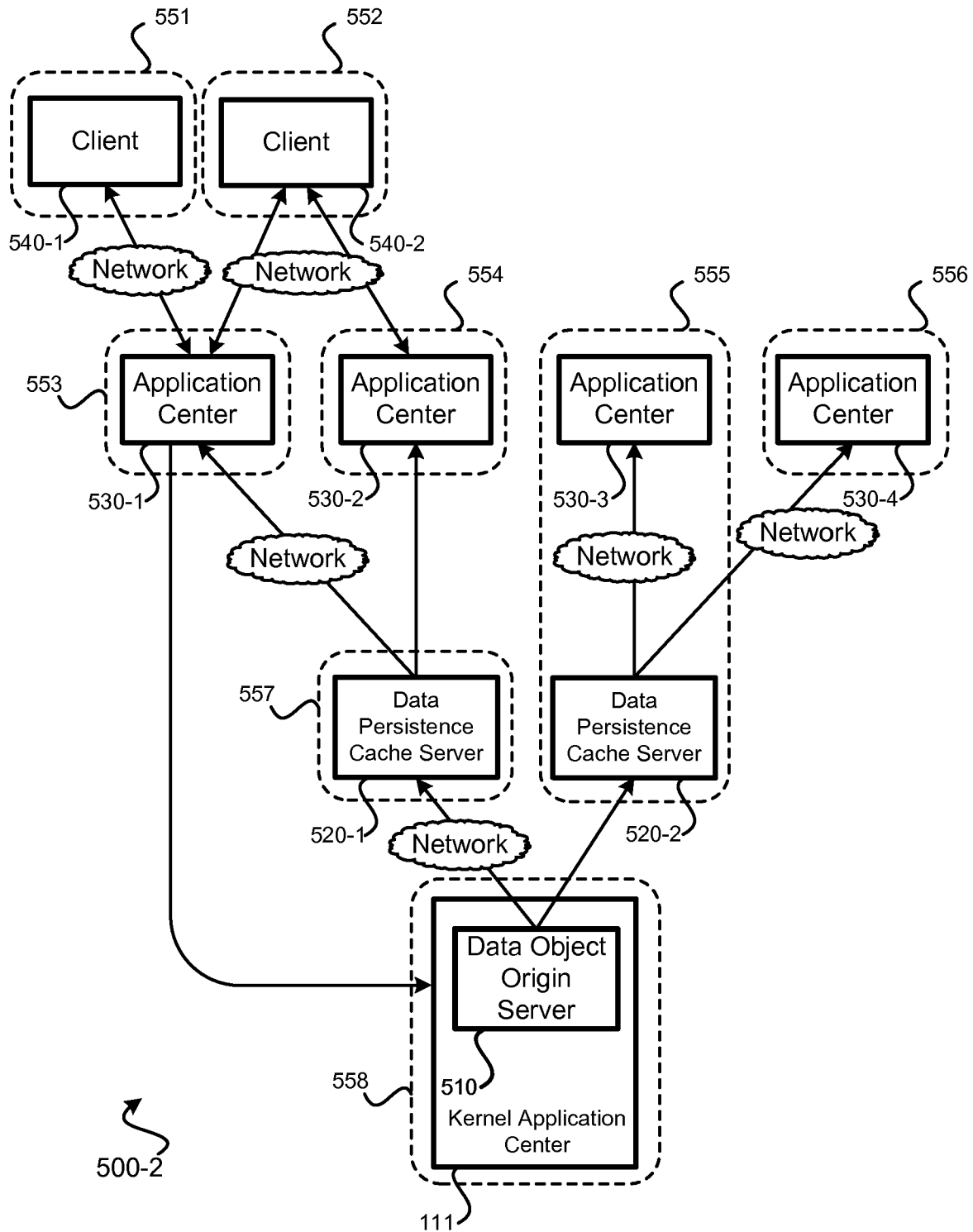


FIG. 5B

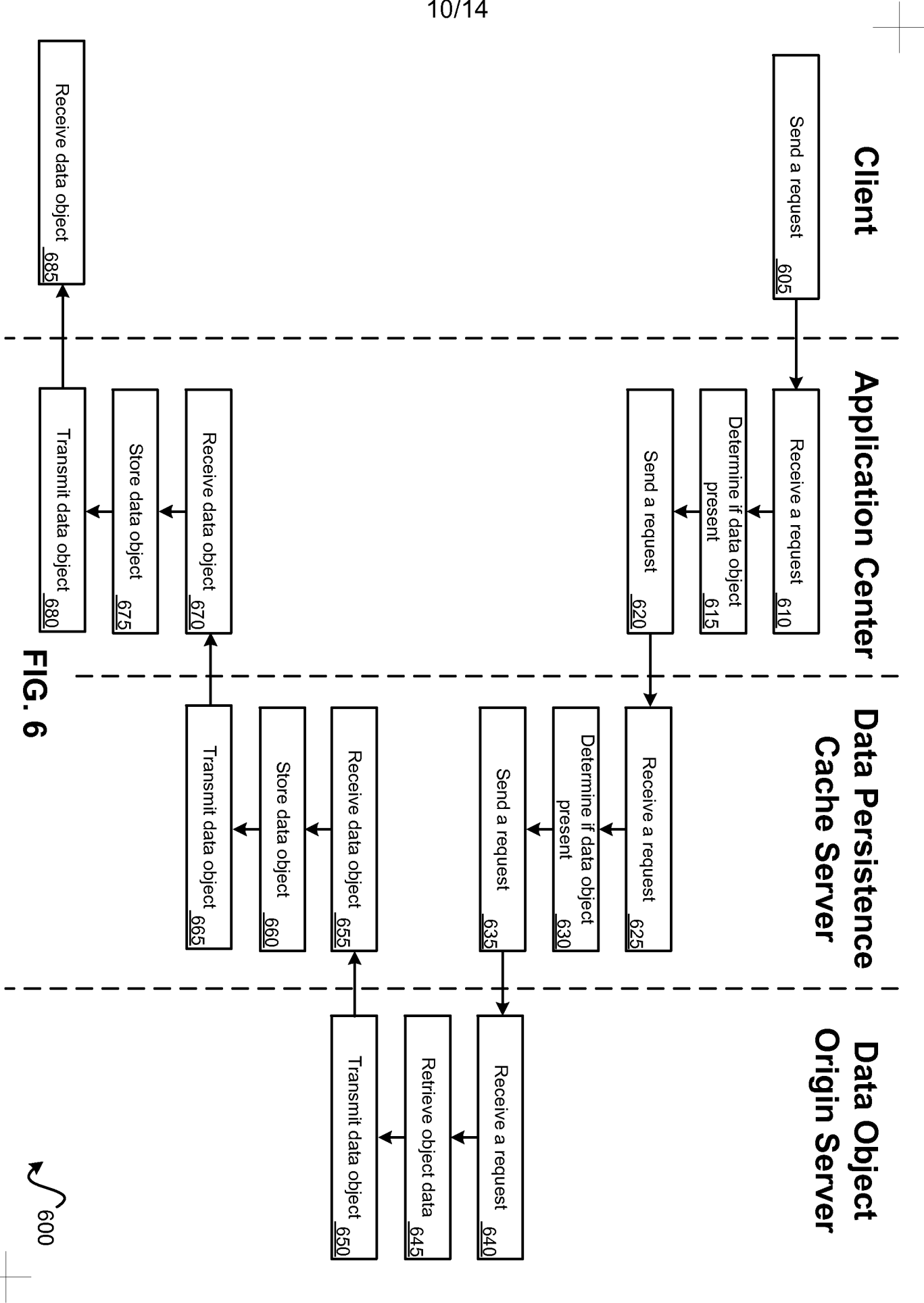


FIG. 6

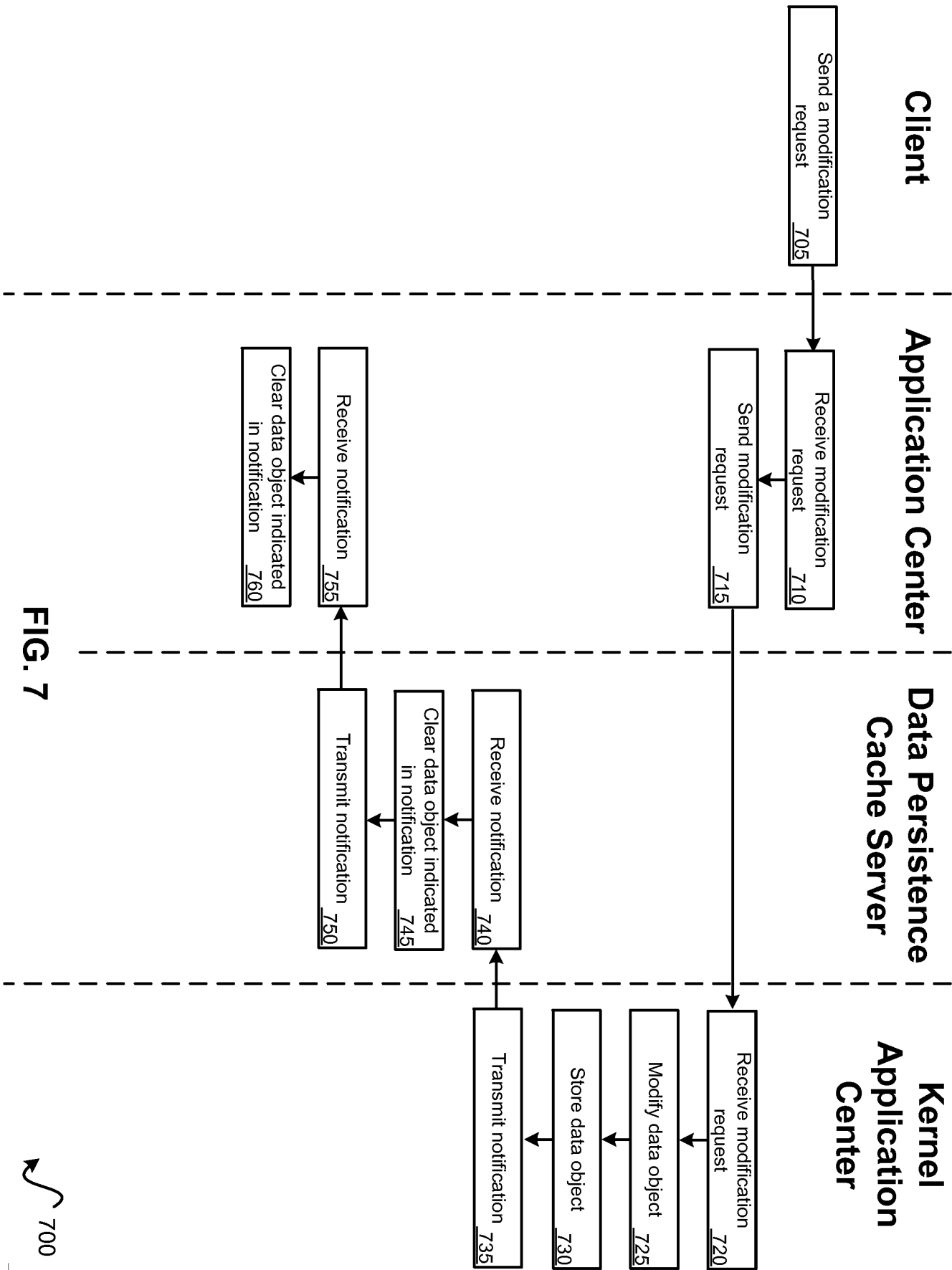


FIG. 7

700

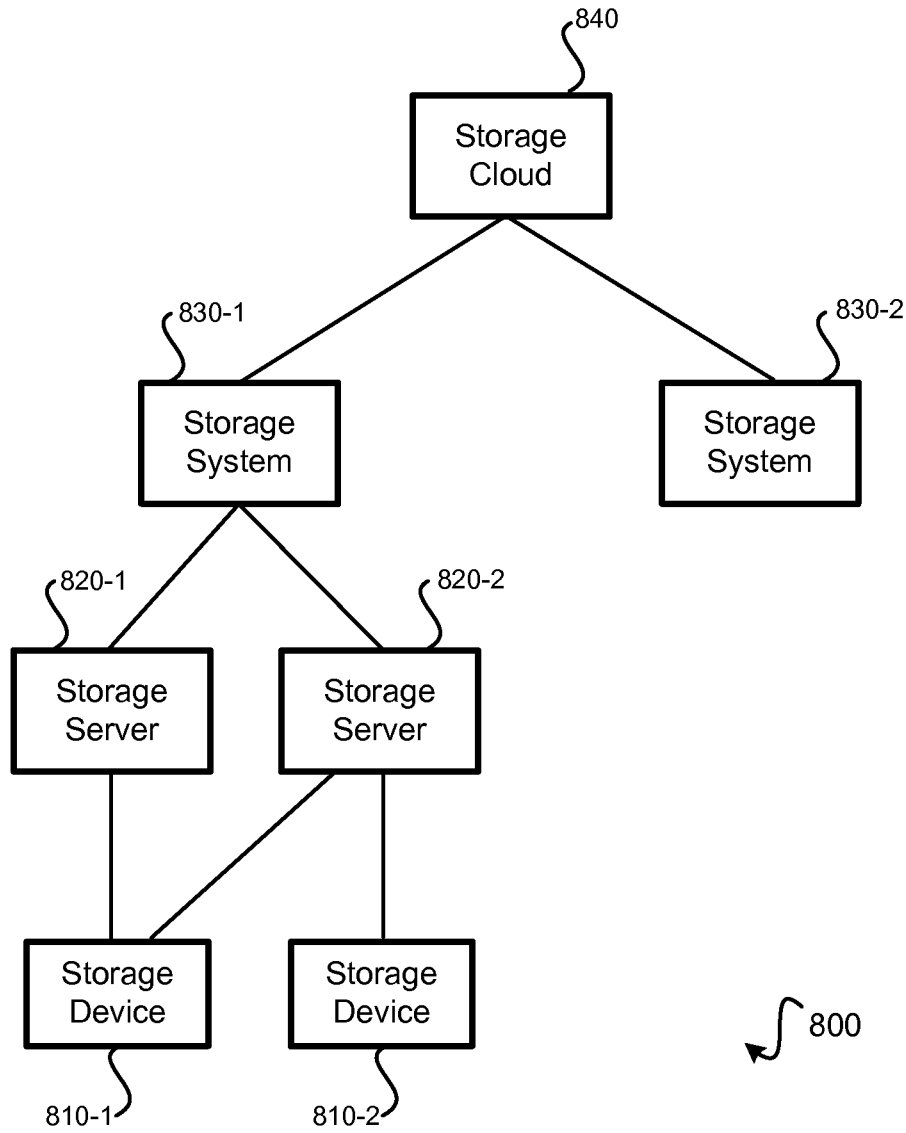


FIG. 8

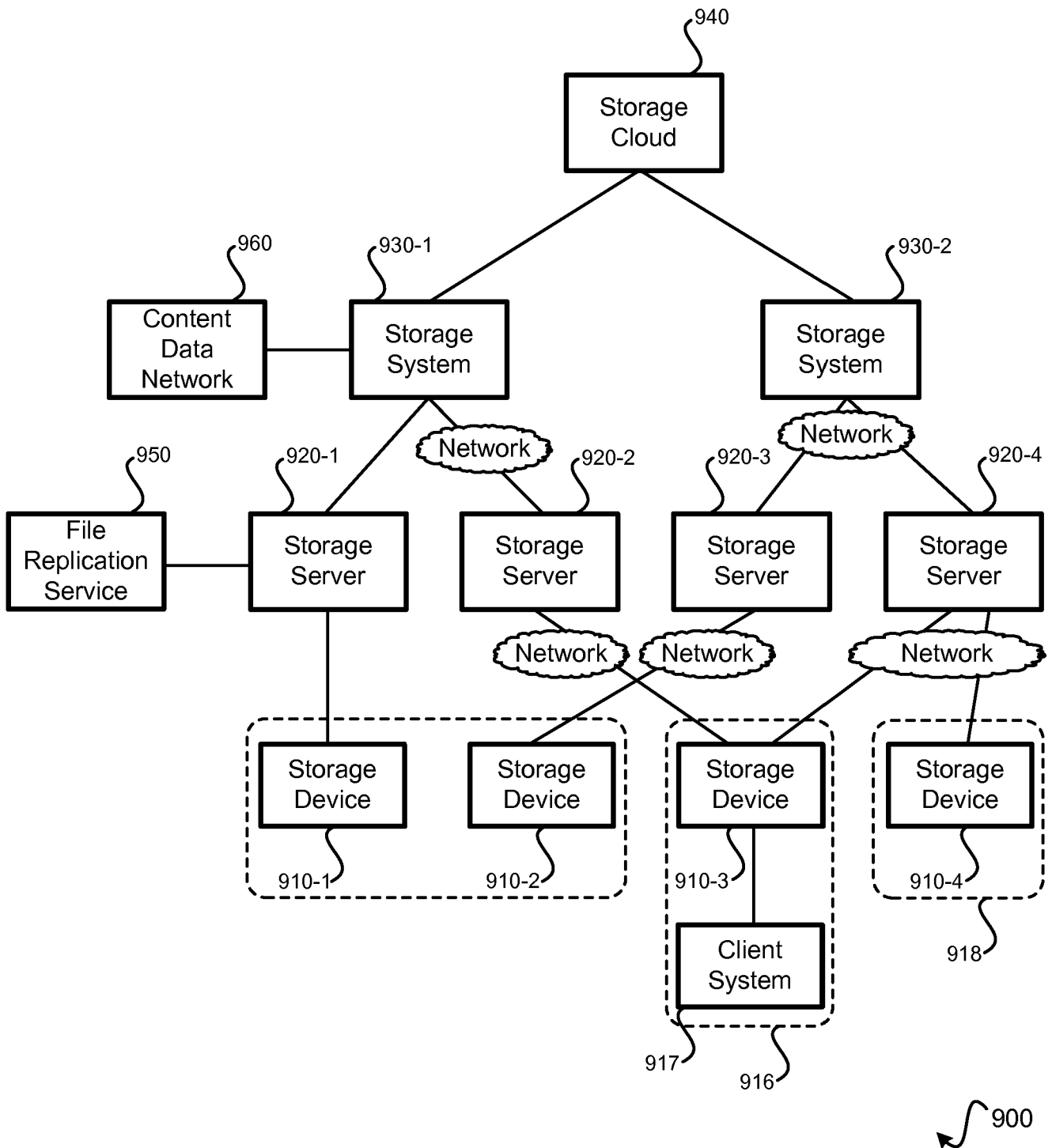


FIG. 9

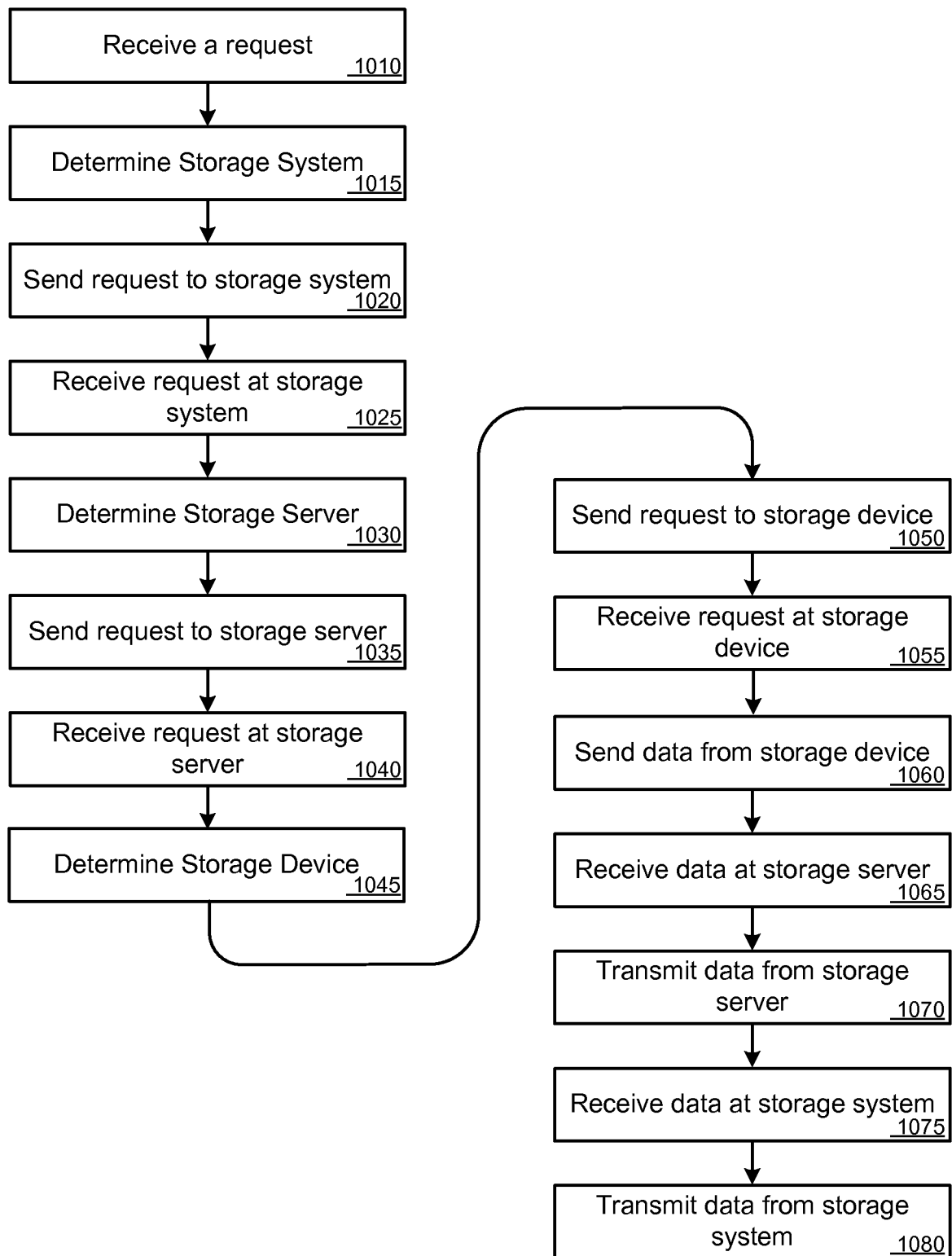


FIG. 10

1000

Application Data Sheet

Application Information

Application number::

Filing Date:: January 31, 2011

Application Type:: Regular

Subject Matter:: Utility

Suggested classification::

Suggested Group Art Unit::

CD-ROM or CD-R??::

Number of CD disks::

Number of copies of CDs::

Sequence Submission::

Computer Readable Form (CRF)?::

Number of copies of CRF::

Title:: CLOUD DATA PERSISTENCE ENGINE

Attorney Docket Number:: 028406-000710US

Request for Early Publication:: No

Request for Non-Publication:: No

Suggested Drawing Figure:: 5A

Total Drawing Sheets:: 14

Small Entity?:: Yes

Latin name::

Variety denomination name::

Petition included?:: No

Petition Type::

Licensed US Govt. Agency::

Contract or Grant Numbers One::

Secrecy Order in Parent Appl.: No

Applicant Information

Applicant Authority Type:: Inventor
Primary Citizenship Country:: US
Status:: Full Capacity
Given Name:: Albert
Middle Name:: J.
Family Name:: McGowan
Name Suffix::
City of Residence:: Phoenix
State or Province of Residence:: AZ
Country of Residence:: US
Street of Mailing Address:: 4201 East Camelback Road, #101
City of Mailing Address:: Phoenix
State or Province of mailing address:: AZ
Country of mailing address:: US
Postal or Zip Code of mailing address:: 85018

Applicant Authority Type:: Inventor
Primary Citizenship Country:: US
Status:: Full Capacity
Given Name:: Richard
Middle Name:: L.
Family Name:: Carls
Name Suffix::
City of Residence:: Scottsdale
State or Province of Residence:: AZ
Country of Residence:: US
Street of Mailing Address:: 19793 West 84th Way
City of Mailing Address:: Scottsdale
State or Province of mailing address:: AZ

Country of mailing address:: US
Postal or Zip Code of mailing address:: 85255

Correspondence Information

Correspondence Customer Number:: 20350

Representative Information

Representative Designation:: Representative Number:: Representative Name::
Primary 64,190 Karam J. Saab

Domestic Priority Information

Application:: Continuity Type:: Parent Application:: Parent Filing Date::

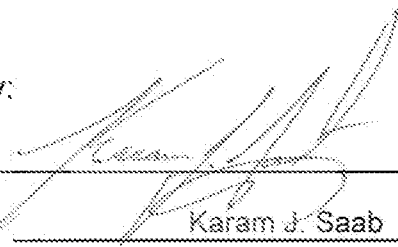
Foreign Priority Information

Country:: Application number:: Filing Date::
Australia 2010202782 July 1, 2010

Assignee Information

Assignee Name:: Unicorn Media, Inc.
Street of mailing address:: 24 West 5th Street
City of mailing address:: Tempe
State or Province of mailing address:: Arizona
Country of mailing address:: US
Postal or Zip Code of mailing address:: 85281

Submitted by:

Signature  Date January 31, 2011
Printed Name Karam J. Saab Registration Number 64,190

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		CLOUD DATA PERSISTENCE ENGINE		
First Named Inventor/Applicant Name:		Albert J. McGowan		
Filer:		Karam J. Saab/Corina Aschenbrenner		
Attorney Docket Number:		028406-000710US		
Filed as Small Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Utility filing Fee (Electronic filing)	4011	1	82	82
Utility Search Fee	2111	1	270	270
Utility Examination Fee	2311	1	110	110
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				462

Electronic Acknowledgement Receipt

EFS ID:	9345561
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab/Corina Aschenbrenner
Filer Authorized By:	Karam J. Saab
Attorney Docket Number:	028406-000710US
Receipt Date:	31-JAN-2011
Filing Date:	
Time Stamp:	17:04:18
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$ 462
RAM confirmation Number	4668
Deposit Account	201430
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		PPHPatApp.pdf	163516	yes	39
			955a31c7eb1361d6429492f4c41c3f689d0673f4		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Specification		1	32	
	Claims		33	38	
	Abstract		39	39	
Warnings:					
Information:					
2	Drawings-only black and white line drawings	PPHDraws.pdf	96585	no	14
			0ec83c80ec13a54d1d1dee494b2d696d6434710b		
Warnings:					
Information:					
3	Application Data Sheet	PPHADs.pdf	640610	no	3
			fba1625d47fb2f9bd2e757b13c981e5c6e0b96e		
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
4	Fee Worksheet (PTO-875)	fee-info.pdf	33087	no	2
			33114df2efe4a1c16f844337f65e90a06819ebab		
Warnings:					
Information:					
Total Files Size (in bytes):			933798		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

**REQUEST FOR PARTICIPATION IN THE PATENT PROSECUTION HIGHWAY (PPH) PILOT PROGRAM
BETWEEN IP AUSTRALIA (IPAU) AND THE USPTO**

Application No.:	13/017,866	Filing Date:	January 31, 2011
First Named Inventor:	Albert John McGowan et al.		
Attorney Docket No.:	028406-000710US		

Title of the
invention: CLOUD DATA PERSISTENCE ENGINE

THIS REQUEST FOR PARTICIPATION IN THE PPH pilot program along with the required documents must be submitted via EFS-Web. Information regarding EFS-Web is available at http://www.uspto.gov/EBC/efs_help.html.

APPLICANT HEREBY REQUESTS PARTICIPATION IN THE PATENT PROSECUTION HIGHWAY (PPH) PILOT PROGRAM AND PETITIONS TO MAKE THE ABOVE-IDENTIFIED APPLICATION SPECIAL UNDER THE PPH PILOT PROGRAM.

The above-identified application (1) validly claims priority under 35 U.S.C. 119(a) and 37 CFR 1.55 to one or more corresponding IPAU application(s) or to a PCT application that does not contain any priority claim, or (2) is a national stage entry of a PCT application that does not contain any priority claim.

The IPAU/PCT application number(s) is/are: AU 2010202782

The filing date of the IPAU/PCT application(s) is/are: July 1, 2010

I. List of Required Documents:

- a. A copy of all IPAU office actions which are relevant to patentability in the above-identified IPAU application(s)
- ☒ Is attached.
- b. A copy of all claims which were determined to be patentable by IPAU in the above-identified IPAU application(s)
- ☒ Is attached.
- c. (1) An information disclosure statement listing the documents cited in the IPAU office actions
- ☒ Is attached.
- ☐ Has already been filed in the above-identified U.S. application on _____

(2) Copies of all documents (except for U.S. patents or U.S. patent application publications)

- ☒ Are attached.
- ☐ Have already been filed in the above-identified U.S. application on _____

[Page 1 of 2]

This collection of information is required by 35 U.S.C. 119, 37 CFR 1.55, and 37 CFR 1.102(d). The information is required to obtain or retain a benefit by the public, which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REQUEST FOR PARTICIPATION IN THE PATENT PROSECUTION HIGHWAY (PPH) PILOT PROGRAM
BETWEEN IPAU AND THE USPTO

(continued)

Application No.:	13/017,866
First Named Inventor:	Albert John McGowan et al.

II. Claims Correspondence Table:

[illegible]

III. All the claims in the US application sufficiently correspond to the patentable/allowable claims in the IPAU application.

Signature 	Date January 31, 2011
Name (Print/Typed) Karam J. Saab	Registration Number 64,190



Australian Government

IP Australia

Discovery House, Phillip ACT 2606
PO Box 200, Woden ACT 2606
Australia
Phone: 1300 651 010
International Callers: +61-2 6283 2899
Facsimile: +61-2 6283 7999
Email: assist@ipaustalia.gov.au
Website: www.ipaustralia.gov.au

10 November 2010

Pizzeys
PO Box 291
Woden
Canberra ACT 2606
Australia

RE: Patent Application number 2010202782
in the name(s) of Unicorn Media, Inc..

Your Ref: 31560UNI/MRR/kj

Dear Sir/Madam

Notice of Acceptance

I am pleased to advise that there are no objections to the application. The Examiner has incorporated into the complete specification amendments made under PCT Article 34 and/or Section 104 based on the following:

PCT: - Not Applicable

S104 amendments up to and including item number: - Not Applicable

The application and complete specification were accepted on 28 October 2010 and a notice of the acceptance will appear in the Supplement to the Australian Official Journal of Patents on 25 November 2010.

For applications filed after 31 August 2002 or that entered National Phase after 31 August 2002, a fee for acceptance of an application applies. This fee includes a component determined by the number of claims in excess of 20. If the acceptance fee applies and has not been paid, an Invitation to Pay (ITP) will issue. If the amount is paid by the due date, your patent will be sealed as soon as practicable after the 3 month period for opposition has expired. The total number of claims at acceptance has been reported as: 22

If you need any further information please contact (02) 6283 2898. Alternatively, you may contact us by email at assist@ipaustalia.gov.au.

Yours faithfully

Electronic Records Administration

IPA Australia • Patents • Trade Marks • Designs • Plant Breeder's Rights

ABN 38 113 072 755

Petitioner Disney Ex-1004, 0064

Bibliographic Details at Acceptance

PAMS Details for Patent Application No. 2010202782

Your Reference: 31560UNI/MRR/kj

Acceptance Date: 28 October 2010

Acceptance to be Advertised: 25 November 2010

Complete Filing Date: 1 July 2010

OPI Date:

National Phase Entry Date:

Applicant Name: Unicorn Media, Inc.

Address: 24 West 5th Street
Tempe AZ 85281
United States of America

Inventor Name(s): McGowan, Albert J.
Carls, Richard L.

Title: Cloud data persistence engine

Agent Name: Pizzeys

Address for Correspondence: PO Box 291
Woden
Canberra ACT 2606
Australia

Address for Legal Service: Level 2, Woden Plaza Offices
Woden Town Square
Woden
Canberra ACT 2606
Australia

Prior Art Documents:
US 2003/0061272

Int. Cl:
G06F 17/30 (2006.01)

Continuation Fee Due Date: 1 July 2015

WHAT IS CLAIMED IS:

1 1. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:
3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a plurality of cache servers, wherein:
5 the first server is located at a first location;
6 each of the first plurality of data objects comprise a link to one of a
7 plurality of media assets; and
8 the plurality of media assets are stored at a second location separate from
9 the first location;
10 a second server that stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein:
13 the second server is one of the plurality of cache servers; and
14 the second server is communicatively coupled with a first plurality of
15 application servers;
16 a third server that stores a third plurality of data objects and is configured to
17 request the data object from the first server if the data object is not present in the third plurality
18 of data objects, wherein:
19 the third server is one of the plurality of cache servers; and
20 the third server is communicatively coupled with a second plurality of
21 application servers; and
22 a fourth server that stores a fourth plurality of data objects and is configured to
23 request the data object from the second server if the data object is not present in the fourth
24 plurality of data objects, wherein:
25 the fourth server is one of the first plurality of application server; and
26 the fourth server is configured to receive a request from a first client for
27 the data object;
28 a fifth server that stores a fifth plurality of data objects and is configured to
29 request the data object from the second server if the data object is not present in the fifth plurality
30 of data objects, wherein:
31 the fifth server is one of the first plurality of application servers; and

32 the fifth server is configured to receive a request from a second client for
33 the data object;
34 a sixth server that stores a sixth plurality of data objects and is configured to
35 request the data object from the third server if the data object is not present in the sixth plurality
36 of data objects, wherein:
37 the sixth server is one of the second plurality of application servers; and
38 the sixth server is configured to receive a request from a third client for the
39 data object; and
40 a seventh server that stores a seventh plurality of data objects and is configured to
41 request the data object from the third server if the data object is not present in the seventh
42 plurality of data objects, wherein:
43 the seventh server is one of the second plurality of application servers; and
44 the seventh server is configured to receive a request from a fourth client
45 for the data object.

1 2. The cloud data persistence system of claim 1, wherein the second location
2 comprises a content delivery network.

1 3. The cloud data persistence system of claim 1, wherein:
2 data objects in the second plurality of data objects are at least partially determined
3 based on when requests for those data objects were last received at the second server;
4 data objects in the third plurality of data objects are at least partially determined
5 based on when requests for those data objects were last received at the third server;
6 data objects in the fourth plurality of data objects are at least partially determined
7 based on when requests for those data objects were last received at the fourth server;
8 data objects in the fifth plurality of data objects are at least partially determined
9 based on when requests for those data objects were last received at the fifth server;
10 data objects in the sixth plurality of data objects are at least partially determined
11 based on when requests for those data objects were last received at the sixth server; and
12 data objects in the seventh plurality of data objects are at least partially
13 determined based on when requests for those data objects were last received at the seventh
14 server.

1 4. The cloud data persistence system of claim 1, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth
4 plurality of data objects, and the seventh plurality of data objects.

1 5. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:
3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a second server and a third server, wherein:
5 the first server is located at a first location;
6 each of the first plurality of data objects comprises a link to a media asset
7 of a plurality of media assets;
8 the plurality of media assets is located at a second location; and
9 the second location is distinct from the first location;
10 the second server stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein the second server is communicatively coupled with a fourth server and
13 fifth server; and
14 the fourth server stores a third plurality of data objects and is configured to
15 request the data object from the second server if the data object is not present in the third
16 plurality of data objects, wherein the fourth server is configured to receive a request from a first
17 client for the data object.

1 6. The cloud data persistence system of claim 5 wherein the first plurality of
2 data objects comprises all data objects present in the second plurality of data objects and the third
3 plurality of data objects.

1 7. The cloud data persistence system of claim 5, wherein the third server
2 stores a fourth plurality of data objects and is configured to request the data object from the first
3 server if the data object is not present in the fourth plurality of data objects, wherein:
4 the third server and the second server are data persistence cache servers;
5 and

6 the third server is communicatively coupled with a sixth and seventh
7 server.

1 8. The cloud data persistence system of claim 5, wherein the fifth server
2 stores a fifth plurality of data objects and is configured to request the data object from the second
3 server if the data object is not present in the fifth plurality of data objects, wherein:

4 the fifth server and the fourth servers are application servers; and
5 the fifth server is configured to receive requests from a second client for
6 the data object; and

7 the fifth server is geographically separated from the fourth server.

1 9. The cloud data persistence system of claim 5 wherein the request from the
2 first client for the data object comprises a universal resource indicator.

1 10. The cloud data persistence system of claim 5 wherein each of the first
2 plurality of data objects comprise a universal resource locator and a title linked to a media asset
3 of the plurality of media assets.

1 11. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on a frequency of requests for those data objects received from clients.

1 12. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on when requests for those data objects were last received from clients.

1 13. The cloud data persistence system of claim 5, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

1 14. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data, the method comprising:

3 receiving, by an application center, a first request from a client for a data object,
4 wherein:

5 the application center is one of a plurality of application centers that
6 communicate with a plurality of clients;
7 the application center is located at a first location;
8 the data object comprises a link to a media asset; and
9 the media asset is stored at a second location separate from the first
10 location;
11 determining, by the application center, the data object is not stored among a first
12 plurality of data objects stored at the application center;
13 transmitting, by the application center, a second request for the data object to a
14 first cache server, wherein:
15 the first cache server is communicatively coupled with the plurality of
16 application centers; and
17 the first cache server is one of a plurality of cache servers;
18 receiving, by the cache server, the second request for the data object;
19 determining, by the cache server, the data object is not stored among a second
20 plurality of data objects stored at the cache server;
21 transmitting, by the cache server, a third request for the data object to an origin
22 server, wherein the origin server is communicatively coupled with the plurality of cache servers;
23 receiving, by the origin server, the third request for the data object;
24 locating, by the origin server, the data object among a third plurality of data
25 objects;
26 transmitting, by the origin server, the data object linked to the third request to the
27 cache server;
28 receiving, by the cache server, the data object linked to the third request;
29 transmitting, by the cache server, the data object linked to the second request;
30 receiving, by the application center, the data object linked to the second request;
31 and
32 transmitting, by the application center, the data object linked to the first request to
33 the client.

1 15. The method of claim 14 further comprising, after receiving the first data
2 object from the origin server, storing, by the cache server, the first data object.

1 16. The method of claim 14 further comprising, after receiving the first data
2 object from the cache server, storing, by the application center, the first data object.

1 17. The method of claim 14 wherein the cache server is located at a third
2 location.

1 18. The method of claim 14 wherein the first request for the data object
2 comprises a universal resource indicator.

1 19. The method of claim 14, wherein data objects in the first plurality of data
2 objects and the second plurality of data objects are at least partially determined based on when
3 requests for those data objects were last received from clients.

1 20. The method of claim 14, wherein the third plurality of data objects
2 comprises all data objects in the first and the second pluralities of data objects.

1 21. A cloud data persistence system for distribution of data substantially as
2 herein described.

1 22. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data substantially as herein described.

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on January 31, 2011

PATENT
Attorney Docket No.: 028406-000710US

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Corina K. Aschenbrenner/
Corina K. Aschenbrenner

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Albert John McGowan et al.

Application No.: 13/017,866

Filed: January 31, 2011

For: CLOUD DATA PERSISTENCE
ENGINE

Customer No.: 20350

Confirmation No.: 7494

Examiner: Not Yet Assigned

Art Unit: Not Yet Assigned

INFORMATION DISCLOSURE
STATEMENT UNDER 37 CFR §1.97 and
§1.98

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

The references cited on attached form PTO/SB/08A and PTO/SB/08B are being called to the attention of the Examiner. Copies of the references in compliance with the requirements of 37 CFR §1.98(a)(2) are enclosed. It is respectfully requested that the cited references be expressly considered during the prosecution of this application, and the references be made of record therein and appear among the "references cited" on any patent to issue therefrom.

The references cited in this IDS were cited in an Examiner's First Report mailed on July 28, 2010, by the Australian Patent Office for priority application AU 2010202782. A

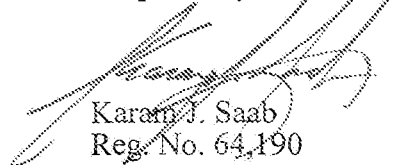
Application No.: 13/017,866
Page 2

copy of the Examiner's First Report is attached hereto as required under The Patent Prosecution Highway Pilot Program and as identified on the Request for Participation therein.

As provided for by 37 CFR §1.97(g) and (h), no inference should be made that the information and references cited are prior art merely because they are in this statement and no representation is being made that a search has been conducted or that this statement encompasses all the possible relevant information.

Applicant believes that no fee is required for submission of this statement. However, if a fee is required, the Commissioner is authorized to deduct such fee from the undersigned's Deposit Account No. 20-1430. Please deduct any additional fees from, or credit any overpayment to, the above-noted Deposit Account.

Respectfully submitted,



Karam J. Saab
Reg. No. 64,190

KILPATRICK TOWNSEND & STOCKTON LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000
Fax: 303-571-4321
K4S:cka
63128669 v1

Substitute for form 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Use as many sheets as necessary)</i>				Complete if Known	
				Application Number	13/017,866
				Filing Date	January 31, 2011
				First Named Inventor	Albert J. McGowan et al.
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	1	of	1	Attorney Docket Number	028406-000710US

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number Kind Code ² (if known)			
	A1.	US-2003-0061272	03-27-2003	Krishnamurth et al.	
	A2.	US-7,373,394	05-13-2008	Li et al.	

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No. ¹	Foreign Patent Document			Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴	Kind Code ⁵ (if known)				

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials *	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ⁶
	C1	Australian Patent Office, Examiner's First Report on Patent Application No. 2010202782 for Unicorn Media, Inc. (Atty. Ref. No. 028406-000700AU), dated July 28, 2010, 2 pages.	☐
	C2	Chang et al, Google, Inc., "Bigtable: A Distributed Storage System for Structured Data," OSDI November, 2006, pgs. 1-14, http://labs.google.com/papers/bigtable.html .	

Examiner Signature		Date Considered	
-----------------------	--	--------------------	--

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kind Codes of U.S. Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

63128669 v1



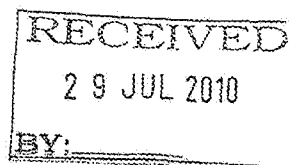
Australian Government

IP Australia

Discovery House, Phillip ACT 2606
PO Box 200, Woden ACT 2606
Australia
Phone: 1300 651 010
International Callers: +61-2 6283 2999
Facsimile: +61-2 6283 7999
Email: assist@ipaustalia.gov.au
Website: www.ipaustalia.gov.au

28 July 2010

Pizzeys
PO Box 291
Woden
Canberra ACT 2606
Australia



31560001/MER

Your Ref: 3156UNI/MRR/kj

Examiner's first report on patent application no. 2010202782
by Unicorn Media, Inc.

Last proposed amendment no.

Dear Madam/Sir,

I am replying to the request for normal examination. I have examined the application and I believe that there are lawful grounds of objection to the application. These grounds of objection are:

1. Claims 1-22 are not novel or inventive in light of each of the following documents:

D1: Google Big Table (retrieved on 27 July 2010 from the internet).
<<http://labs.google.com/papers/bigtable.html>>, published November 2006

D2: US 2003/0061272 A1 (KRISHNAMURTHY et al.), 27 March 2003

D3: US 7373394 B1 (Li et al.), 13 May 2008.
Figure 1, column 1 lines 21-55

Claim 5 appears to be the broadest claim, defining a main server with access to media assets, located separately from at least two cache servers and an application server where a user issues a request for data.

D1 discloses in general terms how cloud data persistence in Big Table works for tabular data sharing arrangements, particularly with a master server and many tablet (cache) servers (see Section 5 Implementation).

D2 discloses specific server arrangements for media distribution across the internet (see Figure 3, para [0005]-[0007], [0012]-[0013]), including a user requesting a document, checking with cache server and if necessary a master server to obtain the data requested.

D3 discloses server arrangements for media distribution across the internet (see Figure 1, column 1 lines 21-55), including a user requesting a document, checking with a geographically close cache server and if necessary an origin server to obtain the data requested.

The other claims do not seem to add any new material, for the most part being concerned with additional servers, caches and users. However, with internet wide systems already disclosed, rendering all these combinations to be also known.

NOTE: There is a current postponement of acceptance in place. If you overcome all other objections before the expiration of that postponement, the Commissioner will only accept the application at that time if you have filed a clear and unambiguous statement requesting the withdrawal of that postponement. Otherwise, a further adverse report will be issued.

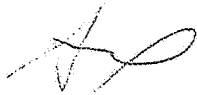
You have 21 months from the date of this report to overcome all my objection(s) otherwise your application will lapse.

You will need to pay a monthly fee for any response you file after 12 months from the date of the first report.

You will also need to pay any annual continuation fees that apply. These will normally be first due five years from the filing date. Please note however that earlier commencement dates apply for divisional applications.

Information about fees may be obtained by phoning 1300 651 010.

Yours faithfully,



ADRIAN GILLMORE
Patent Examination B
C4 - Electronics
Phone: (02) 6283 2125

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

{fay,jeff,sanjay,wilsonh,kerr,m3h,tushar,fikes,gruber}@google.com

Google, Inc.

Abstract

Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to real-time data serving). Despite these varied demands, Bigtable has successfully provided a flexible, high-performance solution for all of these Google products. In this paper we describe the simple data model provided by Bigtable, which gives clients dynamic control over data layout and format, and we describe the design and implementation of Bigtable.

1 Introduction

Over the last two and a half years we have designed, implemented, and deployed a distributed storage system for managing structured data at Google called Bigtable. Bigtable is designed to reliably scale to petabytes of data and thousands of machines. Bigtable has achieved several goals: wide applicability, scalability, high performance, and high availability. Bigtable is used by more than sixty Google products and projects, including Google Analytics, Google Finance, Orkut, Personalized Search, Writely, and Google Earth. These products use Bigtable for a variety of demanding workloads, which range from throughput-oriented batch-processing jobs to latency-sensitive serving of data to end users. The Bigtable clusters used by these products span a wide range of configurations, from a handful to thousands of servers, and store up to several hundred terabytes of data.

In many ways, Bigtable resembles a database: it shares many implementation strategies with databases. Parallel databases [14] and main-memory databases [13] have

achieved scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of the data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted strings, although clients often serialize various forms of structured and semi-structured data into these strings. Clients can control the locality of their data through careful choices in their schemas. Finally, Bigtable schema parameters let clients dynamically control whether to serve data out of memory or from disk.

Section 2 describes the data model in more detail, and Section 3 provides an overview of the client API. Section 4 briefly describes the underlying Google infrastructure on which Bigtable depends. Section 5 describes the fundamentals of the Bigtable implementation, and Section 6 describes some of the refinements that we made to improve Bigtable's performance. Section 7 provides measurements of Bigtable's performance. We describe several examples of how Bigtable is used at Google in Section 8, and discuss some lessons we learned in designing and supporting Bigtable in Section 9. Finally, Section 10 describes related work, and Section 11 presents our conclusions.

2 Data Model

A Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes.

(row:string, column:string, time:int64) → string


```
// Open the table
Table *T = OpenOrCreate("/bigtable/web/webtable");

// Write a new anchor and delete an old anchor
RowMutation rl(T, "com.cnn.www");
rl.Set("anchor:www.c-span.org", "CNN");
rl.Delete("anchor:www.abc.com");
Operation op;
Apply(&op, &rl);
```

Figure 2: Writing to Bigtable.

applications. Applications that need to avoid collisions must generate unique timestamps themselves. Different versions of a cell are stored in decreasing timestamp order, so that the most recent versions can be read first.

To make the management of versioned data less onerous, we support two per-column-family settings that tell Bigtable to garbage-collect cell versions automatically. The client can specify either that only the last n versions of a cell be kept, or that only new-enough versions be kept (e.g., only keep values that were written in the last seven days).

In our Webtable example, we set the timestamps of the crawled pages stored in the contents: column to the times at which these page versions were actually crawled. The garbage-collection mechanism described above lets us keep only the most recent three versions of every page.

3 API

The Bigtable API provides functions for creating and deleting tables and column families. It also provides functions for changing cluster, table, and column family metadata, such as access control rights.

Client applications can write or delete values in Bigtable, look up values from individual rows, or iterate over a subset of the data in a table. Figure 2 shows C++ code that uses a `RowMutation` abstraction to perform a series of updates. (Irrelevant details were elided to keep the example short.) The call to `Apply` performs an atomic mutation to the Webtable: it adds one anchor to `www.cnn.com` and deletes a different anchor.

Figure 3 shows C++ code that uses a `Scanner` abstraction to iterate over all anchors in a particular row. Clients can iterate over multiple column families, and there are several mechanisms for limiting the rows, columns, and timestamps produced by a scan. For example, we could restrict the scan above to only produce anchors whose columns match the regular expression `anchor:*.cnn.com`, or to only produce anchors whose timestamps fall within ten days of the current time.

```
Scanner scanner(T);
ScanStream *stream;
stream = scanner.FetchColumnFamily("anchor");
stream->SetReturnAllVersions();
scanner.Lookup("com.cnn.www");
for (; !stream->Done(); stream->Next()) {
    printf("%s %s %ld %s\n",
           scanner.RowName(),
           stream->ColumnName(),
           stream->MicroTimestamp(),
           stream->Value());
}
```

Figure 3: Reading from Bigtable.

Bigtable supports several other features that allow the user to manipulate data in more complex ways. First, Bigtable supports single-row transactions, which can be used to perform atomic read-modify-write sequences on data stored under a single row key. Bigtable does not currently support general transactions across row keys, although it provides an interface for batching writes across row keys at the clients. Second, Bigtable allows cells to be used as integer counters. Finally, Bigtable supports the execution of client-supplied scripts in the address spaces of the servers. The scripts are written in a language developed at Google for processing data called Sawzall [28]. At the moment, our Sawzall-based API does not allow client scripts to write back into Bigtable, but it does allow various forms of data transformation, filtering based on arbitrary expressions, and summarization via a variety of operators.

Bigtable can be used with MapReduce [12], a framework for running large-scale parallel computations developed at Google. We have written a set of wrappers that allow a Bigtable to be used both as an input source and as an output target for MapReduce jobs.

4 Building Blocks

Bigtable is built on several other pieces of Google infrastructure. Bigtable uses the distributed Google File System (GFS) [17] to store log and data files. A Bigtable cluster typically operates in a shared pool of machines that run a wide variety of other distributed applications, and Bigtable processes often share the same machines with processes from other applications. Bigtable depends on a cluster management system for scheduling jobs, managing resources on shared machines, dealing with machine failures, and monitoring machine status.

The Google *SSTable* file format is used internally to store Bigtable data. An *SSTable* provides a persistent, ordered immutable map from keys to values, where both keys and values are arbitrary byte strings. Operations are provided to look up the value associated with a specified

key, and to iterate over all key/value pairs in a specified key range. Internally, each SSTable contains a sequence of blocks (typically each block is 64KB in size, but this is configurable). A block index (stored at the end of the SSTable) is used to locate blocks; the index is loaded into memory when the SSTable is opened. A lookup can be performed with a single disk seek: we first find the appropriate block by performing a binary search in the in-memory index, and then reading the appropriate block from disk. Optionally, an SSTable can be completely mapped into memory, which allows us to perform lookups and scans without touching disk.

Bigtable relies on a highly-available and persistent distributed lock service called Chubby [8]. A Chubby service consists of five active replicas, one of which is elected to be the master and actively serve requests. The service is live when a majority of the replicas are running and can communicate with each other. Chubby uses the Paxos algorithm [9, 23] to keep its replicas consistent in the face of failure. Chubby provides a namespace that consists of directories and small files. Each directory or file can be used as a lock, and reads and writes to a file are atomic. The Chubby client library provides consistent caching of Chubby files. Each Chubby client maintains a *session* with a Chubby service. A client's session expires if it is unable to renew its session lease within the lease expiration time. When a client's session expires, it loses any locks and open handles. Chubby clients can also register callbacks on Chubby files and directories for notification of changes or session expiration.

Bigtable uses Chubby for a variety of tasks: to ensure that there is at most one active master at any time; to store the bootstrap location of Bigtable data (see Section 5.1); to discover tablet servers and finalize tablet server deaths (see Section 5.2); to store Bigtable schema information (the column family information for each table); and to store access control lists. If Chubby becomes unavailable for an extended period of time, Bigtable becomes unavailable. We recently measured this effect in 14 Bigtable clusters spanning 11 Chubby instances. The average percentage of Bigtable server hours during which some data stored in Bigtable was not available due to Chubby unavailability (caused by either Chubby outages or network issues) was 0.0047%. The percentage for the single cluster that was most affected by Chubby unavailability was 0.0326%.

5 Implementation

The Bigtable implementation has three major components: a library that is linked into every client, one master server, and many tablet servers. Tablet servers can be

dynamically added (or removed) from a cluster to accommodate changes in workloads.

The master is responsible for assigning tablets to tablet servers, detecting the addition and expiration of tablet servers, balancing tablet-server load, and garbage collection of files in GFS. In addition, it handles schema changes such as table and column family creations.

Each tablet server manages a set of tablets (typically we have somewhere between ten to a thousand tablets per tablet server). The tablet server handles read and write requests to the tablets that it has loaded, and also splits tablets that have grown too large.

As with many single-master distributed storage systems [17, 21], client data does not move through the master: clients communicate directly with tablet servers for reads and writes. Because Bigtable clients do not rely on the master for tablet location information, most clients never communicate with the master. As a result, the master is lightly loaded in practice.

A Bigtable cluster stores a number of tables. Each table consists of a set of tablets, and each tablet contains all data associated with a row range. Initially, each table consists of just one tablet. As a table grows, it is automatically split into multiple tablets, each approximately 100-200 MB in size by default.

5.1 Tablet Location

We use a three-level hierarchy analogous to that of a B⁺-tree [10] to store tablet location information (Figure 4).

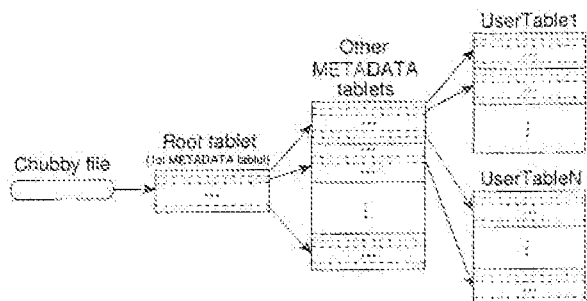


Figure 4: Tablet location hierarchy.

The first level is a file stored in Chubby that contains the location of the *root tablet*. The *root tablet* contains the location of all tablets in a special METADATA table. Each METADATA tablet contains the location of a set of user tablets. The *root tablet* is just the first tablet in the METADATA table, but is treated specially—it is never split—to ensure that the tablet location hierarchy has no more than three levels.

The METADATA table stores the location of a tablet under a row key that is an encoding of the tablet's table

identifier and its end row. Each METADATA row stores approximately 1KB of data in memory. With a modest limit of 128 MB METADATA tablets, our three-level location scheme is sufficient to address 2^{34} tablets (or 2^{61} bytes in 128 MB tablets).

The client library caches tablet locations. If the client does not know the location of a tablet, or if it discovers that cached location information is incorrect, then it recursively moves up the tablet location hierarchy. If the client's cache is empty, the location algorithm requires three network round-trips, including one read from Chubby. If the client's cache is stale, the location algorithm could take up to six round-trips, because stale cache entries are only discovered upon misses (assuming that METADATA tablets do not move very frequently). Although tablet locations are stored in memory, so no GFS accesses are required, we further reduce this cost in the common case by having the client library prefetch tablet locations: it reads the metadata for more than one tablet whenever it reads the METADATA table.

We also store secondary information in the METADATA table, including a log of all events pertaining to each tablet (such as when a server begins serving it). This information is helpful for debugging and performance analysis.

5.2 Tablet Assignment

Each tablet is assigned to one tablet server at a time. The master keeps track of the set of live tablet servers, and the current assignment of tablets to tablet servers, including which tablets are unassigned. When a tablet is unassigned, and a tablet server with sufficient room for the tablet is available, the master assigns the tablet by sending a tablet load request to the tablet server.

Bigtable uses Chubby to keep track of tablet servers. When a tablet server starts, it creates, and acquires an exclusive lock on, a uniquely-named file in a specific Chubby directory. The master monitors this directory (the *servers directory*) to discover tablet servers. A tablet server stops serving its tablets if it loses its exclusive lock: e.g., due to a network partition that caused the server to lose its Chubby session. (Chubby provides an efficient mechanism that allows a tablet server to check whether it still holds its lock without incurring network traffic.) A tablet server will attempt to reacquire an exclusive lock on its file as long as the file still exists. If the file no longer exists, then the tablet server will never be able to serve again, so it kills itself. Whenever a tablet server terminates (e.g., because the cluster management system is removing the tablet server's machine from the cluster), it attempts to release its lock so that the master will reassign its tablets more quickly.

The master is responsible for detecting when a tablet server is no longer serving its tablets, and for reassigning those tablets as soon as possible. To detect when a tablet server is no longer serving its tablets, the master periodically asks each tablet server for the status of its lock. If a tablet server reports that it has lost its lock, or if the master was unable to reach a server during its last several attempts, the master attempts to acquire an exclusive lock on the server's file. If the master is able to acquire the lock, then Chubby is live and the tablet server is either dead or having trouble reaching Chubby, so the master ensures that the tablet server can never serve again by deleting its server file. Once a server's file has been deleted, the master can move all the tablets that were previously assigned to that server into the set of unassigned tablets. To ensure that a Bigtable cluster is not vulnerable to networking issues between the master and Chubby, the master kills itself if its Chubby session expires. However, as described above, master failures do not change the assignment of tablets to tablet servers.

When a master is started by the cluster management system, it needs to discover the current tablet assignments before it can change them. The master executes the following steps at startup. (1) The master grabs a unique *master lock* in Chubby, which prevents concurrent master instantiations. (2) The master scans the servers directory in Chubby to find the live servers. (3) The master communicates with every live tablet server to discover what tablets are already assigned to each server. (4) The master scans the METADATA table to learn the set of tablets. Whenever this scan encounters a tablet that is not already assigned, the master adds the tablet to the set of unassigned tablets, which makes the tablet eligible for tablet assignment.

One complication is that the scan of the METADATA table cannot happen until the METADATA tablets have been assigned. Therefore, before starting this scan (step 4), the master adds the root tablet to the set of unassigned tablets if an assignment for the root tablet was not discovered during step 3. This addition ensures that the root tablet will be assigned. Because the root tablet contains the names of all METADATA tablets, the master knows about all of them after it has scanned the root tablet.

The set of existing tablets only changes when a tablet is created or deleted, two existing tablets are merged to form one larger tablet, or an existing tablet is split into two smaller tablets. The master is able to keep track of these changes because it initiates all but the last. Tablet splits are treated specially since they are initiated by a tablet server. The tablet server commits the split by recording information for the new tablet in the METADATA table. When the split has committed, it notifies the master. In case the split notification is lost (either

because the tablet server or the master died), the master detects the new tablet when it asks a tablet server to load the tablet that has now split. The tablet server will notify the master of the split, because the tablet entry it finds in the METADATA table will specify only a portion of the tablet that the master asked it to load.

5.3 Tablet Serving

The persistent state of a tablet is stored in GFS, as illustrated in Figure 5. Updates are committed to a commit log that stores redo records. Of these updates, the recently committed ones are stored in memory in a sorted buffer called a *memtable*; the older updates are stored in a sequence of SSTables. To recover a tablet, a tablet server

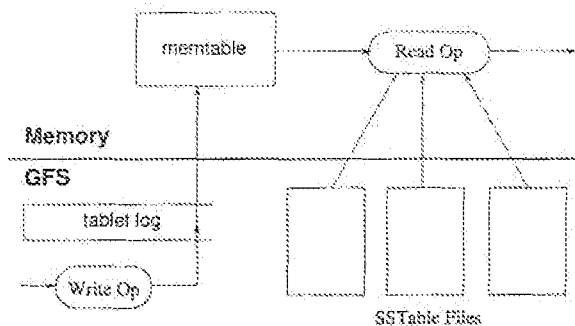


Figure 5: Tablet Representation

reads its metadata from the METADATA table. This metadata contains the list of SSTables that comprise a tablet and a set of redo points, which are pointers into any commit logs that may contain data for the tablet. The server reads the indices of the SSTables into memory and reconstructs the memtable by applying all of the updates that have committed since the redo points.

When a write operation arrives at a tablet server, the server checks that it is well-formed, and that the sender is authorized to perform the mutation. Authorization is performed by reading the list of permitted writers from a Chubby file (which is almost always a hit in the Chubby client cache). A valid mutation is written to the commit log. Group commit is used to improve the throughput of lots of small mutations [13, 16]. After the write has been committed, its contents are inserted into the memtable.

When a read operation arrives at a tablet server, it is similarly checked for well-formedness and proper authorization. A valid read operation is executed on a merged view of the sequence of SSTables and the memtable. Since the SSTables and the memtable are lexicographically sorted data structures, the merged view can be formed efficiently.

Incoming read and write operations can continue while tablets are split and merged.

5.4 Compactions

As write operations execute, the size of the memtable increases. When the memtable size reaches a threshold, the memtable is frozen, a new memtable is created, and the frozen memtable is converted to an SSTable and written to GFS. This *minor compaction* process has two goals: it shrinks the memory usage of the tablet server, and it reduces the amount of data that has to be read from the commit log during recovery if this server dies. Incoming read and write operations can continue while compactions occur.

Every minor compaction creates a new SSTable. If this behavior continued unchecked, read operations might need to merge updates from an arbitrary number of SSTables. Instead, we bound the number of such files by periodically executing a *merging compaction* in the background. A merging compaction reads the contents of a few SSTables and the memtable, and writes out a new SSTable. The input SSTables and memtable can be discarded as soon as the compaction has finished.

A merging compaction that rewrites all SSTables into exactly one SSTable is called a *major compaction*. SSTables produced by non-major compactions can contain special deletion entries that suppress deleted data in older SSTables that are still live. A major compaction, on the other hand, produces an SSTable that contains no deletion information or deleted data. Bigtable cycles through all of its tablets and regularly applies major compactions to them. These major compactions allow Bigtable to reclaim resources used by deleted data, and also allow it to ensure that deleted data disappears from the system in a timely fashion, which is important for services that store sensitive data.

6 Refinements

The implementation described in the previous section required a number of refinements to achieve the high performance, availability, and reliability required by our users. This section describes portions of the implementation in more detail in order to highlight these refinements.

Locality groups

Clients can group multiple column families together into a *locality group*. A separate SSTable is generated for each locality group in each tablet. Segregating column families that are not typically accessed together into separate locality groups enables more efficient reads. For example, page metadata in Webtable (such as language and checksums) can be in one locality group, and the contents of the page can be in a different group: an ap-

plication that wants to read the metadata does not need to read through all of the page contents.

In addition, some useful tuning parameters can be specified on a per-locality group basis. For example, a locality group can be declared to be in-memory. SSTables for in-memory locality groups are loaded lazily into the memory of the tablet server. Once loaded, column families that belong to such locality groups can be read without accessing the disk. This feature is useful for small pieces of data that are accessed frequently: we use it internally for the location column family in the METADATA table.

Compression

Clients can control whether or not the SSTables for a locality group are compressed, and if so, which compression format is used. The user-specified compression format is applied to each SSTable block (whose size is controllable via a locality group specific tuning parameter). Although we lose some space by compressing each block separately, we benefit in that small portions of an SSTable can be read without decompressing the entire file. Many clients use a two-pass custom compression scheme. The first pass uses Bentley and McIlroy's scheme [6], which compresses long common strings across a large window. The second pass uses a fast compression algorithm that looks for repetitions in a small 16 KB window of the data. Both compression passes are very fast—they encode at 100–200 MB/s, and decode at 400–1000 MB/s on modern machines.

Even though we emphasized speed instead of space reduction when choosing our compression algorithms, this two-pass compression scheme does surprisingly well. For example, in Webtable, we use this compression scheme to store Web page contents. In one experiment, we stored a large number of documents in a compressed locality group. For the purposes of the experiment, we limited ourselves to one version of each document instead of storing all versions available to us. The scheme achieved a 10-to-1 reduction in space. This is much better than typical Gzip reductions of 3-to-1 or 4-to-1 on HTML pages because of the way Webtable rows are laid out: all pages from a single host are stored close to each other. This allows the Bentley-McIlroy algorithm to identify large amounts of shared boilerplate in pages from the same host. Many applications, not just Webtable, choose their row names so that similar data ends up clustered, and therefore achieve very good compression ratios. Compression ratios get even better when we store multiple versions of the same value in Bigtable.

Caching for read performance

To improve read performance, tablet servers use two levels of caching. The Scan Cache is a higher-level cache that caches the key-value pairs returned by the SSTable interface to the tablet server code. The Block Cache is a lower-level cache that caches SSTables blocks that were read from GFS. The Scan Cache is most useful for applications that tend to read the same data repeatedly. The Block Cache is useful for applications that tend to read data that is close to the data they recently read (e.g., sequential reads, or random reads of different columns in the same locality group within a hot row).

Bloom filters

As described in Section 5.3, a read operation has to read from all SSTables that make up the state of a tablet. If these SSTables are not in memory, we may end up doing many disk accesses. We reduce the number of accesses by allowing clients to specify that Bloom filters [7] should be created for SSTables in a particular locality group. A Bloom filter allows us to ask whether an SSTable might contain any data for a specified row/column pair. For certain applications, a small amount of tablet server memory used for storing Bloom filters drastically reduces the number of disk seeks required for read operations. Our use of Bloom filters also implies that most lookups for non-existent rows or columns do not need to touch disk.

Commit-log implementation

If we kept the commit log for each tablet in a separate log file, a very large number of files would be written concurrently in GFS. Depending on the underlying file system implementation on each GFS server, these writes could cause a large number of disk seeks to write to the different physical log files. In addition, having separate log files per tablet also reduces the effectiveness of the group commit optimization, since groups would tend to be smaller. To fix these issues, we append mutations to a single commit log per tablet server, co-mingling mutations for different tablets in the same physical log file [18, 20].

Using one log provides significant performance benefits during normal operation, but it complicates recovery. When a tablet server dies, the tablets that it served will be moved to a large number of other tablet servers: each server typically loads a small number of the original server's tablets. To recover the state for a tablet, the new tablet server needs to reapply the mutations for that tablet from the commit log written by the original tablet server. However, the mutations for these tablets

were co-mingled in the same physical log file. One approach would be for each new tablet server to read this full commit log file and apply just the entries needed for the tablets it needs to recover. However, under such a scheme, if 100 machines were each assigned a single tablet from a failed tablet server, then the log file would be read 100 times (once by each server).

We avoid duplicating log reads by first sorting the commit log entries in order of the keys (table, row name, log sequence number). In the sorted output, all mutations for a particular tablet are contiguous and can therefore be read efficiently with one disk seek followed by a sequential read. To parallelize the sorting, we partition the log file into 64 MB segments, and sort each segment in parallel on different tablet servers. This sorting process is coordinated by the master and is initiated when a tablet server indicates that it needs to recover mutations from some commit log file.

Writing commit logs to GFS sometimes causes performance hiccups for a variety of reasons (e.g., a GFS server machine involved in the write crashes, or the network paths traversed to reach the particular set of three GFS servers is suffering network congestion, or is heavily loaded). To protect mutations from GFS latency spikes, each tablet server actually has two log writing threads, each writing to its own log file; only one of these two threads is actively in use at a time. If writes to the active log file are performing poorly, the log file writing is switched to the other thread, and mutations that are in the commit log queue are written by the newly active log writing thread. Log entries contain sequence numbers to allow the recovery process to elide duplicated entries resulting from this log switching process.

Speeding up tablet recovery

If the master moves a tablet from one tablet server to another, the source tablet server first does a minor compaction on that tablet. This compaction reduces recovery time by reducing the amount of uncompact state in the tablet server's commit log. After finishing this compaction, the tablet server stops serving the tablet. Before it actually unloads the tablet, the tablet server does another (usually very fast) minor compaction to eliminate any remaining uncompact state in the tablet server's log that arrived while the first minor compaction was being performed. After this second minor compaction is complete, the tablet can be loaded on another tablet server without requiring any recovery of log entries.

Exploiting immutability

Besides the SSTable caches, various other parts of the Bigtable system have been simplified by the fact that all

of the SSTables that we generate are immutable. For example, we do not need any synchronization of accesses to the file system when reading from SSTables. As a result, concurrency control over rows can be implemented very efficiently. The only mutable data structure that is accessed by both reads and writes is the memtable. To reduce contention during reads of the memtable, we make each memtable row copy-on-write and allow reads and writes to proceed in parallel.

Since SSTables are immutable, the problem of permanently removing deleted data is transformed to garbage collecting obsolete SSTables. Each tablet's SSTables are registered in the METADATA table. The master removes obsolete SSTables as a mark-and-sweep garbage collection [25] over the set of SSTables, where the METADATA table contains the set of roots.

Finally, the immutability of SSTables enables us to split tablets quickly. Instead of generating a new set of SSTables for each child tablet, we let the child tablets share the SSTables of the parent tablet.

7 Performance Evaluation

We set up a Bigtable cluster with N tablet servers to measure the performance and scalability of Bigtable as N is varied. The tablet servers were configured to use 1 GB of memory and to write to a GFS cell consisting of 1786 machines with two 400 GB IDE hard drives each. N client machines generated the Bigtable load used for these tests. (We used the same number of clients as tablet servers to ensure that clients were never a bottleneck.) Each machine had two dual-core Opteron 2 GHz chips, enough physical memory to hold the working set of all running processes, and a single gigabit Ethernet link. The machines were arranged in a two-level tree-shaped switched network with approximately 100-200 Gbps of aggregate bandwidth available at the root. All of the machines were in the same hosting facility and therefore the round-trip time between any pair of machines was less than a millisecond.

The tablet servers and master, test clients, and GFS servers all ran on the same set of machines. Every machine ran a GFS server. Some of the machines also ran either a tablet server, or a client process, or processes from other jobs that were using the pool at the same time as these experiments.

R is the distinct number of Bigtable row keys involved in the test. R was chosen so that each benchmark read or wrote approximately 1 GB of data per tablet server.

The *sequential write* benchmark used row keys with names 0 to $R - 1$. This space of row keys was partitioned into $10N$ equal-sized ranges. These ranges were assigned to the N clients by a central scheduler that as-

Experiment	# of Tablet Servers			
	1	50	250	500
random reads	1212	593	479	241
random reads (mem)	10811	8511	8000	6250
random writes	8850	3745	3425	2000
sequential reads	4425	2463	2625	2469
sequential writes	8547	3623	2451	1905
scans	15385	10526	9524	7843

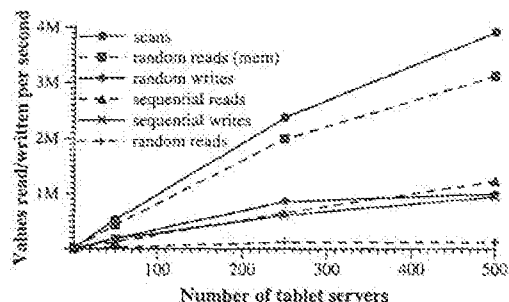


Figure 6: Number of 1000-byte values read/written per second. The table shows the rate per tablet server; the graph shows the aggregate rate.

signed the next available range to a client as soon as the client finished processing the previous range assigned to it. This dynamic assignment helped mitigate the effects of performance variations caused by other processes running on the client machines. We wrote a single string under each row key. Each string was generated randomly and was therefore uncompressible. In addition, strings under different row key were distinct, so no cross-row compression was possible. The *random write* benchmark was similar except that the row key was hashed modulo H immediately before writing so that the write load was spread roughly uniformly across the entire row space for the entire duration of the benchmark.

The *sequential read* benchmark generated row keys in exactly the same way as the sequential write benchmark, but instead of writing under the row key, it read the string stored under the row key (which was written by an earlier invocation of the sequential write benchmark). Similarly, the *random read* benchmark shadowed the operation of the random write benchmark.

The *scan* benchmark is similar to the sequential read benchmark, but uses support provided by the Bigtable API for scanning over all values in a row range. Using a scan reduces the number of RPCs executed by the benchmark since a single RPC fetches a large sequence of values from a tablet server.

The *random reads (mem)* benchmark is similar to the random read benchmark, but the locality group that contains the benchmark data is marked as *in-memory*, and therefore the reads are satisfied from the tablet server's memory instead of requiring a GFS read. For just this benchmark, we reduced the amount of data per tablet server from 1 GB to 100 MB so that it would fit comfortably in the memory available to the tablet server.

Figure 6 shows two views on the performance of our benchmarks when reading and writing 1000-byte values to Bigtable. The table shows the number of operations per second per tablet server; the graph shows the aggregate number of operations per second.

Single tablet-server performance

Let us first consider performance with just one tablet server. Random reads are slower than all other operations by an order of magnitude or more. Each random read involves the transfer of a 64 KB SSTable block over the network from GFS to a tablet server, out of which only a single 1000-byte value is used. The tablet server executes approximately 1200 reads per second, which translates into approximately 75 MB/s of data read from GFS. This bandwidth is enough to saturate the tablet server CPUs because of overheads in our networking stack, SSTable parsing, and Bigtable code, and is also almost enough to saturate the network links used in our system. Most Bigtable applications with this type of an access pattern reduce the block size to a smaller value, typically 8KB.

Random reads from memory are much faster since each 1000-byte read is satisfied from the tablet server's local memory without fetching a large 64 KB block from GFS.

Random and sequential writes perform better than random reads since each tablet server appends all incoming writes to a single commit log and uses group commit to stream these writes efficiently to GFS. There is no significant difference between the performance of random writes and sequential writes; in both cases, all writes to the tablet server are recorded in the same commit log.

Sequential reads perform better than random reads since every 64 KB SSTable block that is fetched from GFS is stored into our block cache, where it is used to serve the next 64 read requests.

Scans are even faster since the tablet server can return a large number of values in response to a single client RPC, and therefore RPC overhead is amortized over a large number of values.

Scaling

Aggregate throughput increases dramatically, by over a factor of a hundred, as we increase the number of tablet servers in the system from 1 to 500. For example, the

# of tablet servers	# of clusters
0 .. 19	259
20 .. 49	47
50 .. 99	20
100 .. 499	50
> 500	12

Table 1: Distribution of number of tablet servers in Bigtable clusters.

performance of random reads from memory increases by almost a factor of 300 as the number of tablet server increases by a factor of 500. This behavior occurs because the bottleneck on performance for this benchmark is the individual tablet server CPU.

However, performance does not increase linearly. For most benchmarks, there is a significant drop in per-server throughput when going from 1 to 50 tablet servers. This drop is caused by imbalance in load in multiple server configurations, often due to other processes contending for CPU and network. Our load balancing algorithm attempts to deal with this imbalance, but cannot do a perfect job for two main reasons: rebalancing is throttled to reduce the number of tablet movements (a tablet is unavailable for a short time, typically less than one second, when it is moved), and the load generated by our benchmarks shifts around as the benchmark progresses.

The random read benchmark shows the worst scaling (an increase in aggregate throughput by only a factor of 100 for a 500-fold increase in number of servers). This behavior occurs because (as explained above) we transfer one large 64KB block over the network for every 1000-byte read. This transfer saturates various shared 1 Gigabit links in our network and as a result, the per-server throughput drops significantly as we increase the number of machines.

8 Real Applications

As of August 2006, there are 388 non-test Bigtable clusters running in various Google machine clusters, with a combined total of about 24,500 tablet servers. Table 1 shows a rough distribution of tablet servers per cluster. Many of these clusters are used for development purposes and therefore are idle for significant periods. One group of 14 busy clusters with 8069 total tablet servers saw an aggregate volume of more than 1.2 million requests per second, with incoming RPC traffic of about 741 MB/s and outgoing RPC traffic of about 16 GB/s.

Table 2 provides some data about a few of the tables currently in use. Some tables store data that is served to users, whereas others store data for batch processing; the tables range widely in total size, average cell size,

percentage of data served from memory, and complexity of the table schema. In the rest of this section, we briefly describe how three product teams use Bigtable.

8.1 Google Analytics

Google Analytics (analytics.google.com) is a service that helps webmasters analyze traffic patterns at their web sites. It provides aggregate statistics, such as the number of unique visitors per day and the page views per URL per day, as well as site-tracking reports, such as the percentage of users that made a purchase, given that they earlier viewed a specific page.

To enable the service, webmasters embed a small JavaScript program in their web pages. This program is invoked whenever a page is visited. It records various information about the request in Google Analytics, such as a user identifier and information about the page being fetched. Google Analytics summarizes this data and makes it available to webmasters.

We briefly describe two of the tables used by Google Analytics. The raw click table (~200 TB) maintains a row for each end-user session. The row name is a tuple containing the website's name and the time at which the session was created. This schema ensures that sessions that visit the same web site are contiguous, and that they are sorted chronologically. This table compresses to 14% of its original size.

The summary table (~20 TB) contains various predefined summaries for each website. This table is generated from the raw click table by periodically scheduled MapReduce jobs. Each MapReduce job extracts recent session data from the raw click table. The overall system's throughput is limited by the throughput of GPS. This table compresses to 29% of its original size.

8.2 Google Earth

Google operates a collection of services that provide users with access to high-resolution satellite imagery of the world's surface, both through the web-based Google Maps interface (maps.google.com) and through the Google Earth (earth.google.com) custom client software. These products allow users to navigate across the world's surface: they can pan, view, and annotate satellite imagery at many different levels of resolution. This system uses one table to preprocess data, and a different set of tables for serving client data.

The preprocessing pipeline uses one table to store raw imagery. During preprocessing, the imagery is cleaned and consolidated into final serving data. This table contains approximately 70 terabytes of data and therefore is served from disk. The images are efficiently compressed already, so Bigtable compression is disabled.

Project name	Table size (TB)	Compression ratio	# Cells (billions)	# Column Families	# Locality Groups	% in memory	Latency-sensitive?
<i>Crawl</i>	800	11%	1000	16	8	0%	No
<i>Crawl</i>	50	33%	200	2	2	0%	No
<i>Google Analytics</i>	20	29%	10	1	1	0%	Yes
<i>Google Analytics</i>	200	14%	80	1	1	0%	Yes
<i>Google Base</i>	2	31%	10	29	3	15%	Yes
<i>Google Earth</i>	0.5	64%	8	7	2	33%	Yes
<i>Google Earth</i>	70	–	9	8	3	0%	No
<i>Orkut</i>	9	–	0.9	8	5	1%	Yes
<i>Personalized Search</i>	4	47%	6	93	11	5%	Yes

Table 2: Characteristics of a few tables in production use. *Table size* (measured before compression) and *# Cells* indicate approximate sizes. *Compression ratio* is not given for tables that have compression disabled.

Each row in the imagery table corresponds to a single geographic segment. Rows are named to ensure that adjacent geographic segments are stored near each other. The table contains a column family to keep track of the sources of data for each segment. This column family has a large number of columns: essentially one for each raw data image. Since each segment is only built from a few images, this column family is very sparse.

The preprocessing pipeline relies heavily on MapReduce over Bigtable to transform data. The overall system processes over 1 MB/sec of data per tablet server during some of these MapReduce jobs.

The serving system uses one table to index data stored in GFS. This table is relatively small (~500 GB), but it must serve tens of thousands of queries per second per datacenter with low latency. As a result, this table is hosted across hundreds of tablet servers and contains in-memory column families.

8.3 Personalized Search

Personalized Search (www.google.com/psearch) is an opt-in service that records user queries and clicks across a variety of Google properties such as web search, images, and news. Users can browse their search histories to revisit their old queries and clicks, and they can ask for personalized search results based on their historical Google usage patterns.

Personalized Search stores each user's data in Bigtable. Each user has a unique userid and is assigned a row named by that userid. All user actions are stored in a table. A separate column family is reserved for each type of action (for example, there is a column family that stores all web queries). Each data element uses as its Bigtable timestamp the time at which the corresponding user action occurred. Personalized Search generates user profiles using a MapReduce over Bigtable. These user profiles are used to personalize live search results.

The Personalized Search data is replicated across several Bigtable clusters to increase availability and to reduce latency due to distance from clients. The Personalized Search team originally built a client-side replication mechanism on top of Bigtable that ensured eventual consistency of all replicas. The current system now uses a replication subsystem that is built into the servers.

The design of the Personalized Search storage system allows other groups to add new per-user information in their own columns, and the system is now used by many other Google properties that need to store per-user configuration options and settings. Sharing a table amongst many groups resulted in an unusually large number of column families. To help support sharing, we added a simple quota mechanism to Bigtable to limit the storage consumption by any particular client in shared tables; this mechanism provides some isolation between the various product groups using this system for per-user information storage.

9 Lessons

In the process of designing, implementing, maintaining, and supporting Bigtable, we gained useful experience and learned several interesting lessons.

One lesson we learned is that large distributed systems are vulnerable to many types of failures, not just the standard network partitions and fail-stop failures assumed in many distributed protocols. For example, we have seen problems due to all of the following causes: memory and network corruption, large clock skew, hung machines, extended and asymmetric network partitions, bugs in other systems that we are using (Chubby for example), overflow of GFS quotas, and planned and unplanned hardware maintenance. As we have gained more experience with these problems, we have addressed them by changing various protocols. For example, we added checksumming to our RPC mechanism. We also handled

some problems by removing assumptions made by one part of the system about another part. For example, we stopped assuming a given Chubby operation could return only one of a fixed set of errors.

Another lesson we learned is that it is important to delay adding new features until it is clear how the new features will be used. For example, we initially planned to support general-purpose transactions in our API. Because we did not have an immediate use for them, however, we did not implement them. Now that we have many real applications running on Bigtable, we have been able to examine their actual needs, and have discovered that most applications require only single-row transactions. Where people have requested distributed transactions, the most important use is for maintaining secondary indices, and we plan to add a specialized mechanism to satisfy this need. The new mechanism will be less general than distributed transactions, but will be more efficient (especially for updates that span hundreds of rows or more) and will also interact better with our scheme for optimistic cross-data-center replication.

A practical lesson that we learned from supporting Bigtable is the importance of proper system-level monitoring (i.e., monitoring both Bigtable itself, as well as the client processes using Bigtable). For example, we extended our RPC system so that for a sample of the RPCs, it keeps a detailed trace of the important actions done on behalf of that RPC. This feature has allowed us to detect and fix many problems such as lock contention on tablet data structures, slow writes to GFS while committing Bigtable mutations, and stuck accesses to the METADATA table when METADATA tablets are unavailable. Another example of useful monitoring is that every Bigtable cluster is registered in Chubby. This allows us to track down all clusters, discover how big they are, see which versions of our software they are running, how much traffic they are receiving, and whether or not there are any problems such as unexpectedly large latencies.

The most important lesson we learned is the value of simple designs. Given both the size of our system (about 100,000 lines of non-test code), as well as the fact that code evolves over time in unexpected ways, we have found that code and design clarity are of immense help in code maintenance and debugging. One example of this is our tablet-server membership protocol. Our first protocol was simple: the master periodically issued leases to tablet servers, and tablet servers killed themselves if their lease expired. Unfortunately, this protocol reduced availability significantly in the presence of network problems, and was also sensitive to master recovery time. We redesigned the protocol several times until we had a protocol that performed well. However, the resulting protocol was too complex and depended on

the behavior of Chubby features that were seldom exercised by other applications. We discovered that we were spending an inordinate amount of time debugging obscure corner cases, not only in Bigtable code, but also in Chubby code. Eventually, we scrapped this protocol and moved to a newer simpler protocol that depends solely on widely-used Chubby features.

10 Related Work

The Boxwood project [24] has components that overlap in some ways with Chubby, GFS, and Bigtable, since it provides for distributed agreement, locking, distributed chunk storage, and distributed B-tree storage. In each case where there is overlap, it appears that the Boxwood's component is targeted at a somewhat lower level than the corresponding Google service. The Boxwood project's goal is to provide infrastructure for building higher-level services such as file systems or databases, while the goal of Bigtable is to directly support client applications that wish to store data.

Many recent projects have tackled the problem of providing distributed storage or higher-level services over wide area networks, often at "Internet scale." This includes work on distributed hash tables that began with projects such as CAN [29], Chord [32], Tapestry [37], and Pastry [30]. These systems address concerns that do not arise for Bigtable, such as highly variable bandwidth, untrusted participants, or frequent reconfiguration; decentralized control and Byzantine fault tolerance are not Bigtable goals.

In terms of the distributed data storage model that one might provide to application developers, we believe the key-value pair model provided by distributed B-trees or distributed hash tables is too limiting. Key-value pairs are a useful building block, but they should not be the only building block one provides to developers. The model we chose is richer than simple key-value pairs, and supports sparse semi-structured data. Nonetheless, it is still simple enough that it lends itself to a very efficient flat-file representation, and it is transparent enough (via locality groups) to allow our users to tune important behaviors of the system.

Several database vendors have developed parallel databases that can store large volumes of data. Oracle's Real Application Cluster database [27] uses shared disks to store data (Bigtable uses GFS) and a distributed lock manager (Bigtable uses Chubby). IBM's DB2 Parallel Edition [4] is based on a shared-nothing [33] architecture similar to Bigtable. Each DB2 server is responsible for a subset of the rows in a table which it stores in a local relational database. Both products provide a complete relational model with transactions.

Bigtable locality groups realize similar compression and disk read performance benefits observed for other systems that organize data on disk using column-based rather than row-based storage, including C-Store [1, 34] and commercial products such as Sybase IQ [15, 36], SenSage [31], KDB+ [22], and the ColumnBM storage layer in MonetDB/X100 [38]. Another system that does vertical and horizontal data partitioning into flat files and achieves good data compression ratios is AT&T's Daytona database [19]. Locality groups do not support CPU-cache-level optimizations, such as those described by Ailamaki [2].

The manner in which Bigtable uses memtables and SSTables to store updates to tablets is analogous to the way that the Log-Structured Merge Tree [26] stores updates to index data. In both systems, sorted data is buffered in memory before being written to disk, and reads must merge data from memory and disk.

C-Store and Bigtable share many characteristics: both systems use a shared-nothing architecture and have two different data structures, one for recent writes, and one for storing long-lived data, with a mechanism for moving data from one form to the other. The systems differ significantly in their API: C-Store behaves like a relational database, whereas Bigtable provides a lower level read and write interface and is designed to support many thousands of such operations per second per server. C-Store is also a "read-optimized relational DBMS", whereas Bigtable provides good performance on both read-intensive and write-intensive applications.

Bigtable's load balancer has to solve some of the same kinds of load and memory balancing problems faced by shared-nothing databases (e.g., [11, 35]). Our problem is somewhat simpler: (1) we do not consider the possibility of multiple copies of the same data, possibly in alternate forms due to views or indices; (2) we let the user tell us what data belongs in memory and what data should stay on disk, rather than trying to determine this dynamically; (3) we have no complex queries to execute or optimize.

11 Conclusions

We have described Bigtable, a distributed system for storing structured data at Google. Bigtable clusters have been in production use since April 2005, and we spent roughly seven person-years on design and implementation before that date. As of August 2006, more than sixty projects are using Bigtable. Our users like the performance and high availability provided by the Bigtable implementation, and that they can scale the capacity of their clusters by simply adding more machines to the system as their resource demands change over time.

Given the unusual interface to Bigtable, an interesting question is how difficult it has been for our users to adapt to using it. New users are sometimes uncertain of how to best use the Bigtable interface, particularly if they are accustomed to using relational databases that support general-purpose transactions. Nevertheless, the fact that many Google products successfully use Bigtable demonstrates that our design works well in practice.

We are in the process of implementing several additional Bigtable features, such as support for secondary indices and infrastructure for building cross-data-center replicated Bigtables with multiple master replicas. We have also begun deploying Bigtable as a service to product groups, so that individual groups do not need to maintain their own clusters. As our service clusters scale, we will need to deal with more resource-sharing issues within Bigtable itself [3, 5].

Finally, we have found that there are significant advantages to building our own storage solution at Google. We have gotten a substantial amount of flexibility from designing our own data model for Bigtable. In addition, our control over Bigtable's implementation, and the other Google infrastructure upon which Bigtable depends, means that we can remove bottlenecks and inefficiencies as they arise.

Acknowledgements

We thank the anonymous reviewers, David Nagle, and our shepherd Brad Calder, for their feedback on this paper. The Bigtable system has benefited greatly from the feedback of our many users within Google. In addition, we thank the following people for their contributions to Bigtable: Dan Aguayo, Sameer Ajmani, Zhifeng Chen, Bill Coughran, Mike Epstein, Healdene Goguen, Robert Griesemer, Jeremy Hylton, Josh Hyman, Alex Khesin, Joanna Kulik, Alberto Lerner, Sherry Listgarten, Mike Maloney, Eduardo Pinheiro, Kathy Polizzi, Frank Yellin, and Arthur Zweiginczew.

References

- [1] ABADI, D. J., MADDEN, S. R., AND FERREIRA, M. C. Integrating compression and execution in column-oriented database systems. *Proc. of SIGMOD* (2006).
- [2] AILAMAKI, A., DEWITT, D. J., HILL, M. D., AND SKOUNAKIS, M. Weaving relations for cache performance. In *The VLDB Journal* (2001), pp. 169–180.
- [3] BANGA, G., DRUSCHEL, P., AND MOGUL, J. C. Resource containers: A new facility for resource management in server systems. In *Proc. of the 3rd OSDI* (Feb. 1999), pp. 45–58.
- [4] BARU, C. K., FECTEAU, G., GOYAL, A., HSIAO, H., JHINGRAN, A., PADMANABHAN, S., COPELAND,

- G. P., AND WILSON, W. G. DB2 parallel edition. *IBM Systems Journal* 34, 2 (1995), 292–322.
- [5] BAVIER, A., BOWMAN, M., CHUN, B., CULLER, D., KARLIN, S., PETERSON, L., ROSCOE, T., SPALINK, T., AND WAWRZONIAK, M. Operating system support for planetary-scale network services. In *Proc. of the 1st NSDI* (Mar. 2004), pp. 253–266.
- [6] BENTLEY, J. L., AND MCILROY, M. D. Data compression using long common strings. In *Data Compression Conference* (1999), pp. 287–295.
- [7] BLOOM, B. H. Space/time trade-offs in hash coding with allowable errors. *CACM* 13, 7 (1970), 422–426.
- [8] BURROWS, M. The Chubby lock service for loosely-coupled distributed systems. In *Proc. of the 7th OSDI* (Nov. 2006).
- [9] CHANDRA, T., GRIESEMER, R., AND REDSTONE, J. Paxos made live — An engineering perspective. In *Proc. of PODC* (2007).
- [10] COMER, D. Ubiquitous B-trees. *Computing Surveys* 11, 2 (June 1979), 121–137.
- [11] COPELAND, G. P., ALEXANDER, W., BOUGHTER, E. E., AND KELLER, T. W. Data placement in Bubba. In *Proc. of SIGMOD* (1988), pp. 99–108.
- [12] DEAN, J., AND GHEMAWAT, S. MapReduce: Simplified data processing on large clusters. In *Proc. of the 6th OSDI* (Dec. 2004), pp. 137–156.
- [13] DEWITT, D., KATZ, R., OLKEN, F., SHAPIRO, L., STONEBRAKER, M., AND WOOD, D. Implementation techniques for main memory database systems. In *Proc. of SIGMOD* (June 1984), pp. 1–8.
- [14] DEWITT, D. J., AND GRAY, J. Parallel database systems: The future of high performance database systems. *CACM* 35, 6 (June 1992), 85–98.
- [15] FRENCH, C. D. One size fits all database architectures do not work for DSS. In *Proc. of SIGMOD* (May 1995), pp. 449–450.
- [16] GAWLICK, D., AND KINKADE, D. Varieties of concurrency control in IMS/VS fast path. *Database Engineering Bulletin* 8, 2 (1985), 3–10.
- [17] GHEMAWAT, S., GOBIOFF, H., AND LEUNG, S.-T. The Google file system. In *Proc. of the 19th ACM SOSP* (Dec. 2003), pp. 29–43.
- [18] GRAY, J. Notes on database operating systems. In *Operating Systems — An Advanced Course*, vol. 60 of *Lecture Notes in Computer Science*. Springer-Verlag, 1978.
- [19] GREER, R. Daytona and the fourth-generation language Cymbal. In *Proc. of SIGMOD* (1999), pp. 525–526.
- [20] HAGMANN, R. Reimplementing the Cedar file system using logging and group commit. In *Proc. of the 11th SOSP* (Dec. 1987), pp. 155–162.
- [21] HARTMAN, J. H., AND OUSTERHOUT, J. K. The Zebra striped network file system. In *Proc. of the 14th SOSP* (Asheville, NC, 1993), pp. 29–43.
- [22] KX.COM. kx.com/products/database.php. Product page.
- [23] LAMPORT, L. The part-time parliament. *ACM TOCS* 16, 2 (1998), 133–169.
- [24] MACCORMICK, J., MURPHY, N., NAJORK, M., THEKKATH, C. A., AND ZHOU, L. Boxwood: Abstractions as the foundation for storage infrastructure. In *Proc. of the 6th OSDI* (Dec. 2004), pp. 105–120.
- [25] MCCARTHY, J. Recursive functions of symbolic expressions and their computation by machine. *CACM* 3, 4 (Apr. 1960), 184–195.
- [26] O'NEIL, P., CHENG, E., GAWLICK, D., AND O'NEIL, E. The log-structured merge-tree (LSM-tree). *Acta Inf.* 33, 4 (1996), 351–385.
- [27] ORACLE.COM. www.oracle.com/technology/products/database/clustering/index.html. Product page.
- [28] PIKE, R., DORWARD, S., GRIESEMER, R., AND QUINLAN, S. Interpreting the data: Parallel analysis with Sawzall. *Scientific Programming Journal* 13, 4 (2005), 227–298.
- [29] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable content-addressable network. In *Proc. of SIGCOMM* (Aug. 2001), pp. 161–172.
- [30] ROWSTRON, A., AND DRUSCHEL, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of Middleware 2001* (Nov. 2001), pp. 329–350.
- [31] SENSAGE.COM. sensation.com/products-sensation.htm. Product page.
- [32] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. of SIGCOMM* (Aug. 2001), pp. 149–160.
- [33] STONEBRAKER, M. The case for shared nothing. *Database Engineering Bulletin* 9, 1 (Mar. 1986), 4–9.
- [34] STONEBRAKER, M., ABADI, D. J., BATKIN, A., CHEN, X., CHERNIACK, M., FERREIRA, M., LAU, E., LIN, A., MADDEN, S., O'NEIL, E., O'NEIL, P., RASIN, A., TRAN, N., AND ZDONIK, S. C-Store: A column-oriented DBMS. In *Proc. of VLDB* (Aug. 2005), pp. 553–564.
- [35] STONEBRAKER, M., AOKI, P. M., DEVINE, R., LITWIN, W., AND OLSON, M. A. Mariposa: A new architecture for distributed data. In *Proc. of the Tenth ICDE* (1994), IEEE Computer Society, pp. 54–65.
- [36] SYBASE.COM. www.sybase.com/products/database-servers/sybaseiq. Product page.
- [37] ZHAO, B. Y., KUBIATOWICZ, J., AND JOSEPH, A. D. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. Rep. UCB/CSD-01-1141, CS Division, UC Berkeley, Apr. 2001.
- [38] ZUKOWSKI, M., BORCZ, P. A., NES, N., AND HEMAN, S. MonetDB/X100 — A DBMS in the CPU cache. *IEEE Data Eng. Bull.* 28, 2 (2005), 17–22.

Electronic Patent Application Fee Transmittal				
Application Number:		13017866		
Filing Date:				
Title of Invention:		CLOUD DATA PERSISTENCE ENGINE		
First Named Inventor/Applicant Name:		Albert J. McGowan		
Filer:		Karam J. Saab/Corina Aschenbrenner		
Attorney Docket Number:		028406-000710US		
Filed as Small Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Petition fee- 37 CFR 1.17(h) (Group III)	1464	1	130	130
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				130

Electronic Acknowledgement Receipt

EFS ID:	9347060
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab
Filer Authorized By:	
Attorney Docket Number:	028406-000710US
Receipt Date:	31-JAN-2011
Filing Date:	
Time Stamp:	18:16:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$ 130
RAM confirmation Number	6120
Deposit Account	201430
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Petition to make special under Patent Prosecution Hwy	PPHRequest.pdf	13401934	no	15
			99b4d6831dcf983eae8c89a6c42e68ef0ee302cf		

Warnings:

Information:

2	NPL Documents	NPL_Google_BigtableADistributedStorageSystemforStructuredData.pdf	8119932	no	14
			696311c3146cce97b066eba6b386d243dda87695		

Warnings:

Information:

3	Fee Worksheet (PTO-875)	fee-info.pdf	30062	no	2
			6d27f654c62057045b3ab5ed6569c1ddcab995ef		

Warnings:

Information:

Total Files Size (in bytes):			21551928
------------------------------	--	--	----------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	028406-000710US

CONFIRMATION NO. 7494

FORMALITIES LETTER



0000000045946141

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Date Mailed: 02/16/2011

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is missing.
A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- A surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of \$65 for a small entity in compliance with 37 CFR 1.27, must be submitted.

SUMMARY OF FEES DUE:

Total fee(s) required within **TWO MONTHS** from the date of this Notice is **\$65** for a small entity

- **\$65** Surcharge.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/t1e/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
13/017,866	01/31/2011	2447	462	028406-000710US	20	3

CONFIRMATION NO. 7494

FILING RECEIPT



OC000000045946140

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Date Mailed: 02/16/2011

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Albert J. McGowan, Phoenix, AZ;
Richard L. Carls, Scottsdale, AZ;

Assignment For Published Patent Application

Unicorn Media, Inc., Tempe, AZ

Power of Attorney: None

Domestic Priority data as claimed by applicant

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)

AUSTRALIA 2010202782 07/01/2010

Request to Retrieve - This application either claims priority to one or more applications filed in an intellectual property Office that participates in the Priority Document Exchange (PDX) program or contains a proper **Request to Retrieve Electronic Priority Application(s)** (PTO/SB/38 or its equivalent). Consequently, the USPTO will attempt to electronically retrieve these priority documents.

If Required, Foreign Filing License Granted: 02/10/2011

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/017,866**

Projected Publication Date: To Be Determined - pending completion of Missing Parts

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

CLOUD DATA PERSISTENCE ENGINE

Preliminary Class

709

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER**Title 35, United States Code, Section 184****Title 37, Code of Federal Regulations, 5.11 & 5.15****GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as

set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 13/017,866			
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	82	N/A			
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	270	N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	110	N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	20	minus 20 =	*	x 26 =	0.00	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3	minus 3 =	*	x 110 =	0.00				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$270 (\$135 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	462	TOTAL			
APPLICATION AS AMENDED - PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =	OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =	OR	x =	
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	028406-000710US	7494

20350 7590 03/09/2011
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER

ART UNIT	PAPER NUMBER
2447	

NOTIFICATION DATE	DELIVERY MODE
03/09/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

Docket@kilpatricktownsend.com
ipefiling@kilpatricktownsend.com
jlhice@kilpatrick.foundationip.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

KILPATRICK TOWNSEND &
STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO CA 94111-3834

In re Application of: McGowan et al.
Application No. 13/017866
Filed: January 31, 2011
For: **CLOUD DATA PERSISTENCE
ENGINE**

DECISION ON REQUEST TO
PARTICIPATE IN PATENT
PROSECUTION HIGHWAY
PROGRAM AND PETITION TO
MAKE SPECIAL UNDER 37 CFR
1.102(d)

MAILED

MAR 09 2011

DIRECTOR OFFICE
TECHNOLOGY CENTER 2400

This is a decision on the renewed request to participate in the Patent Prosecution Highway (PPH) program and the petition under 37 CFR 1.102(d), filed January 31, 2011, to make the above-identified application special.

The petition is **GRANTED**.

A grantable request to participate in the PPH program and petition to make special require:

(1) The U.S. application is

- (a) a Paris Convention application which either (i) validly claims priority under 35 U.S.C. 119(a) and 37 CFR 1.55 to one or more applications filed with IPAU, or (ii) validly claims priority under 35 U.S.C. 119(a)/365(a) to a PCT application that contains no priority claims, or
- (b) a national stage application under the PCT (an application which entered the national stage in the U.S. from a PCT international application after compliance with 35 U.S.C. 371), which PCT application (i) validly claims priority under 35 U.S.C. 365(b) to an application filed with IPAU, or (ii) validly claims priority under 35 U.S.C. 365(b) to a PCT application that contains no priority claims, or (iii) contains no priority claim, or
- (c) a so-called bypass application filed under 35 U.S.C. 111(a) which validly claims benefit under 35 U.S.C. 120 to a PCT application, which PCT application (i) validly claims priority under 35 U.S.C. 365(b) to an application filed with IPAU, or (ii) validly claims priority under 35 U.S.C. 365(b) to a PCT application that contains no priority claims, or (iii) contains no priority claim.

(2) The IPAU application(s) has at least one claim that was determined by IPAU to be allowable. Applicant must submit a copy of the allowable claims from the IPAU application(s). If the IPAU office action does not explicitly state that a particular claim is allowable, applicant must include a statement in the request for participation in the PPH pilot program that no rejection has been made in the IPAU office action regarding that claim and therefore, the claim is deemed allowable by IPAU.

(3) All the claims in each U.S. application for which a request for participation in the PPH pilot program is made must sufficiently correspond or be amended to sufficiently correspond to the allowable claims in the IPAU application(s). Claims will be considered to sufficiently correspond where, accounting for differences due to claim format requirements, the claims are of the same or similar scope. Applicant is also required to submit a claims correspondence table in English. The claims correspondence table must indicate how all the claims in the U.S. application correspond to the allowable claims in the IPAU application(s).

(4) Examination of the U.S. application for which participation in the PPH pilot program is requested has not begun.

(5) Applicant must submit a copy of all the office actions (which are relevant to patentability) from each of the IPAU application(s) containing the allowable claims that are the basis for the request. In addition, applicant must submit copies of any office actions (which are relevant to patentability) from the IPAU application(s) issued after the grant of the request for participation in the PPH pilot program in the USPTO (especially where IPAU might have reversed a prior holding of allowability).

(6) Applicant must submit an information disclosure statement (IDS) listing the documents cited by the IPAU examiner in the IPAU office action (unless such an IDS has already been filed in the U.S. application). Applicant must submit copies of all the documents cited in the IPAU office action (unless the copies have already been filed in the U.S. application) except U.S. patents or U.S. patent application publications.

The request to participate in the PPH program and petition are found to comply with all the above requirements. Accordingly, the above-identified application has been accorded "special" status.

Telephone inquiries concerning this decision should be directed to Christopher Grant at 571-272-7294.

All other inquiries concerning the examination or status of the application is accessible in the PAIR system at <http://www.uspto.gov/ebc/index.html>.

The application is undergoing pre-examination processing. Once it is released for examination, the application will be forwarded to the examiner for action on the merits commensurate with this decision.

/Christopher Grant/

Christopher Grant
Quality Assurance Specialist
Technology Center 2400

PTO/SB/80 (11-08)

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:

☒ Practitioners associated with the Customer Number:

20350

OR

☐ Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b). I further authorize any of the above-identified practitioners to execute a Statement Under 37 CFR 3.73(b) on the undersigned's behalf to certify the chain of title and establish the undersigned's ownership in any and all patent applications in which rights have been assigned to the undersigned.

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:

☒ The address associated with Customer Number:

20350

OR

☐ Firm or Individual Name			
Address			
City	State	Zip	
Country			
Telephone			Email

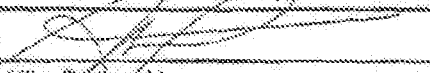
Assignee Name and Address:

Unicom Media, Inc.
24 West 5th Street
Tampa, AZ 85581

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/86 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	7-22-10
Name	Mike Gockewald	Telephone	480.214.8484
Title	Vice-President		

62790241 v1



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	028406-000710US

CONFIRMATION NO. 7494

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

FORMALITIES LETTER



0000000045945141

Date Mailed: 02/16/2011

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is missing.

A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.

Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- A surcharge (for late submission of filing fee, search fee, examination fee or oath or declaration) as set forth in 37 CFR 1.16(f) of **\$65** for a small entity in compliance with 37 CFR 1.27, must be submitted.

SUMMARY OF FEES DUE:

Total fee(s) required within **TWO MONTHS** from the date of this Notice is **\$65** for a small entity

- \$65 Surcharge.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/s/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Electronic Patent Application Fee Transmittal

Application Number:	13017866			
Filing Date:	31-Jan-2011			
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE			
First Named Inventor/Applicant Name:	Albert J. McGowan			
Filer:	Karam J. Saab/Corina Aschenbrenner			
Attorney Docket Number:	028406-000710US			
Filed as Small Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Late filing fee for oath or declaration	2051	1	65	65
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 2 months with \$0 paid	2252	1	245	245
Miscellaneous:				
Total in USD (\$)				310

Electronic Acknowledgement Receipt

EFS ID:	10264118
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab/Corina Aschenbrenner
Filer Authorized By:	Karam J. Saab
Attorney Docket Number:	028406-000710US
Receipt Date:	08-JUN-2011
Filing Date:	31-JAN-2011
Time Stamp:	19:30:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 310
RAM confirmation Number	7963
Deposit Account	201430
Authorized User	SAAB,KARAM

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		ResponsetoMissingParts.pdf	1890044 b746333797943cdd0cae3c0738559a14b85fc556	yes	9
	Multipart Description/PDF files in .zip description				
	Document Description		Start		End
	Extension of Time		1		1
	Transmittal Letter		2		3
	Oath or Declaration filed		4		5
	Assignee showing of ownership per 37 CFR 3.73(b).		6		6
	Power of Attorney		7		7
	Miscellaneous Incoming Letter		8		9
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	32060 e55fb5151dba2448f17127c1c9282984f38be8e5	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			1922104		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional)	
FY 2009		92853-798163	
<i>(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)</i>			
Application Number 13/017,866		Filed January 31, 2011	
For CLOUD DATA PERSISTENCE ENGINE			
Art Unit 2447		Examiner Not Yet Assigned	

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

	<u>Fee</u>	<u>Small Entity Fee</u>	
☐ One month (37 CFR 1.17(a)(1))	\$130	\$65	\$ _____
☒ Two months (37 CFR 1.17(a)(2))	\$490	\$245	\$ 245
☐ Three months (37 CFR 1.17(a)(3))	\$1110	\$555	\$ _____
☐ Four months (37 CFR 1.17(a)(4))	\$1730	\$865	\$ _____
☐ Five months (37 CFR 1.17(a)(5))	\$2350	\$1175	\$ _____

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☒ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☐ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 20-1430.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the ☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☒ attorney or agent of record. Registration Number 64,190

☐ attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

Signature

Date

Karam J. Saab, Reg. No. 64,190
Typed or printed name

303.607.3352
Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on June 8, 2011

PATENT
Attorney Docket No.: 92853-798163

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Corina K. Aschenbrenner/
Corina K. Aschenbrenner

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Albert J. McGowan

Application No.: 13/017,866

Filed: January 31, 2011

For: CLOUD DATA PERSISTENCE
ENGINE

Customer No.: 20350

Confirmation No.: 7494

Examiner: Not Yet Assigned

Art Unit: 2447

TRANSMITTAL LETTER --
RESPONSE TO NOTICE OF MISSING
PARTS

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

Pursuant to the Notice to File Missing Parts of Nonprovisional Application, dated
Missing Parts, enclosed are the following to be made of record in the above-identified
application:

- 1) Executed Declaration
- 2) Power of Attorney and Certificate of Assignee Under 37 C.F.R. § 3.73(b)
- 3) Petition to Extend Time
- 4) Copy of Notice of Missing Parts

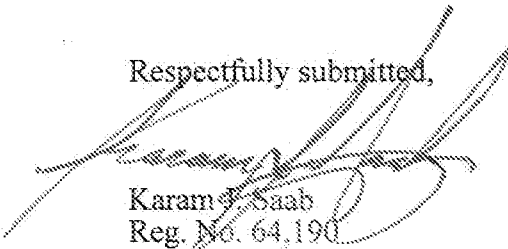
Please charge credit card provided for the following fees:

Small Entity:	(a) Filing Fee	\$0
	(b) Search Fee	\$0

(c)	Examination Fee	\$0
(d)	Application Size Fee	\$0
(e)	Excess Claims Fees (§ 1.16(b), (c)):	
	20 - 20 = 0 x \$52 =	\$0
	3 - 3 = 0 x \$220 =	\$0
(f)	Multiple Dependent Claims	
(g)	Missing Parts Surcharge	\$65
	TOTAL FEES TO BE CHARGED	\$65

The Commissioner is hereby authorized to charge any additional fees associated with this paper or during the pendency of this application, or credit any overpayment, to Deposit Account No. 20-1430.

Respectfully submitted,


Karam J. Saab
Reg. No. 64,190

Customer No. 20350

KILPATRICK TOWNSEND & STOCKTON LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000
Fax: 415 576-0300
K4S:cka

63473570 v1

DECLARATION

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: CLOUD DATA PERSISTENCE ENGINE the specification of which _____ is attached hereto or X was filed on January 31, 2011 as Application No. 13/017,366 and was amended on _____ (if applicable).

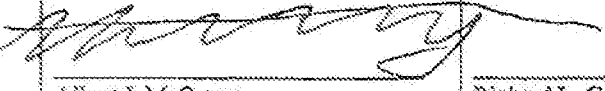
I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119
Australia	2010202782	07/01/2010	

Full Name of Inventor 1:	Last Name: MCGOWAN	Suffix:	First Name: ALBERT	Middle Name or Initial: J.
Residence & Citizenship:	City: Phoenix		State/Foreign Country: Arizona	Country of Citizenship: United States
Post Office Address:	Post Office Address: 4201 East Camelback Road #101		City: Phoenix	State/Country: Arizona
				Postal Code: 85018
Full Name of Inventor 2:	Last Name: CARLS	Suffix:	First Name: RICHARD	Middle Name or Initial: L.
Residence & Citizenship:	City: Scottsdale		State/Foreign Country: Arizona	Country of Citizenship: United States
Post Office Address:	Post Office Address: 19793 West 84th Way		City: Scottsdale	State/Country: Arizona
				Postal Code: 85255

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1 	Signature of Inventor 2 
Albert J. McGowan Date 7-10-11	Richard L. Carls Date

63181072 v1

DECLARATION

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: CLOUD DATA PERSISTENCE ENGINE the specification of which _____ is attached hereto or X was filed on January 31, 2011 as Application No. 13/017,866 and was amended on _____ (if applicable).

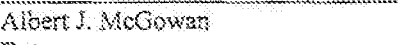
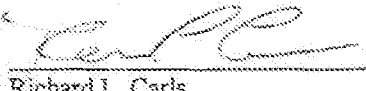
I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119.
Australia	2010202782	07/01/2010	

Full Name of Inventor 1:	Last Name: MCGOWAN	Suffix:	First Name: ALBERT	Middle Name or Initial: J.
Residence & Citizenship:	City: Phoenix	State/Foreign Country: Arizona		Country of Citizenship: United States
Post Office Address:	Post Office Address: 4201 East Camelback Road #101	City: Phoenix	State/Country: Arizona	Postal Code: 85018
Full Name of Inventor 2:	Last Name: CARLS	Suffix:	First Name: RICHARD	Middle Name or Initial: L.
Residence & Citizenship:	City: Scottsdale	State/Foreign Country: Arizona		Country of Citizenship: United States
Post Office Address:	Post Office Address: 19793 N. 84th Way	City: Scottsdale	State/Country: Arizona	Postal Code: 85255

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1	Signature of Inventor 2
	
Albert J. McGowan	Richard L. Carls
Date	Date <u>4/16/2011</u>

STATEMENT UNDER 37 CFR 3.73(b)Applicant/Patent Owner: Albert J. McGowanApplication No./Patent No.: 13/017,866Filed/Issue Date: January 31, 2011Titled: CLOUD DATA PERSISTENCE ENGINEUnicom Media, Inc., a corporation

(Name of Assignee)

(Type of Assignee: corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest in;
2. ☐ an assignee of less than the entire right, title, and interest in
(The extent (by percentage) of its ownership interest is _____ %); or
3. ☐ the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

- A. ☒ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 026230, Frame 0734, or for which a copy thereof is attached.

OR

- B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

2. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet (s).

- ☒ As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Signature

June 8, 2011

Date

Karam J. Saab, Reg. No. 64,190

Printed or Typed Name

Attorney of Record

Title

Substitute for form 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use as many sheets as necessary)				Complete if Known	
				Application Number	13/017,866
				Filing Date	January 31, 2011
				First Named Inventor	Albert J. McGowan
				Art Unit	2447
				Examiner Name	Not Yet Assigned
Sheet	1	of	1	Attorney Docket Number	92853-798163 (000710US)

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number Kind Code ² (if known)			
	A1	US 2003/0061272 A1	03-27-2003	Krishnamurthy, et al.	
	A2	US 7,373,394 B1	05-13-2008	Li, et al.	

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No. ¹	Foreign Patent Document			Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴	Kind Code ⁵ (if known)				

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials *	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ⁶
	C1	CHANG, F., et al., "Bigtable: A Distributed Storage System for Structured Data," [online], Google Inc., November 2006, [Retrieved from the Internet on July 28, 2011], Retrieved from: http://labs.google.com/papers/bigtable.html , 14 pages.	☐

Examiner Signature		Date Considered	
-----------------------	--	--------------------	--

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kind Codes of U.S. Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

63637149 v1

Electronic Acknowledgement Receipt

EFS ID:	10734700
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab/David Voegtle
Filer Authorized By:	Karam J. Saab
Attorney Docket Number:	92853-798163 (000710US)
Receipt Date:	15-AUG-2011
Filing Date:	31-JAN-2011
Time Stamp:	17:56:58
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		2011_08_15_92853_798163_IDS.pdf	1606283 5d6a50568ee3d9c7f6c0352d96795348bb564973	yes	3

	Multipart Description/PDF files in .zip description								
	Document Description		Start	End					
	Transmittal Letter		1	2					
	Information Disclosure Statement (IDS) Form (SB08)		3	3					
Warnings:									
Information:									
2	Non Patent Literature	Google_Big_Table_07_28_2011.pdf	221214 e2e60f19214efaa7d842b4a3b3b322964bf78dc7	no	14				
Warnings:									
Information:									
Total Files Size (in bytes):			1827497						
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>									

KILPATRICK TOWNSEND & STOCKTON LLP

By: /David R. Voegtle/
David R. Voegtle

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Albert J. McGowan

Application No.: 13/017,866

Filed: January 31, 2011

For: CLOUD DATA PERSISTENCE
ENGINE

Customer No.: 20350

Confirmation No.: 7494

Examiner: Not Yet Assigned

Art Unit: 2447

INFORMATION DISCLOSURE
STATEMENT UNDER 37 CFR §1.97 and
§1.98

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

The references cited on attached form PTO/SB/08A&B are being called to the attention of the Examiner. Copies of the references are enclosed in compliance with the requirements of 37 CFR §1.98(a)(2).

It is respectfully requested that the cited references be expressly considered during the prosecution of this application, and the references be made of record therein and appear among the "references cited" on any patent to issue therefrom.

As provided for by 37 CFR §1.97(g) and (h), no inference should be made that the information and references cited are prior art merely because they are in this statement and no

representation is being made that a search has been conducted or that this statement encompasses all the possible relevant information.

Applicant believes that no fee is required for submission of this statement.

However, if a fee is required, the Commissioner is authorized to deduct such fee from the undersigned's Deposit Account No. 20-1430. Please deduct any additional fees from the above-noted Deposit Account.

Respectfully submitted,

/Karam J. Saab/

Karam J. Saab
Reg. No. 64,190

KILPATRICK TOWNSEND & STOCKTON LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000
Fax: 303-571-4321
K4S:d1v
63637149 v1



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
13/017,866	01/31/2011	2447	527	92853-798163	20	3

CONFIRMATION NO. 7494

UPDATED FILING RECEIPT



OC000000050602718

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Date Mailed: 10/28/2011

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Albert J. McGowan, Phoenix, AZ;
Richard L. Carls, Scottsdale, AZ;

Assignment For Published Patent Application

Unicorn Media, Inc., Tempe, AZ

Power of Attorney: The patent practitioners associated with Customer Number 20350

Domestic Priority data as claimed by applicant

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)

AUSTRALIA 2010202782 07/01/2010

Request to Retrieve - This application either claims priority to one or more applications filed in an intellectual property Office that participates in the Priority Document Exchange (PDX) program or contains a proper **Request to Retrieve Electronic Priority Application(s)** (PTO/SB/38 or its equivalent). Consequently, the USPTO will attempt to electronically retrieve these priority documents.

If Required, Foreign Filing License Granted: 02/10/2011

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/017,866**

Projected Publication Date: 02/02/2012

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

CLOUD DATA PERSISTENCE ENGINE

Preliminary Class

709

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER**Title 35, United States Code, Section 184****Title 37, Code of Federal Regulations, 5.11 & 5.15****GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as

set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 13/017,866				
APPLICATION AS FILED - PART I										
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY				
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)			
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	95	N/A				
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	310	N/A				
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	125	N/A				
TOTAL CLAIMS (37 CFR 1.16(i))	20	minus 20 =	*	x 30 =	0.00	OR				
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3	minus 3 =	*	x 125 =	0.00					
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00					
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00					
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	530	TOTAL				
APPLICATION AS AMENDED - PART II										
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY		
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))							OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY		
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))							OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.										



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	92853-798163

CONFIRMATION NO. 7494

POA ACCEPTANCE LETTER



OC000000050602689

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Date Mailed: 10/28/2011

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/08/2011.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/twoldeyes/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



Australian Government

Patent Office
Canberra

I, JAMES MCGREGOR, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Complete specification in connection with Application No. 2010202782 for a patent by UNICORN MEDIA, INC. as filed on 01 July 2010.



WITNESS my hand this
Eighteenth day of November 2011

A handwritten signature in black ink, appearing to read 'James McGregor'.

JAMES MCGREGOR
TEAM LEADER EXAMINATION
SUPPORT AND SALES

2010202782 01 Jul 2010

Regulation 3.2

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION
STANDARD PATENT

APPLICANT: **UNICORN MEDIA, INC**

Invention Title: **CLOUD DATA PERSISTENCE ENGINE**

The following statement is a full description of this invention, including the best method of performing it known to me:

CLOUD DATA PERSISTENCE ENGINE

BACKGROUND OF THE INVENTION

5 [0001] This disclosure relates in general to cloud-based computer processing and, but not by way of limitation, to distribution, storage, and retrieval of data on a cloud networking arrangement.

[0002] In a typical data hosting architecture, multiple servers may be spread over a geographical area. The data stored on each of these servers may be the same (or substantially similar) data as that stored on each other server. A client attempting to access data hosted on
10 such a system may communicate with a particular server, often the server closest to the client's physical location.

[0003] While such an arrangement may allow the client to quickly access all data stored on the server system, it may not be efficient for all data to be stored on each server. Clients in different locations (e.g., different cities, states, countries, and/or continents) may request data that varies
15 significantly by location. For example, particular data may frequently be requested by clients in Europe, while the same data may be infrequently requested by clients in the Americas. Therefore, if all of the data is maintained at each server, such a system may result in significant quantities of data being stored at servers where the data is infrequently, if at all, accessed. Accordingly, such a system may result in a substantial amount of data being stored at servers that
20 wastes storage resources and is infrequently accessed.

BRIEF SUMMARY OF THE INVENTION

[0004] Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a
25 link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data object stored locally, it may contact a second server to determine if it has the data object stored. If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then

send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

[0005] In some embodiments, a cloud data persistence system for distribution of data is present. The system may include a first server which stores a first plurality of data objects and is communicatively coupled with a plurality of cache servers. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to one of a plurality of media assets. The plurality of media assets may be stored at a second location separate from the first location. The system may include a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects. The second server may be one of the plurality of cache servers. The second server may be communicatively coupled with a first plurality of application servers. The system may include a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects. The third server may be one of the plurality of cache servers. The third server may be communicatively coupled with a second plurality of application servers. The system may include a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fourth plurality of data objects. The fourth server may be one of the first plurality of application server. The fourth server may be configured to receive a request from a first client for the data object. The system may also include a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects. The fifth server may be one of the first plurality of application servers. The fifth server may be configured to receive a request from a second client for the data object. The system may further include a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects. The sixth server may be one of the second plurality of application servers. The sixth server may be configured to receive a request from a third client for the data object. The system may include a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects. The seventh server may be one of the second plurality of application servers. The seventh server may be configured to receive a request from a fourth client for the data object.

[0006] In some embodiments, a method for retrieving data objects using a cloud data persistence system configured for distribution of data is presented. The method may include receiving a first request from a client for a data object. The application center may be one of a plurality of application centers that communicate with a plurality of clients. The application center may be located at a first location. The data object may comprise a link to a media asset. The media asset may be stored at a second location separate from the first location. The method may further include determining the data object is not stored among a first plurality of data objects stored at the application center. The method may include transmitting a second request for the data object to a first cache server. The first cache server may be communicatively coupled with the plurality of application centers. The first cache server may be one of a plurality of cache servers. The method may include receiving the second request for the data object. The method may also include determining the data object is not stored among a second plurality of data objects stored at the cache server. The method may further include transmitting a third request for the data object to an origin server. The origin server may be communicatively coupled with the plurality of cache servers. The method may also include receiving the third request for the data object. The method may include locating the data object among a third plurality of data objects. The method may also include transmitting the data object linked to the third request to the cache server. The method may include receiving the data object linked to the third request. The method may include transmitting the data object linked to the second request. Also, the method may include receiving the data object linked to the second request. Further, the method may include transmitting the data object linked to the first request to the client.

[0007] In some embodiments, a cloud data persistence system for distribution of data is present. The system includes a first server which stores a first plurality of data objects and is communicatively coupled with a second server and a third server. The first server may be located at a first location. Each of the first plurality of data objects may comprise a link to a media asset of a plurality of media assets. The plurality of media assets may be located at a second location. The second location may be distinct from the first location. The system may include the second server, which may store a second plurality of data objects and may be configured to request a data object from the first server if the data object is not present in the second plurality of data objects. The second server may be communicatively coupled with a fourth server and fifth server. The system may include the fourth server that stores a third plurality of data objects and may be configured to request the data object from the second server

if the data object is not present in the third plurality of data objects. The fourth server may be configured to receive a request from a first client for the data object.

[0008] In some embodiments, the first plurality of data objects comprises all data objects present in the second plurality of data objects and the third plurality of data objects. In some
5 embodiments, the third server stores a fourth plurality of data objects and is configured to request the data object from the first server if the data object is not present in the fourth plurality of data objects. The third server and the second server may be data persistence cache servers. The third server may be communicatively coupled with a sixth and seventh server. In some embodiments, the fifth server stores a fifth plurality of data objects and is configured to request the data object
10 from the second server if the data object is not present in the fifth plurality of data objects. The fifth server and the fourth servers may be application servers. The fifth server may be configured to receive requests from a second client for the data object. The fifth server may be geographically separated from the fourth server. In some embodiments, the request from the first client for the data object comprises a universal resource indicator. In some embodiments, each
15 of the first plurality of data objects comprise a universal resource locator and a title linked to a media asset of the plurality of media assets. Also, in some embodiments, data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on a frequency of requests for those data objects received from clients. In some embodiments, data objects in the second plurality of data objects and the third plurality of
20 data objects are at least partially determined based on when requests for those data objects were last received from clients. Further, in some embodiments, the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

25

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present disclosure is described in conjunction with the appended figures:

[0010] FIG. 1 illustrates a block diagram of a media servicing system.

[0011] FIG. 2A illustrates a block diagram of an embodiment of a kernel application center connected with application centers.

- [0012] FIG. 2B illustrates a block diagram of an alternative embodiment of a kernel application center.
- [0013] FIG. 3 illustrates a block diagram of an embodiment of an application center.
- [0014] FIG. 4A illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion.
- [0015] FIG. 4B illustrates a block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery network.
- [0016] FIG. 4C illustrates another block diagram of processes and objects utilized by a cloud-hosted integrated multi-node pipelining system for media ingestion with a content delivery network.
- [0017] FIG. 5A illustrates a simplified block diagram of an embodiment of a cloud data persistence system.
- [0018] FIG. 5B illustrates another simplified block diagram of an embodiment of a cloud data persistence system.
- [0019] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to retrieve a data object from a cloud data persistence system.
- [0020] FIG. 7 illustrates a simplified swim diagram of an embodiment of a method for responding to a client request to modify a data object on a cloud data persistence system.
- [0021] FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management and storage system.
- [0022] FIG. 9 illustrates another simplified block diagram of an embodiment of an asset management and storage system.
- [0023] FIG. 10 illustrates an embodiment of a method for retrieving data from an asset management and storage system.
- [0024] In the appended figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is

applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION OF THE INVENTION

5 [0025] The ensuing description provides preferred exemplary embodiment(s) only, and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing description of the preferred exemplary embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred exemplary embodiment. It is understood that various changes may be made in the function and arrangement of elements without departing
10 from the spirit and scope as set forth in the appended claims.

[0026] As opposed to storing copies of a piece of data, such as a data object that contains information linked to a media object, at each server on a system of multiple servers, it may be more efficient to store only data objects that are frequently and/or recently accessed by clients at particular servers. Such an arrangement, generally referred to as a cloud data persistence engine
15 or system, may allow the data objects stored at each server to vary based on which data objects the server's clients are currently and/or have historically requested. Therefore, data objects stored at a first server in a first location may include some of the same data objects and some different data objects than data objects stored at a second server in a second location (possibly located in a different city, country or on a different continent than the first server). Storing only
20 data objects that are frequently and/or recently accessed by clients at each server may result in considerable efficiencies. For example, by storing only data objects that are frequently and/or recently accessed by clients, the servers may require significantly less storage space because they do not need to store a potentially large amount of data objects that are infrequently accessed or have not recently been accessed. A related benefit of this may be that by decreasing the amount
25 of storage necessary at each server, the speed of access to the storage at the servers may be increased. For example, instead of relying on hard drives, solid state drives (which may cost significantly more per megabyte, but may offer improved access times) may be used for the storage at the servers.

[0027] While a server may infrequently receive a request for a particular data object from a
30 client, the server may still need to be able to fulfill such a request. Therefore, if a server is contacted by a client for a data object that it does not have stored locally, the server may be able

to contact another server, which may have the requested data object stored. Such a server may be configured to satisfy multiple servers with requests for such data objects.

5 [0028] While this second cache server may contain the data object requested by the client, it may also only store data objects that are frequently and/or recently (albeit less frequently and/or recently than the data stored at the first server) accessed by clients. If the second server does not have the data object requested by the client stored, the second server may have the ability to contact yet another server. This server, referred to as the data object origin server, may maintain a copy of all data objects stored on the system. Therefore, if the data object is present on the server system, the data object origin server is able to fulfill the request for the data. The data
10 object-origin server may then route the requested data to the client via the second server and the first server.

[0029] While the above arrangement may increase efficiencies when a client requests a data object, a client (perhaps the same or a different client) may request to modify (including, but not limited to changing, adding and/or deleting) a data object. In such instances, the modification to
15 the data object (assuming the client has the appropriate access rights) may be routed to the data object origin server by a server receiving the request from the client. The modification may be applied to the relevant data at the data object origin server. The data object origin server may then send a notification to other servers, or only the cache servers that have the previous version of the data stored, that the previous version of the data is no longer valid. These servers may in
20 turn send a notification to other associated servers, or only those servers that have the previous version of the data stored, that the previous version of the data object is no longer valid. This may prevent a client from receiving an out-of-date version of the data no matter which server is accessed by the client. The modified data residing at the origin may then propagate out to the other servers as requests are made by clients to those servers for the data object.

25 [0030] While the above embodiments may be implemented in a variety of different systems, some particular embodiments may be implemented as part of a media service system. FIG. 1 is a block diagram illustrating a media servicing system 100, according to some embodiments of the present invention. The system may deliver media content to the end user 140 (alternatively referred to as a user or a client) through a network such as the Internet 120. A media asset
30 provided by a content provider 130 can be processed and indexed by cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, and further stored on content delivery network

(CDN) 150. Additionally or alternatively, the CHIMPS 110 may also be adapted to store the media asset.

[0031] The media servicing system further enables a content provider 130 or other entity to gather information regarding user behavior during media playback. For example, a content provider 130 can be provided with data indicating that end users tend to stop watching a video at a certain point in playback, or that users tended to follow links associated with certain advertisements displayed during playback. With this data, a content provider 130 can adjust factors such as media content, advertisement placement and content, etc., to increase revenue associated with the media content and provide the end user 140 with a more desirable playback experience.

[0032] End user 140 can request a media asset to hear and/or see with a client program running on a personal computer or other electronic device connected to Internet 120. The client program can be, for example, a media player, browser, or other application adapted to request and/or play media assets. In response to a request for a media asset, the CHIMPS 110 can utilize any number of application centers 112 and/or kernel application center(s) 111 to provide the client program with a data object concerning the requested media asset. The data object can include information about the media asset, including where the media asset can be located, such as within the CDN 150 or within the CHIMPS 150 itself. Location information may be provided by Universal Resource Indicator (URI), a Universal Resource Locator (URL) or other indicator.

During playback of the media asset, the CHIMPS 150 can collect data regarding the playback through beaconing provided by a client program executed on a device operated by the end user 140 and/or indexing service from within the CHIMPS and/or CDN. The CHIMPS 150 can subsequently provide the data and/or any analytics information derived from the data to the content provider 130.

[0033] FIG. 2A is a block diagram illustrating an embodiment of a kernel application 111-1 center connected with application centers from within the CHIMPS 110-1. The kernel application center 111-1 and application centers 112 can be geographically distant and can be connected via the Internet 120, wide area network (WAN), and/or other data communication network. Because application centers can be geographically separated, DNS services (not shown) can be used to allow an end user 140 to connect to the nearest available application center 112. The kernel application center 111-1 can connect with application centers 112 within

the CHIMPS 110-1 through an internal interface 270, thereby enabling the application centers 112 access to the various components within the kernel application center 111-1.

[0034] Components within the kernel application center 111-1 can communicate through network 260 such as a local area network (LAN) and can include one or more data object origin servers 240 and a storage array 230 with which data objects and/or media assets may be stored and distributed. The storage array 230 may also be utilized by services running on processing server(s) 220 and/or transcoding server(s) 250 that may require temporary or long-term storage. Kernel server 210 can utilize processing server(s) 220, transcoding server(s) 250 to provide various functional capabilities to the CHIMPS 110.

10. [0035] For example, as described in more detail below, the CHIMPS 110-1 can provide transcoding service for media assets provided by a content provider 130 for syndication. Such a service can allow a content provider 130 to upload a media asset to an application center 112, after which the application center 112 would notify the kernel server 210 that the media asset has been uploaded. The kernel server can then notify services running on the processing server(s) 220 of the upload. These services can utilize transcoding server(s) to transcode the media asset, which can then be moved to a CDN and/or stored locally by storage array 230 and data object origin server(s) 240. Services running on the processing server(s) 220 can also update the associated data object stored by the storage array 230 and data object origin server(s) 240.

20. [0036] FIG. 2B is a block diagram illustrating an alternative embodiment of a kernel application center 111-2. In addition to the components of the embodiment of FIG. 2A, this embodiment incorporates an application center 112 within the kernel application center 111-2. The application center 112 incorporated within kernel application center 111-2 may be located at or near the other components of the kernel application center 111-2, and can be communicatively connected to the other components via network 260. The incorporated application center 112 can therefore have faster access to kernel application center functionality because it does not need to communicate over long distances. In consideration of this advantage, it will be understood that the CHIMPS 110 can include multiple kernel centers with one or more application centers incorporated therein. Additionally or alternatively, components of the kernel application center may be incorporated into one or more application centers 112 in the CHIMPS 110 to provide quicker access to certain functionality.

[0037] FIG. 3 is a block diagram illustrating an embodiment of an application center 112. The application center 112 can include caching server(s) 330 and a storage array 310 for storing and distributing data objects of media assets requested by end users through end user interface 360. Caching server(s) 330 and storage array 310 can also be used to collect, process, and/or store metrics information from beaconing data, media chunk requests, and/or other data sources, including data collected through end user interface 360. The application center can further include ingest server(s) 320 for ingesting uploaded media assets from a content provider 130 through a content provider interface 370. The media assets may be stored on the storage array 310. As with the kernel application center 111, the components of the application center 112 can be communicatively linked through a network 340, such as a LAN. The application center can further include an internal interface 350, providing a communication link from the application center to the rest of the CHIMPS. It is through internal interface 350, for example, that media assets stored on storage array 310 can be made available to a kernel application center 111 for services such as transcoding.

[0038] FIG. 4A is a block diagram 400-1 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Although FIG. 4A further indicates the physical systems in which my execute or store these processes and objects, it will be understood that the processes and objects disclosed may be executed or stored on more than one system, including systems not disclosed in FIG. 4A. In other words, the processes and objects shown in FIG. 4 allow for a variety of implementations through one or more of hardware, software, firmware, microcode, etc.

[0039] Media can be ingested into the CHIMPS 110 when a content provider 130 uploads a media asset to ingestion server(s) 410 in an application center 112 by utilizing a client 405. The client 405 can be a stand-alone application or browser based, for example, and can communicate with ingest server(s) 410 through an application programming interface (API) configured for the ingestion of media assets.

[0040] Ingest server(s) 410 can communicate with devices in the kernel application center 111 executing programs such as kernel server 425 and file replication service 430. The kernel server 425 can be configured to organize the workflow among services such as transcoding 440 file system manager 435, and other services 445 (e.g., analytics, dynamic API, etc.) Upon a

particular event, for example, the kernel server can be configured to notify the relevant services of the event, causing the services to process tasks associated with the event.

[0041] The file replication service 430, under direction of the kernel server 425, can coordinate the movement of the media assets between services. For example, retrieving the uploaded media asset from the ingest server(s) 410 and storing it on the file archive 450, or retrieving transcoded media assets from transcoding server(s) 460 and storing them in the media asset origin.

[0042] The data object updater 420 keeps the data object origin 415 up to date in response to any changes in the system. When, for example, a file is uploaded, transcoded, and stored in media asset origin 455, the location and other metadata concerning the transcoded media assets need to be created or updated in the data object origin 415 to ensure an end user who accesses the object in the data object origin 415 has the correct information regarding the related media asset. Because the data object updater 420 receives updates from the kernel server 425 (which is notified when a transcoded media asset is stored in the media asset origin 455, the system ensures the data objects in the data object origin are constantly up to date.

[0043] The upload of a media asset to the ingest server(s) 410, as described above, can provide an example of how the kernel server 425 may coordinate workflow. For instance, in response to the upload, the ingest server(s) 410 can notify the kernel server 425 that a media asset has been uploaded. The kernel server 425 informs the file replication service 430 of the uploaded media asset, and the file replication service 430 moves the uploaded media asset into the file archive 450 and notifies the kernel server 425 of the move. In response, the kernel server 425 notifies the file replication service 430, the file system manager 435, and the transcoding master 440 of the move. The file replication service 430 then will know it can delete the uploaded media asset from the ingest server(s) 410, the file system manager 435 will update the file system accordingly, and the transcoding master 440 will notify transcoding service(s) 460 of different transcoding tasks to be performed. The transcoding service(s) 460 can then retrieve the uploaded media asset from the file archive 450 to create transcoded media assets. The transcoding service(s) 460 notify the kernel server 425 once transcoding is complete, and the kernel server 425 relays this information to the file replication service 430. The file replication service 425 then takes the transcoded media assets from the transcoding services 460 and moves them to the media asset origin 455. Once the file replication service 430 notifies the kernel server 425 of the move, the kernel server 425, in turn, notifies the file replication service 430 and the data object

updater 420. The data object updater 420, which updates the data object origin 415 accordingly, and the file replication service 430 deletes the transcoded media assets from the transcoding services 460.

[0044] The modular nature of the system enables all tasks associated with an event to be completed quickly. As illustrated in the example above, workflow relating to a particular event, such as a media asset upload, can be spread among the various services simultaneously. Moreover, because the system's modularity enables it to be scaled to accommodate differing hardware capacities, and because the system can be configured to dynamically allocate hardware to different services according to the needs of the system, the speed of completing tasks relating to a particular event can further be increased.

[0045] FIG. 4B is a block diagram 400-2 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-2 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in place of media asset origin 455. Therefore, in block diagram 400-2, as opposed to the media assets being stored at media asset origin 455, they are stored at content data network 150, which may be operated and maintained by a third party.

[0046] FIG. 4C is a block diagram 400-3 of processes and objects utilized by the CHIMPS 110 for media ingestion, according to some embodiments. Block diagram 400-3 may be substantially similar to block diagram 400-1 except for the use of a content delivery network in conjunction with media asset origin 455. Here, in block diagram 400-3, media assets may reside at either media asset origin 455, content delivery network 150, or both.

[0047] Embodiments of such systems may include other systems that manage various requests from end users. One such system may be a cloud data persistence system. Referring to FIG. 5A, a simplified block diagram of an embodiment of a cloud data persistence system 500-1 is illustrated. Such a system may allow for information, such as data objects, to be efficiently distributed to multiple of clients.

[0048] Some embodiments of cloud data persistence systems are used to distribute data objects that are linked to media assets. A data object may contain information pertinent to the data object it is linked to. For example, a data object may include details about a media asset and permissions for the media asset. More specifically, a data object may include fields containing: a title, keywords linked to the data object, the data object's (and/or associated media asset's) date

of creation, the duration of the associated media asset, the file formats the associated media asset is available in, and what clients are permitted to access and/or edit the data object and/or the media asset. The data object may also contain a link to its associated media asset. Such a link may be in the form of a universal resource indicator, such as a universal resource locator. Other
5 information may also be stored as a data object that is linked to a media asset. A media asset may represent many different forms of media, such as an audio file (in a variety of different formats), a video file, or a flash file, to name only a few examples.

[0049] Important to note, data objects may be stored separate from media assets. For example, a data object containing information linked to a particular media asset may not be stored in the
10 same location as the media asset. For example, a media asset may reside on multiple servers that are part of a content delivery network, while the data object that contains information about the media asset is stored in some other location, possibly on some other network. Also, while the above discusses linked media assets, it should be understood that data objects may also be linked to other assets, files or data that do not constitute media. In such embodiments, the data objects
15 may contain different fields of data.

[0050] Such a cloud data persistence system may be used to distribute data objects to one or several clients, but also may be used to distribute data objects to thousands of clients geographically disbursed worldwide. In the embodiment of cloud data persistence system 500-1, three levels of servers are present: a data object origin server 510, data persistence cache servers
20 520, and application centers 530. These application centers are in contact with one or more clients 540. It should be understood that each application center may include one or more servers. For example, an application center may be represented by application center 112 of FIG. 3. Alternatively, an application center may comprise a single server. While three levels of servers are present in the illustrated embodiment, it should be understood that greater or fewer
25 levels of servers may be present in other embodiments. Also, various servers may be integrated into the same server. For example, one server may serve as both an application center 530 and a data persistence cache server 520. It should also be understood that the terms of "application center," "data persistence cache server," and "data object origin server" are used to simplify the nomenclature used to describe the various levels of the cloud data persistence system 500-1, and
30 these terms should not be interpreted as limiting the functionality of their respective identified servers.

- 5 [0051] Referring to cloud data persistence system 500-1, clients 540 make requests for data objects from application centers 530 via one or more networks. In the illustrated embodiment, only two clients (540-1 and 540-2) are present to maintain simplicity; many more clients (possibly representing many end users) may be possible. These clients may communicate with the same application center (530-1) or may communicate with different application centers (530-1 and 530-2). Further, because a client at one time interacts with a particular application center, at a different time, the same client may interact with a different application center. For example, if a particular application center is experiencing a heavy client load, possibly caused by a large number of clients attempting to access it, a client may be routed to another application center.
- 10 [0052] The client may make a request for a particular data object to the application center. This request may be made via a network, such as the Internet. Other networks may also be possible, such as a private intranet. The request for the data object may involve the client sending a universal resource indicator (URI) to an application center. A URI may take a variety of different formats. In particular, the URI may be in the form of a universal resource locator (URL). However, it should be understood that other forms of URIs may also be possible.
- 15 [0053] If client 540-1 transmits a URI to application center 530-1, application center 530-1 may have the data object linked to the URI stored locally. If this is the case, application center 530-1 may respond by sending the data object identified by the URI back to client 540-1. However, application center 530-1 may have only a subset of the data objects linked to valid URIs stored locally. This may be due to space constraints at application center 530-1. For example, if clients infrequently request particular data objects, it may be inefficient to distribute those data objects to each of application centers 530.
- 20 [0054] If the data object requested by client 540-1 is not present at application center 530-1, application center 530-1 may contact another server to retrieve it. Application center 530-1 may be located in close proximity to data persistence cache server 520-1, or may be geographically separated from data persistence cache server 520-1. Application center 530-1 may communicate with data persistence cache server 520-1 directly (such as a direct cable connection) or may communicate using a network. This network may be a private network, such as a local area network or dedicated connection between the servers, or may use a public network, such as the Internet. Such a request from application center 530-1 to data persistence cache server 520-1 may include the URI received by application center 530-1 from client 540-1. In some
- 30

embodiments, data persistence cache servers supply multiple application centers with requested data objects. In cloud data persistence system 500-1, data persistence cache server 520-1 serves two application centers (530-1 and 530-2), while data persistence cache server 520-2 serves two different application centers (530-3 and 530-4). It should be understood that a data persistence cache may serve a varying number of application centers. For example, a data persistence cache may serve 20 application centers.

[0055] It should also be recognized that the connections between application centers 530 and data persistence cache servers 520 are dynamic. If a particular data persistence cache is offline, busy, or otherwise unavailable, an application center may be able to route requests to a different data persistence cache server. For example, referring to cloud data persistence system 500-1, if data persistence cache 520 became unavailable, application center 530-1 may be able to route a request for the data object to data persistence cache server 520-2.

[0056] While, if application centers 530 and data persistence caches 520 communicate via a network (e.g., the Internet), the servers may be located anywhere a network connection is available worldwide. In some embodiments, a data persistence cache is located near a group of application centers. For example, if a region, such as New England, contains two application centers for each state, a single data persistence cache may be located in or near New England to serve these twelve scattered application centers. Having the data persistence cache located near a group of application centers may decrease communication latency when an application center and the data persistence cache communicate.

[0057] Upon receiving a request for a data object from application center 530-1, data persistence cache server 520-1 may determine if it has the data object stored. If it does, data persistence cache server 520-1 may send the data object to application center 530-1, which may, in turn, send the data object to client 540-1. Upon the application center receiving the data object, it may store the data object to be able to satisfy requests from the same or a different client for the same data object at a future time. Application center 530-1 may make a determination as to whether the data object should be stored or not. For example, the determination may be based on the frequency it is receiving requests for the data object from clients 540. If the data object is frequently being requested by clients 540, application center 530-1 may store the data object. This may only occur if the data object is being requested more frequently than some other data objects. For example, only the 10,000 most frequently requested

data objects may be stored by the application center. In some embodiments, application center 530-1 may store the most recently requested data objects. Therefore, this data object would be stored at application center 530-1, and possibly some other data object would no longer be stored at application center 530-1. In some embodiments, different application centers use different
5 mechanisms to determine what data objects to store. Also, in some embodiments, some data objects are always stored at application centers 530. This may occur if a particular data object has been labeled a "high priority" (or some equivalent thereof).

[0058] However, if data persistence cache server 520-1 determines that it does not have the data object requested by client 540-1 stored, data persistence cache server 520-1 may contact
10 another server. Data persistence cache server 520-1 may contact another server called a data object origin server 510. Data object origin server 510 may supply data persistence cache servers 520 with various data objects. In cloud data persistence system 500-1, data object origin server 510 is shown as communicating with two data persistence cache servers 520. However, it should be understood that data object origin server 510 may communicate with one, three, or
15 some other number of data persistence cache servers. For example, data object origin server 510 may communicate with more, such as 20 data persistence cache servers.

[0059] Data object origin server 510 may be physically located with one or more data persistence cache servers. Data object origin server 510 may communicate directly, such as via a direct connection, with data persistence cache servers. For example, in cloud data persistence
20 system 500-1, data object origin server 510 communicates directly with the data persistence cache server 520-2. Data object origin server 510 may also communicate via any of the previously mentioned networks with a data persistence cache server. Again, referring to cloud data persistence system 500-1, data object origin server 510 communicates with data persistence cache server 520-1 via a network, possibly the Internet.

25 [0060] While application centers 530 and data persistence cache servers 520 may store only a subset of the various data objects that clients 540 may request, data object origin server 510 may store all data objects that exist on system 500-1. In other words, if a data object requested by a client is not present on data object origin server 510, the data object may not exist on system 500-1.

30 [0061] Just as a data persistence cache server may be physically located near a group of application centers, data object origin server 510 may be located near the group of data

persistence cache servers 520. Returning to the previous regional example, if a data persistence cache server is located in New England, another may be located near New York City, another near Washington DC, and perhaps yet another near Chicago. If one data object origin server is servicing these data persistence cache servers, it may be located in a central location to serve them all. In some embodiments, the data object origin server may be co-located with one or more data persistence cache servers. Alternatively, a data object origin server may be located at a separate physical location from any other database servers.

[0062] A request from the data persistence cache server 520-1 to data object origin server 510 may comprise a URI. This may be the same URI that was sent from client 540-1 to application center 530-1, then from application center 530-1 to data persistence cache server 520-1. In some embodiments, the URI transmitted from data persistence cache 520-1 may be in a different format than the other URIs. Upon receiving the URI, data object origin server 510 may locate the data object the URI is linked to. The data object origin 510 may then transmit the data object to data persistence cache server 520-1. The data persistence cache server 520-1 may make determination as to whether to store the data objects as previously described in relation to application center 530-1. For example, the determination of whether to store the data object may be based on the frequency with which application centers 530 request the data object from data persistence cache server 520-1. In some embodiments, data persistence cache server 520-1 stores the most recently requested data objects. Data persistence cache server 520-1 may then transmit the data object to application center 530-1. Application center 530-1 may then transmit the data object to client 540-1.

[0063] If the data object was first found at application center 530-1, the latency between client 540-1's request and client 540-1 receiving the data object may be the shortest. If instead the data object is first found at data persistence cache server 520-1, the latency may be longer. Finally, if the data object is only found at data object origin 510, the latency may be the longest. Regardless of where the data object is found, client 540-1 may be unaware of whether application center 530-1, a data persistence cache server 520-1, or the data object origin server 510-1, had the data object stored.

[0064] While only one data object origin server 510 is present in cloud data persistence system 500-1, more than one data object origin server is possible. For example, a data object origin server may serve a particular region, such as a country. Other data object origin servers may be

present in other countries. These data object origin servers may each maintain identical or substantially similar data sets. Therefore, a data object found on one data object origin server would also be found on other data object origin servers. While not illustrated in FIG. 5, it should be understood that data object origin server 510 may be in communication with a kernel server and/or a data object updater server, such as illustrated in FIGs. 2A, 2B, and 4. Such an arrangement may allow multiple different data object origin servers to have the same data objects stored.

[0065] While the above deals with the communication flow in cloud data persistence system 500-1 when a client requests a data object, a client (or, of course, an end user acting through the client) may also modify a data object. Depending on the data object, a request for a particular data object may be much more frequent than a request to modify a data object. By way of example only, consider a video clip posted on an online video sharing website. If the video is popular, data object information, such as the video owner's name and the video's title may be sent to millions of different clients. However, the end user who created the video may only infrequently modify (via a client) data linked to the video.

[0066] In cloud data persistence system 500-1, the request to modify a data object, as opposed to a request to retrieve a data object, may be treated differently. If client 540-1 makes a request to application center 530-1 to modify a data object, application center 530-1 (assuming the end user and/or client has the proper permissions to modify the data object) may route the request to Kernel application center 111. Kernel application center 111 may then update data object origin server 510, and any other data object origin servers requiring updating. Application center 530-1 may not modify or delete the previous version of the data object if it is stored at application center 530. The data object may then be updated at data object origin server 510. While the data object may now be updated at data object origin server 510, other application centers and data persistence cache servers, such as application center 530-3 and data persistence cache 520-1, may still have out-of-date versions of the data object. Therefore, upon receiving the modifications to the data object, kernel application center 111 may notify data persistence cache servers 520 to no longer use their current version of the data object. This may involve data persistence cache servers 520 deleting their out-of-date version of the data object. In turn, data persistence cache servers 520 may notify application centers 530 to similarly no longer use their out-of-date versions of the data object. This may again involve the out-of-date data object being deleted from application centers 530. Application centers 530 and data persistence cache servers

520 may not receive a new version of the data object along with the notification. Rather, application centers 530 and data persistence cache servers 520 may only receive the new version of the data object if a client requests the data object and it is routed through the data persistence cache server and the application center.

5 [0067] If kernel application center 111 knows what particular data objects are stored at data persistence cache servers 520, a notification may only be sent to the data persistence cache server known to store the data object. Similarly, if data persistence cache servers 520 know what data objects are stored at application centers 530, only those application centers that have out-of-date versions of the data object may be sent the notification. In some embodiments, whenever a
10 modification of a data object is received at kernel application center 111, all data persistence cache servers 520 are notified, which in turn notify all application centers 530. In such embodiments, therefore, kernel application center 111 does not need to be aware of what particular data objects are stored at data persistence cache servers 520, and data persistence cache servers 520 do not need to be aware of what data objects are stored at application centers
15 530. In some embodiments, Kernel application center 111 notifies both data persistence cache servers 520 and application centers 530.

[0068] Referring to FIG. 5B, a simplified block diagram of an embodiment of a cloud data persistence system 500-2 is illustrated. Such a system may be substantially similar to system 500-1. However, system 500-2 illustrates how various components of system 500-2 may be
20 geographically separated. It should be understood that each component of system 500-2 may be located at a different physical location. In some embodiments, client 540-1 is at location 551 while client 540-2 is at location 552. Locations 551 and 552 may be close together, possibly only several miles if both client 540-1 and client 540-2 are using the same application center. Clients 540-1 and 540-2 may be a great distance apart if they are using different application
25 centers.

[0069] Application centers 530-1, 530-2, 530-3, and 530-4 are shown as different locations of 553, 554, 555, and 556, respectively. Application centers that use the same data persistence cache server may be closer together than application centers that do not use the same data persistence cache servers. Further, as those with skill in the art will recognize, while an
30 application server may not be physically closer to a data persistence cache server, due to the state

of the network, communication between two physically further apart servers may be quicker. In such a case, the further servers may be in communication as opposed to the closer servers.

[0070] Data persistence cache server 520-1 is at another location 557. Location 557 may be near or in between location 553 and 554. Data persistence cache server 520-2 is at the same location 555 as application center 530-3. In some embodiments, data persistence cache server 520-2 is at some other location from application center 530-3. Finally, data object origin server 510 may be present at location 558. This location may be at the kernel application center 111, or may be separate.

[0071] FIG. 6 illustrates a simplified swim diagram of an embodiment of a method 600 for responding to a client request to retrieve a data object. Such a method may use a cloud data persistence system, such as cloud data persistence system 500-1 of FIG. 5A. Alternatively, method 600 may rely on some other cloud data persistence system. The embodiment of FIG. 6 presupposes that the data object requested is not present on either the application center contacted or the cache server contacted. As described in relation to FIG. 5, if one of those servers has the data object stored, the method would differ.

[0072] At block 605, a request may be sent from a client to an application center. The client may be client 540-1 or 540-2 of FIG. 5, or may represent some other client. Similarly, the application center of FIG. 6 may represent any of application centers 530 of FIG. 5, or may represent some other application center. The request sent by the client at block 605 of FIG. 6 may contain a URI, possibly a URL, indicating the data object the client is requesting to retrieve. The request may be received either directly or via a network (such as the Internet), by the application center at block 610.

[0073] At block 615, the application center may determine if the data object requested by the client is present. This may involve the application center searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the application center may search an index of all the data objects it has stored. In this embodiment, the application center does not have the data object stored locally. Therefore, at block 620, a request is sent from the application center to a data persistence cache server. This request may contain the same URI received in the request from the client at block 610. The cache server contacted may be a data persistence cache server 520 of FIG. 5A or may represent some other data persistence cache

server. The request by the application center may be received by the data persistence cache server either directly or via a network (such as the Internet) at block 625.

[0074] At block 630, the data persistence cache server may determine if the data object requested by the application center is present. This may involve the data persistence cache server searching a storage device, such as a hard drive or solid state drive for the data object. In some embodiments, the data persistence cache server may search an index of all the data objects it has stored. In this embodiment, the data persistence cache server does not have the data object stored locally. Therefore, at block 635, a request is sent from the data persistence cache server to an data object origin server. This request may contain the same URI received in the request from the client at block 610 and the application center at block 625. The data object origin server contacted may be the data object origin server 510 of FIG. 5A or may represent some other data object origin server. The request by the data persistence cache server may be received by the object origin server either directly or via a network (such as the Internet) at block 640.

[0075] At block 645, the data object origin server may locate the data object requested.

Assuming that the URI indicating the data object initially requested by the client is valid, the object origin server should have the data object stored. If the data object is not present at the data object origin server, the data object may be assumed to not exist or that the URI submitted by the client is incorrect. To retrieve the data object, the data object origin server may search one or more local (or remote) hard drives or solid state drives for the data object. In some embodiments, the data object origin server uses an index to locate and retrieve the data object. Once the data object has been located, at block 650 it may be transmitted to the data persistence cache server.

[0076] At block 655, the data persistence cache server may receive the data object. The data received from the data object origin server may also include the URI sent in the request at block 635. At block 660, the data persistence cache server may determine whether the data object should be stored locally. If the data persistence cache server stores the most recent data objects requested, the data persistence cache server may always store the data object when it is received from the data object origin server. If the data persistence cache server stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested more frequently. As those with skill in the art will understand, there are other factors that may be used to determine whether the data object is stored locally at the data persistence

cache server. As another example, the decision as to whether to store the data object may be based partially on frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored or not, it may be transmitted to the application center at block 665.

5 [0077] At block 670, the application center may receive the data object. This may be the same application center that received the request at block 610. The data received from the data persistence cache server may also include the URI sent in the request by the application center at block 620. At block 675, the application center may determine whether the data object should be stored locally. The factors used to consider whether to store the data object may be the same or
10 different from the factors used at the data persistence cache server. Again, if the application center stores the most recent data objects requested, the application center may always store data objects received from the data persistence cache server. If the application center stores only the most frequently requested data objects, the data object may not be stored if other data objects are requested by clients more frequently. As those with skill in the art will understand, there are
15 other factors that may be used to determine whether the data object is stored locally at the application center. As another example, the decision as to whether to store the data object may be based partially on the frequency of requests for the data object and partially on the data object being recently requested. Whether the data object is stored by the application center or not, it may be transmitted to the client at block 680. Further, to be clear, while it is possible that the
20 same data objects may be stored at both the application and the data persistence cache server, it is likely that at least some different data objects would be stored at the application center than the data persistence cache server. Also, the amount of storage available at the application center and the data persistence cache server may vary. For example, the data persistence cache server may have 50% more storage space than an application center; therefore, it may store roughly 50%
25 more data objects.

[0078] At block 685, the client may receive the data object it requested at block 605. The client may be unaware as to whether the data object was present at the application center, the data persistence cache server, or the data origin cache server. To the client, it may appear as if the client is only interacting with the application center.

30 [0079] While FIG. 6 illustrates an embodiment of a method of a client requesting a data object, assuming the associated end user and/or client has the proper permissions, the end user and/or

client may also request to modify a data object. FIG. 7 illustrates a swim diagram of a simplified embodiment of a method 700 of a client modifying a data object. Method 700 may be conducted using a cloud data persistence system, such as cloud data persistence system 700 of FIG. 7. Alternatively, some other cloud data persistence system may be used. A modification may involve changing data within a data object, adding data to a data object, or deleting the data object, to name only a few examples. It should be understood that in some embodiments, modifications to data objects are much less frequent than requests to retrieve a data object. At block 705, the client may transmit the request to modify the data object to an application center. The client may be the same client as detailed in FIG. 6 or client 540-1 or 540-2 of FIG. 5. Alternatively, the client of FIG. 7 may represent some other client. Similarly, the application center of FIG. 7 may represent the application center of FIG. 6 or any of application centers 530 of FIG. 5. The application center of FIG. 7 may also represent some other application center.

[0080] At block 710, the application center may receive the modification request from the client. The request may include a URI (such as a URL) indicating the data object the client is attempting to modify. The request may also include the changes, additional information, or an indication to delete the data object. If the application center determines it has the data object stored, at block 712, the application center may delete or block access to the data object so that other clients do not receive an out-of-date version of the data object. At block 715, a modification request may be transmitted from the application center to the data object origin server. This transmission may be routed from the application to the data object origin server (e.g., not through the data persistence cache server) or may be relayed by the data persistence cache server.

[0081] At block 720, the kernel application center may receive the modification request. The kernel application center may be the kernel application center 111 of FIG. 5A and FIG. 5B. Kernel application center of FIG. 7 may also represent a data object origin server. After receiving the modification request at block 720, the kernel application center may modify the data object at one or more data object origin servers at block 725. Data object origin servers may always have a stored version of the data object because they maintain a stored version of all valid data objects. If the data object origin server does not have a version of the data object requested to modify, it may be assumed that the data object does not exist or the URI sent by the client at block 705 was incorrect. The modification at block 725 may involve the data object being changed, having data added to it, or the data object being deleted. At block 730, the modified

data object may be stored at one or more data object origin servers. Obviously, if the data object is being deleted, this step may not occur.

5 [0082] At block 735, a notification may be sent by the kernel application center that indicates that the data object has been modified. Of note, this modification may not contain the modified data object, but rather may contain only information that indicates that previous versions of the data object are no longer valid. This notification may be transmitted to one or more data persistence cache servers at block 735. While the swim diagram illustrates the notification being sent to only one data persistence cache server, this is for simplicity only. The notification transmitted by the data object origin server may be sent to all data persistence cache servers (or 10 those data persistence cache servers that have the previous version of the data object stored) in communication with the data object origin server. The notification may only be sent to those data persistence cache servers that have an out-of-date version of the data object if the kernel application center is aware of which data persistence cache servers have the data object stored. This may be possible if the data persistence cache server informs the kernel application center of 15 its stored data objects. Alternatively, all data persistence caches may receive the notification whether they have the previous version of the data object stored or not. If a data persistence cache server does not have the previous version of the data object stored, the data persistence cache server may ignore the notification.

[0083] At block 740, the notification may be received by the data persistence cache server. 20 The data persistence cache server may be the data persistence cache server of FIG. 6 and/or one of the data persistence cache servers 520 of FIG. 5. The data persistence cache server of FIG. 7 may represent some other data persistence cache server. At block 745, the previous version of the data object may be cleared (e.g., deleted or otherwise removed) from the data persistence cache server. At block 750, a notification that the previous version of the data object is out-of- 25 date may be transmitted to one or more application centers at block 750.

[0084] While the swim diagram illustrates the notification being sent to only one application center, this is for simplicity only. The notification transmitted by the data persistence cache server may be sent to all application centers (or those application centers that have the previous version of the data object stored) in communication with the data persistence cache server. The 30 notification may only be sent to those application centers that have the previous version of the data object if the data persistence cache server is aware of which application centers have the

previous version of the data object stored. This may be possible if the application center informs the data persistence cache server (or the kernel application center and/or data object origin server) of its stored data objects. Alternatively, all application centers may receive the notification whether they have the previous version of the data object stored or not. If an
5 application center does not have the previous version of the data object stored, the application center may ignore a received notification. The notification sent by the data persistence cache server at block 740 may represent the same or a similar notification as sent by the kernel application center at block 735. For example, the notification may include a URI (possibly a URL) indicating the data object that has been modified. At block 760, the previous version of
10 the data object may be cleared (e.g., deleted or otherwise demarked) from the application center.

[0085] While the previous description and figures relate to how a client may retrieve and modify a data object, the storage of data objects and other data and/or files may be more sophisticated than one server (e.g., an application center, a data persistence cache server, or a data object origin server) having one or more storage devices. For example, each application
15 center of FIG. 5A may represent a cluster of servers, each connected to one or more storage devices, such as hard drives or solid state drives. FIG. 8 illustrates a simplified block diagram of an embodiment of an asset management and storage system 800 showing how various storage devices may be used to store data and accessed. While such a system may be used to store data objects, such as those data objects described previously, such a system may be used to store other
20 forms of electronic data, such as media assets. A media asset may be a media file, such as a movie file, audio file, or image file.

[0086] In system 800, multiple different storage devices 810 may be present. These storage systems may be servers having storage devices such as hard drives and/or solid state drives. Storage devices 810-1 and 810-2 may be located in the same location or may be physically in
25 different locations. Each of storage devices 810 may be linked to one or more storage servers 820. These storage servers 820 may not be physical servers, but rather logical abstractions of servers. Alternatively, they may be physically separate servers. Various storage drives on storage servers 820 may be mapped to different storage devices. For example, storage server 820-1 is linked to one storage drive. The storage drive is present on storage device 810-1.
30 Storage server 820-1 and storage device 810-1 may be directly connected, or may be connected via a network, such as the Internet or a private corporate intranet. In system 800, storage server 820-2 has two storage drives. A first storage drive of storage server 820-2 maps to storage

device 810-1. Therefore, in the embodiment of the system 800, storage device 810-1 is mapped to two different storage servers. If, for example, storage device 810-1 has storage totaling 1 TB, this gross storage space may be divided among the two drives mapped to it. For example, the storage drive of storage server 820-1 mapped to storage device 810-1 may be 300 GB. And, the storage drive of storage device 820-2 mapped to storage device 810-1 may be 700 GB. Storage server 820-2 may also have another storage drive. The storage drive is mapped to storage device 810-2. In this embodiment, storage device 810-2 is only mapped to storage server 820-2. Therefore, if storage device 810-2 has total storage of 2 TB, this entire space may be mapped to the linked storage drive of storage server 820-2. Or, as another example, only 800 GB of storage device 810-2 may be mapped to storage server 820-2, with the remaining space being free space.

[0087] Also, different storage devices may require communication, such as requests for media assets, to use varying protocols. For example, storage device 810-1 may be configured to accept file requests via FTP. However, storage device 810-2 may use MICROSOFT WINDOWS's file copy. It should also be understood that other protocols may also be used. Therefore, depending on what protocol a storage device accepts, the storage server links to that storage device may be configured to communicate in the proper protocol. This may be especially useful if the entity operating the storage servers 820 does not control one or more of storage devices 810. For example, storage device 810-2 may be maintained by some other entity, such as a client of the entity operating system 800.

[0088] Other logical abstractions that may be present are storage systems 830. Each storage system, such as storage system 830-1, may contain one or more than one storage servers 820. In system 800, storage system 830-1 includes two storage servers: 820-1 and 820-2. Therefore, a request for a particular piece of data, such as a media asset, received at storage system 830-1, may not indicate the storage server or the storage device where the media asset is located. However, upon receiving the request (possibly in the form of a URL) at storage system 830-1 for the media asset, it may be able to determine the appropriate storage server to contact. In turn, this storage server may be able to determine the appropriate storage drive (linked to a storage device) that contains the media asset. The logical abstractions of the storage system 830 and storage servers 820 may reside on the same physical servers as storage devices 810. In some embodiments, storage server 820 and storage system 830 are present on different servers than storage device 810. Also, storage systems 830 may be physically separate servers from storage servers 820.

[0089] Yet another logical abstraction may be present as storage cloud 840. Storage cloud 840 may include one or more storage systems. Storage cloud 840 may be viewed as the entire storage system. In system 800, storage cloud 840 includes storage systems: 830-1 and 830-2. Therefore, storage cloud 840 may contain all storage systems 830 in system 800. In turn, storage systems 830 may contain all storage servers 820 in system 800. Further, storage server 820 may be linked to all storage devices 810 in system 800.

[0090] FIG. 9 illustrates another simplified block diagram of an embodiment of a system 900 showing how various storage devices may be accessed. System 900 of FIG. 9 may represent the same system as system 800 of FIG. 8, or may represent a different system. In the embodiment illustrated, system 900 contains four storage devices: 910-1, 910-2, 910-3, and 910-4. These storage devices may represent any of storage devices 810 of FIG. 8. Also, these storage devices may represent different storage devices. As will be understood to those with skill in the art, fewer or more storage devices may also be possible. In the illustrated embodiment, storage device 910-1 and storage device 910-2 are illustrated as being operated by first entity 915. First entity 915 may be the entity that operates other portions of system 900, or is a client of the entity that operates other portions of system 900. These storage devices, 910-1 and 910-2, may be co-located with one or more storage servers 920, or may be geographically separated. Further, storage device 910-1 and storage device 910-2 may be present at the same location, or may be geographically separated. Storage device 910-3 may be operated by a second entity 916. In some embodiments, the second entity 916 may be a client of the first entity. In system 900, storage device 910-3 may communicate with other systems 917 of the second entity 916. Therefore, while storage servers 920 may be able to access storage device 910-3, client system 917 may also access and/or modify data stored on storage device 910-3. Storage device 910-4 is illustrated as part of a third entity 918. This entity may be an entity that sells or leases storage on their system. While FIG. 9 illustrates four storage devices operated by various entities, those skilled in the art will recognize that this arrangement is merely for exemplary purposes. Many other arrangements of storage devices, operated by one or more entities may be possible.

[0091] Whether operated by three different entities, as illustrated, or by one entity, storage devices 910 may communicate with multiple storage servers 920. System 900 includes four storage servers: 920-1, 920-2, 920-3, and 920-4. The storage servers 920, which may be logical abstractions, may communicate with the various storage devices 910 via a direct connection (such as 920-1 to storage device 910-1) or via a network (such as 920-2 to storage device 910-3).

The network may be a network such as the Internet or a private corporate intranet. One storage server may be mapped to one storage device, for example, in FIG. 9, storage server 920-1 only has one storage drive mapped to storage device 910-1. A storage device may also be mapped to multiple storage servers, for example, in FIG. 9, storage servers 920-2 and storage server 920-4 are both mapped to storage device 910-3. Also, one storage server may be mapped to multiple storage devices. For example, storage server 920-4 is mapped to storage device 910-3 and storage device 910-4.

[0092] Also, storage servers 920 may communicate with storage devices 910 via a variety of different protocols. For example, the format of requests to some storage devices may be FTP, while others may be MICROSOFT WINDOWS file copy. Other protocols may also be possible. Therefore, any request for data or a media asset that is received by storage servers 920 does not need to indicate the appropriate protocol to retrieve the file or media asset from the appropriate storage device.

[0093] As in FIG. 8, one or more than one storage servers may be mapped to a storage system 930. Storage systems 930 may represent the same storage systems as storage systems 830 of FIG. 8. Alternatively, storage systems 930 may represent some other storage systems. In system 900 of FIG. 9, storage servers 920-1 and storage server 920-2 are mapped to storage system 930-1. Storage system 930-1 may communicate directly with the storage server (such as the connection between storage system 930-1 and storage server 920-1) or may use a network (such as the connection between storage system 930-1 and storage server 920-2). In system 900, storage system 930-2 may communicate with two storage servers: 920-3 and 920-4. While in system 900 storage systems 930 are shown as communicating with two storage servers each, those with skill in the art will recognize that a storage system may also communicate with one or more than two storage servers.

[0094] Storage systems 930 may make up storage cloud 940. Storage cloud 940 may represent the same storage cloud as storage cloud 840 of FIG. 8. Alternatively, storage cloud 940 may represent some other storage cloud. Therefore, storage cloud 940 may be viewed as including storage systems 930, storage servers 920, and storage devices 910.

[0095] Other systems may interact with various components of storage cloud 940. For example, a content data network 960 may communicate with storage system 930-1. If the client requests the media asset from content data network 960 (CDN) that the CDN does not have

stored (a cache miss), the content data network 960 may request the media asset from storage system 930-1. Storage system 930-1 may then contact the appropriate storage server 920, which may, in turn, retrieve the media asset via the correct protocol from the appropriate storage device. From the viewpoint of CDN 960, it may appear that the media asset is stored at storage system 930-1. While CDN 960 is shown only interacting with storage system 930-1, it should be understood that this is for simplicity only. Multiple CDN's may communicate with storage system 930-1. Further, CDN's may also communicate with storage system 930-2.

[0096] While CDN 960 may communicate with storage system 930-1, file replication service 950 may communicate directly with storage servers 920. File replication service 950 may be responsible for distributing various media assets to storage server 920 that need to be stored for possible future retrieval. While file replication service 950 is shown as connected to only storage server 920-1, it should be understood that file replication service 950 may be in communication with multiple storage servers 920. Besides file replication service 950, other services may interact directly with storage server 920.

[0097] FIG. 10 illustrates a simplified embodiment of a method for retrieving data (such as a media asset) from a storage cloud system. Such a method may be used in conjunction with a storage cloud system such as system 800 of FIG. 8, or system 900 of FIG. 9. The method 1000 may also be used in conjunction with some other storage cloud system. At block 1010, a request may be received for a piece of data, such as a media asset. Such a request may be received from an entity such as a content data network or some other entity accessing the storage cloud.

[0098] At block 1015, if the storage cloud contains more than one storage system, the appropriate storage system that contains the data requested may be identified. At block 1020, a request for the data may be sent to the appropriate storage system. At block 1025, the request may be received at the appropriate storage system. At block 1030, the storage system may determine the appropriate storage server that stores the data requested. Such a determination may involve an index being searched by the storage system to determine which storage server to contact. If the storage system only has one link to a storage server, such a determination may not be necessary. At block 1035, the request for the data may be sent to the appropriate storage server. This request may be received by the storage server at block 1040. At block 1045, a storage server may determine the appropriate storage device that stores the requested data. This step may not be necessary if only one storage device is linked to the storage server. At block

1050, the storage server may send a request for the data to the appropriate storage device. This request may be formatted using the appropriate protocol for the storage device receiving the request.

5 [0099] At block 1055, the request may be received at the appropriate storage device in the appropriate protocol. The storage device may then retrieve the appropriate data. At block 1060, the requested data may be transmitted from the storage device and received at the storage server at block 1065. The storage server may then transmit the requested data at block 1070, the requested data is then received by the storage system at block 1075. At block 1080, the requested data may be transmitted from the storage system to the requesting party, in this case
10 the CDN.

[0100] It should be noted that the methods, systems, and devices discussed above are intended merely to be examples. It must be stressed that various embodiments may omit, substitute, or add various procedures or components as appropriate. For instance, it should be appreciated that, in alternative embodiments, the methods may be performed in an order different from that
15 described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain embodiments may be combined in various other embodiments. Different aspects and elements of the embodiments may be combined in a similar manner. Also, it should be emphasized that technology evolves and, thus, many of the elements are examples and should not be interpreted to limit the scope of the invention.

20 [0101] Specific details are given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments. This description provides example embodiments
25 only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the preceding description of the embodiments will provide those skilled in the art with an enabling description for implementing embodiments of the invention. Various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention.

30 [0102] Also, it is noted that the embodiments may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential

process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional steps not included in the figure. Furthermore, embodiments of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the necessary tasks may be stored in a computer-readable medium such as a storage medium. Processors may perform the necessary tasks.

[0103] Having described several embodiments, it will be recognized by those of skill in the art that various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the invention. For example, the above elements may merely be a component of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of steps may be undertaken before, during, or after the above elements are considered. Accordingly, the above description should not be taken as limiting the scope of the invention.

[0104] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0105] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge in Australia.

WHAT IS CLAIMED IS:

- 1 1. A cloud data persistence system for distribution of data, the cloud data
- 2 persistence system comprising:
- 3 a first server which stores a first plurality of data objects and is communicatively
- 4 coupled with a plurality of cache servers, wherein:
- 5 the first server is located at a first location;
- 6 each of the first plurality of data objects comprise a link to one of a
- 7 plurality of media assets; and
- 8 the plurality of media assets are stored at a second location separate from
- 9 the first location;
- 10 a second server that stores a second plurality of data objects and is configured to
- 11 request a data object from the first server if the data object is not present in the second plurality
- 12 of data objects, wherein:
- 13 the second server is one of the plurality of cache servers; and
- 14 the second server is communicatively coupled with a first plurality of
- 15 application servers;
- 16 a third server that stores a third plurality of data objects and is configured to
- 17 request the data object from the first server if the data object is not present in the third plurality
- 18 of data objects, wherein:
- 19 the third server is one of the plurality of cache servers; and
- 20 the third server is communicatively coupled with a second plurality of
- 21 application servers; and
- 22 a fourth server that stores a fourth plurality of data objects and is configured to
- 23 request the data object from the second server if the data object is not present in the fourth
- 24 plurality of data objects, wherein:
- 25 the fourth server is one of the first plurality of application server; and
- 26 the fourth server is configured to receive a request from a first client for
- 27 the data object;
- 28 a fifth server that stores a fifth plurality of data objects and is configured to
- 29 request the data object from the second server if the data object is not present in the fifth plurality
- 30 of data objects, wherein:
- 31 the fifth server is one of the first plurality of application servers; and

32 the fifth server is configured to receive a request from a second client for
 33 the data object;
 34 a sixth server that stores a sixth plurality of data objects and is configured to
 35 request the data object from the third server if the data object is not present in the sixth plurality
 36 of data objects, wherein:
 37 the sixth server is one of the second plurality of application servers; and
 38 the sixth server is configured to receive a request from a third client for the
 39 data object; and
 40 a seventh server that stores a seventh plurality of data objects and is configured to
 41 request the data object from the third server if the data object is not present in the seventh
 42 plurality of data objects, wherein:
 43 the seventh server is one of the second plurality of application servers; and
 44 the seventh server is configured to receive a request from a fourth client
 45 for the data object.

1 2. The cloud data persistence system of claim 1, wherein the second location
 2 comprises a content delivery network.

1 3. The cloud data persistence system of claim 1, wherein:
 2 data objects in the second plurality of data objects are at least partially determined
 3 based on when requests for those data objects were last received at the second server;
 4 data objects in the third plurality of data objects are at least partially determined
 5 based on when requests for those data objects were last received at the third server;
 6 data objects in the fourth plurality of data objects are at least partially determined
 7 based on when requests for those data objects were last received at the fourth server;
 8 data objects in the fifth plurality of data objects are at least partially determined
 9 based on when requests for those data objects were last received at the fifth server;
 10 data objects in the sixth plurality of data objects are at least partially determined
 11 based on when requests for those data objects were last received at the sixth server; and
 12 data objects in the seventh plurality of data objects are at least partially
 13 determined based on when requests for those data objects were last received at the seventh
 14 server.

1 4. The cloud data persistence system of claim 1, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth
4 plurality of data objects, and the seventh plurality of data objects.

1 5. A cloud data persistence system for distribution of data, the cloud data
2 persistence system comprising:
3 a first server which stores a first plurality of data objects and is communicatively
4 coupled with a second server and a third server, wherein:
5 the first server is located at a first location;
6 each of the first plurality of data objects comprises a link to a media asset
7 of a plurality of media assets;
8 the plurality of media assets is located at a second location; and
9 the second location is distinct from the first location;
10 the second server stores a second plurality of data objects and is configured to
11 request a data object from the first server if the data object is not present in the second plurality
12 of data objects, wherein the second server is communicatively coupled with a fourth server and
13 fifth server; and
14 the fourth server stores a third plurality of data objects and is configured to
15 request the data object from the second server if the data object is not present in the third
16 plurality of data objects, wherein the fourth server is configured to receive a request from a first
17 client for the data object.

1 6. The cloud data persistence system of claim 5 wherein the first plurality of
2 data objects comprises all data objects present in the second plurality of data objects and the third
3 plurality of data objects.

1 7. The cloud data persistence system of claim 5, wherein the third server
2 stores a fourth plurality of data objects and is configured to request the data object from the first
3 server if the data object is not present in the fourth plurality of data objects, wherein:
4 the third server and the second server are data persistence cache servers;
5 and

6 the third server is communicatively coupled with a sixth and seventh
7 server.

1 8. The cloud data persistence system of claim 5, wherein the fifth server
2 stores a fifth plurality of data objects and is configured to request the data object from the second
3 server if the data object is not present in the fifth plurality of data objects, wherein:
4 the fifth server and the fourth servers are application servers; and
5 the fifth server is configured to receive requests from a second client for
6 the data object; and
7 the fifth server is geographically separated from the fourth server.

1 9. The cloud data persistence system of claim 5 wherein the request from the
2 first client for the data object comprises a universal resource indicator.

1 10. The cloud data persistence system of claim 5 wherein each of the first
2 plurality of data objects comprise a universal resource locator and a title linked to a media asset
3 of the plurality of media assets.

1 11. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on a frequency of requests for those data objects received from clients.

1 12. The cloud data persistence system of claim 5, wherein data objects in the
2 second plurality of data objects and the third plurality of data objects are at least partially
3 determined based on when requests for those data objects were last received from clients.

1 13. The cloud data persistence system of claim 5, wherein the first plurality of
2 data objects comprises all data objects in the second plurality of data objects, the third plurality
3 of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

1 14. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data, the method comprising:
3 receiving, by an application center, a first request from a client for a data object,
4 wherein:

5 the application center is one of a plurality of application centers that
6 communicate with a plurality of clients;
7 the application center is located at a first location;
8 the data object comprises a link to a media asset; and
9 the media asset is stored at a second location separate from the first
10 location;
11 determining, by the application center, the data object is not stored among a first
12 plurality of data objects stored at the application center;
13 transmitting, by the application center, a second request for the data object to a
14 first cache server, wherein:
15 the first cache server is communicatively coupled with the plurality of
16 application centers; and
17 the first cache server is one of a plurality of cache servers;
18 receiving, by the cache server, the second request for the data object;
19 determining, by the cache server, the data object is not stored among a second
20 plurality of data objects stored at the cache server;
21 transmitting, by the cache server, a third request for the data object to an origin
22 server, wherein the origin server is communicatively coupled with the plurality of cache servers;
23 receiving, by the origin server, the third request for the data object;
24 locating, by the origin server, the data object among a third plurality of data
25 objects;
26 transmitting, by the origin server, the data object linked to the third request to the
27 cache server;
28 receiving, by the cache server, the data object linked to the third request;
29 transmitting, by the cache server, the data object linked to the second request;
30 receiving, by the application center, the data object linked to the second request;
31 and
32 transmitting, by the application center, the data object linked to the first request to
33 the client.

1 15. The method of claim 14 further comprising, after receiving the first data
2 object from the origin server, storing, by the cache server, the first data object.

1 16. The method of claim 14 further comprising, after receiving the first data
2 object from the cache server, storing, by the application center, the first data object.

1 17. The method of claim 14 wherein the cache server is located at a third
2 location.

1 18. The method of claim 14 wherein the first request for the data object
2 comprises a universal resource indicator.

1 19. The method of claim 14, wherein data objects in the first plurality of data
2 objects and the second plurality of data objects are at least partially determined based on when
3 requests for those data objects were last received from clients.

1 20. The method of claim 14, wherein the third plurality of data objects
2 comprises all data objects in the first and the second pluralities of data objects.

1 21. A cloud data persistence system for distribution of data substantially as
2 herein described.

1 22. A method for retrieving data objects using a cloud data persistence system
2 configured for distribution of data substantially as herein described.

ABSTRACT OF THE DISCLOSURE

Various cloud data persistence systems and methods are described. In some embodiments, a client may request a data object from a server. This data object may contain a link to a media asset and other information linked to the video file, such as a title and duration. This media asset may be stored in a different location. If this first server does not have the data object stored locally, it may contact a second server to determine if it has the data object stored. If the second server does not have the data object stored, it may contact a data object origin server, which may have all existing data objects stored. This data object origin server may then send the data object to the second server, which may in turn transmit the data object to the first server. The first server may then transmit the data object to the client.

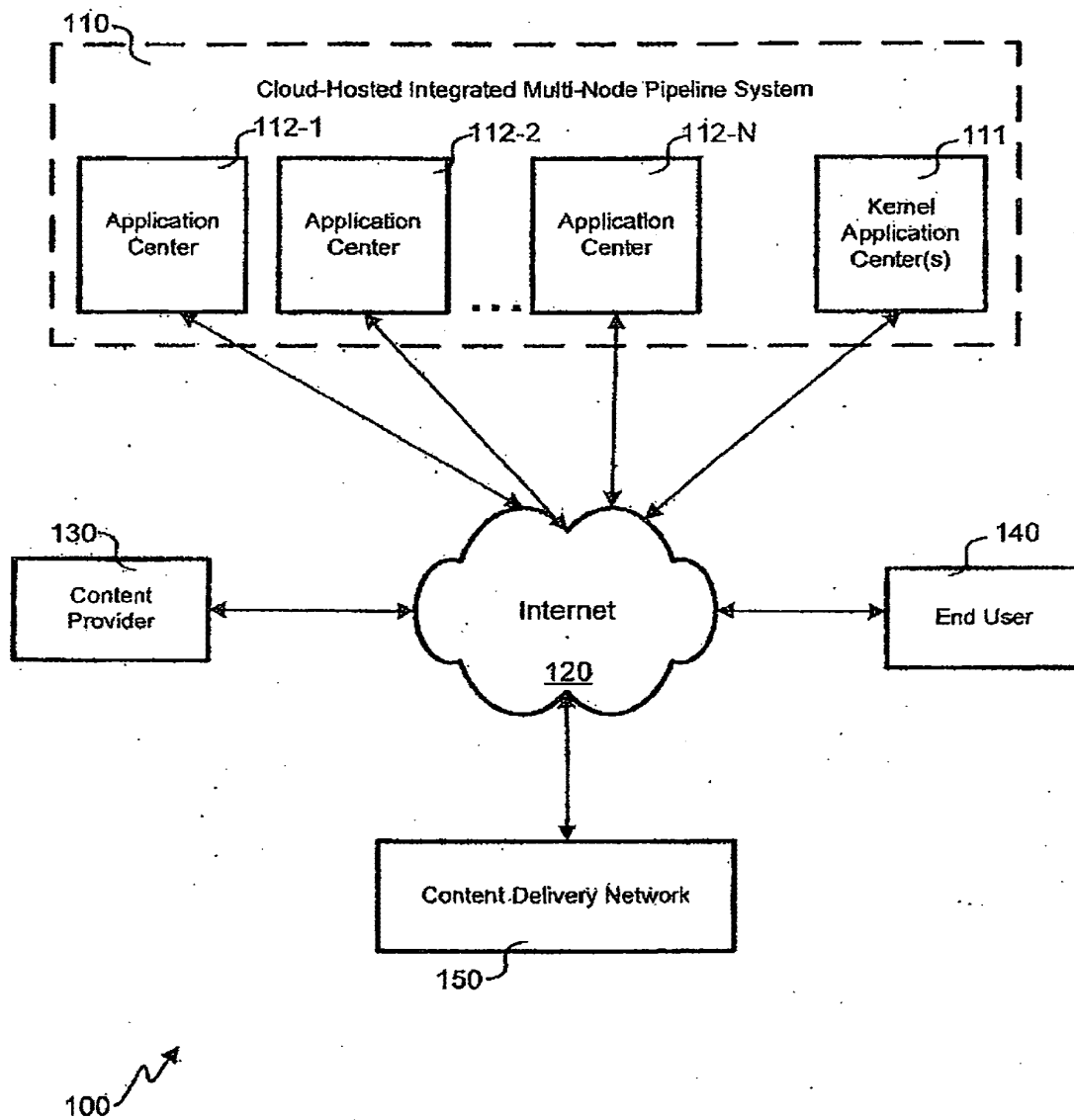


FIG. 1

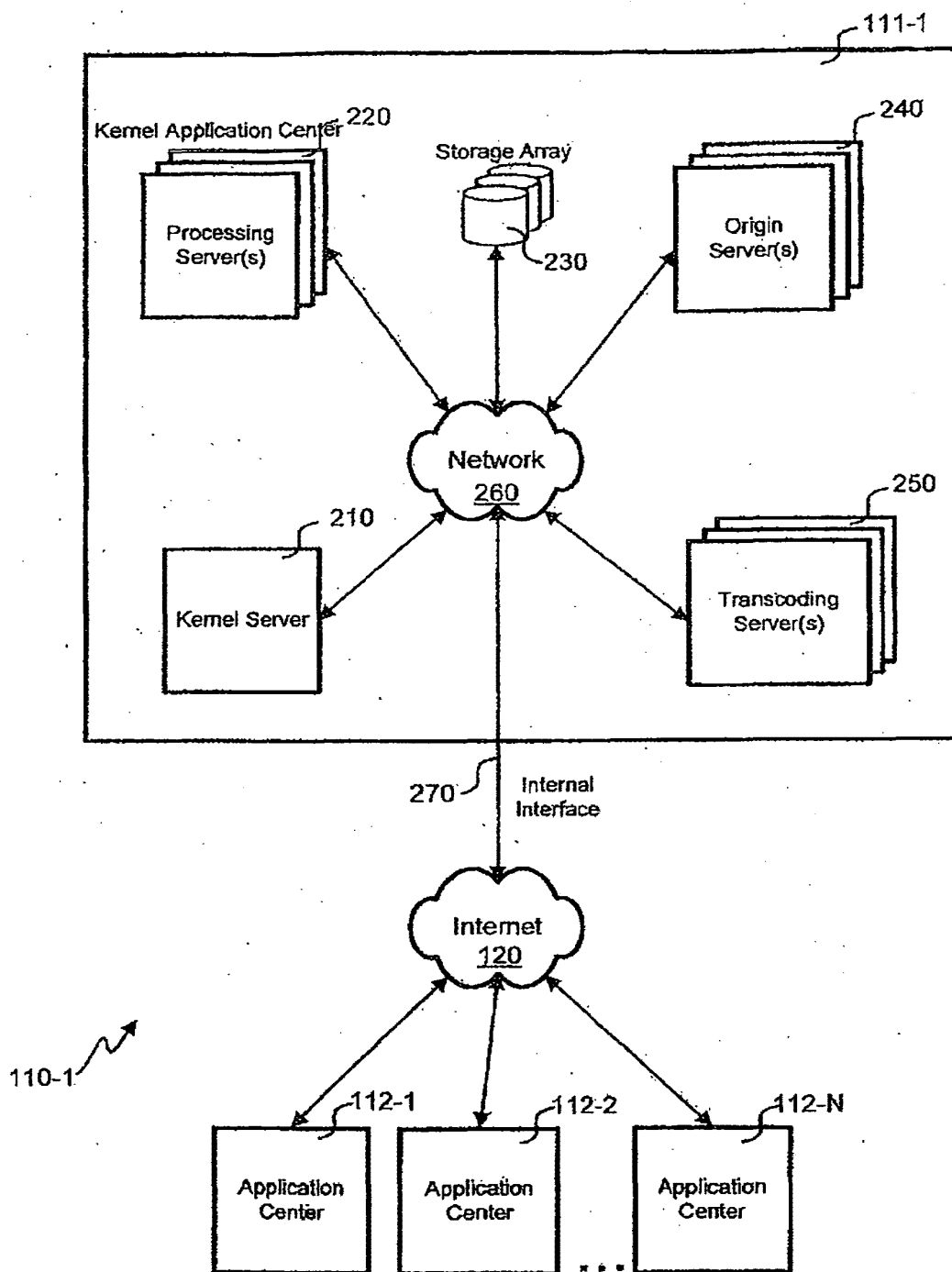


FIG. 2A

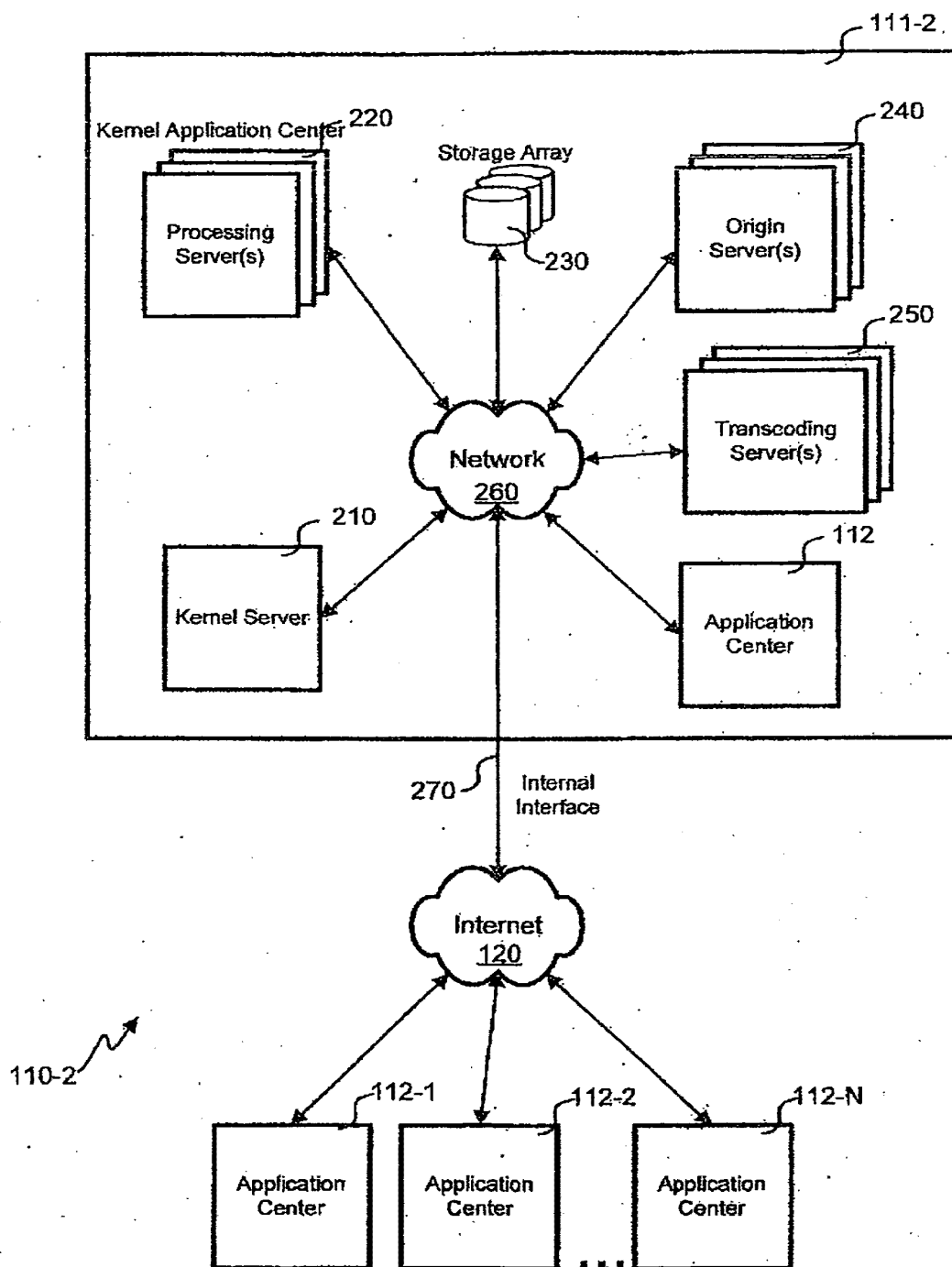


FIG. 2B

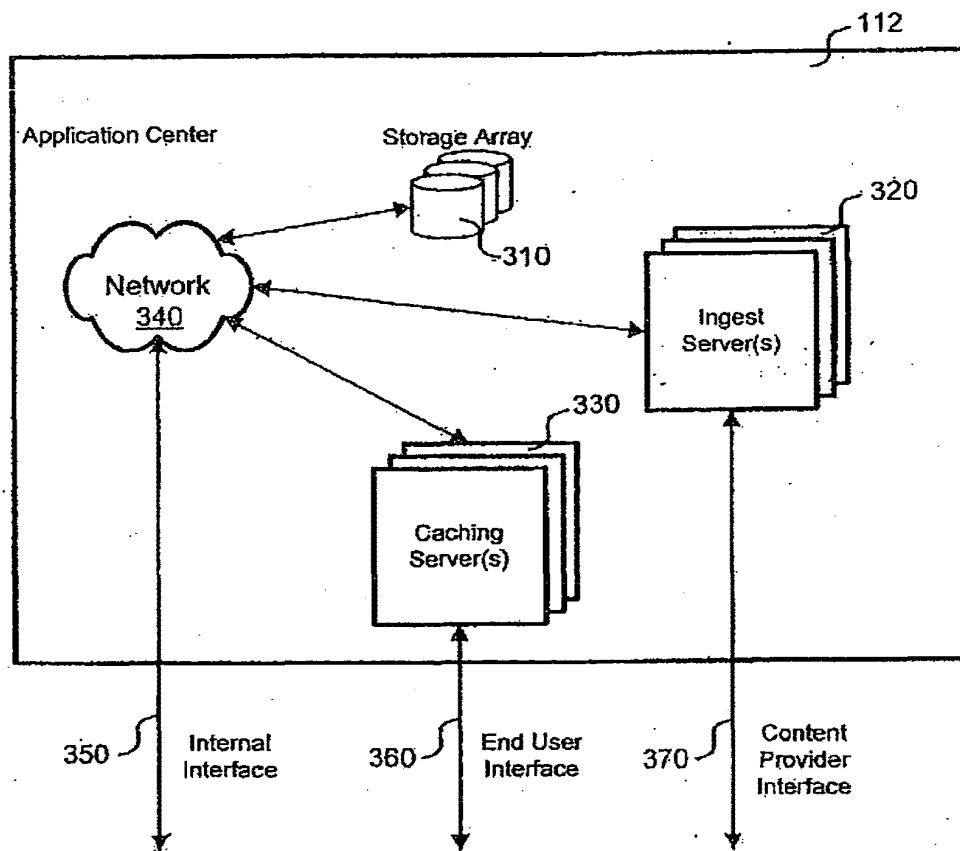


FIG. 3

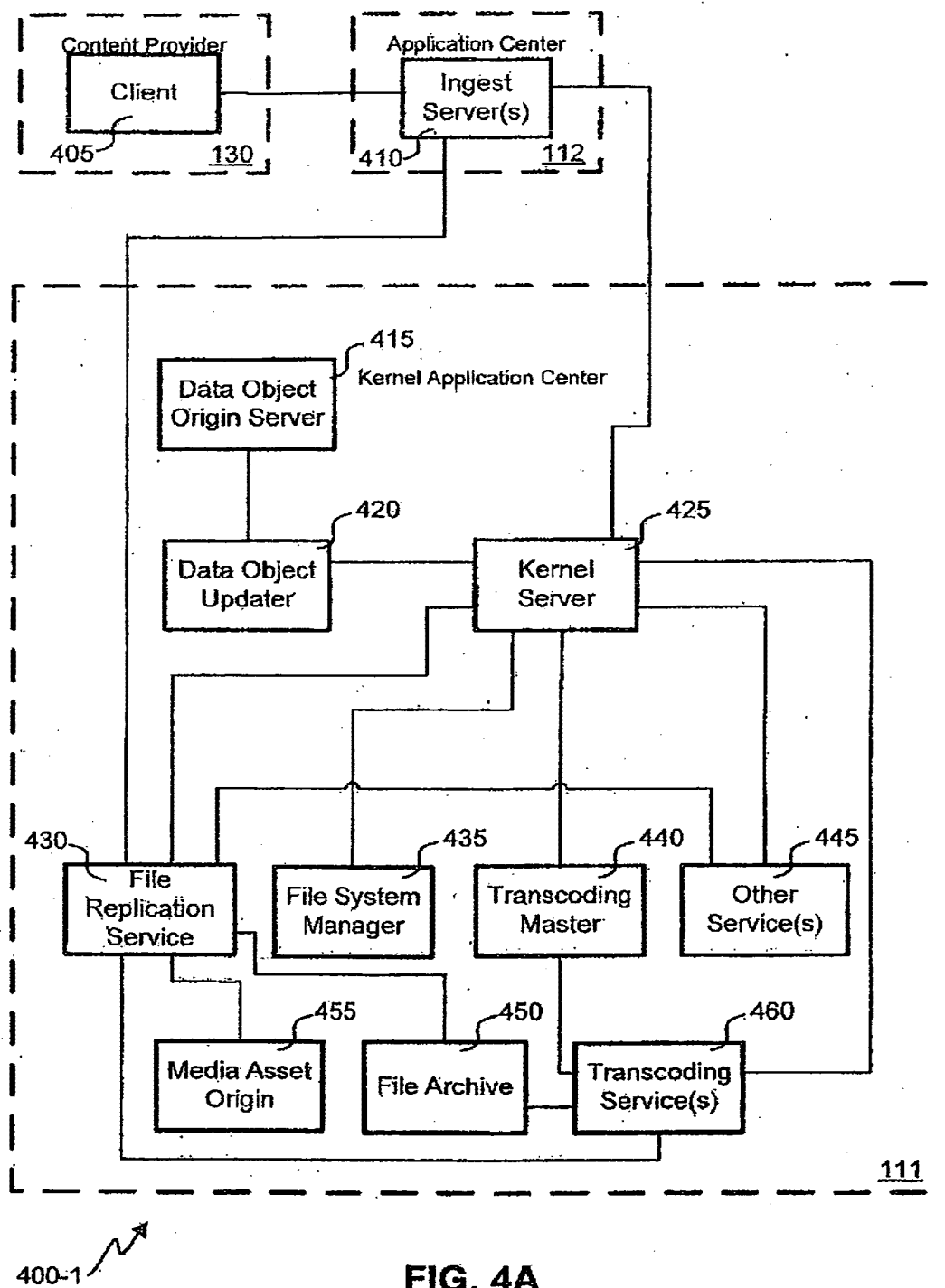


FIG. 4A

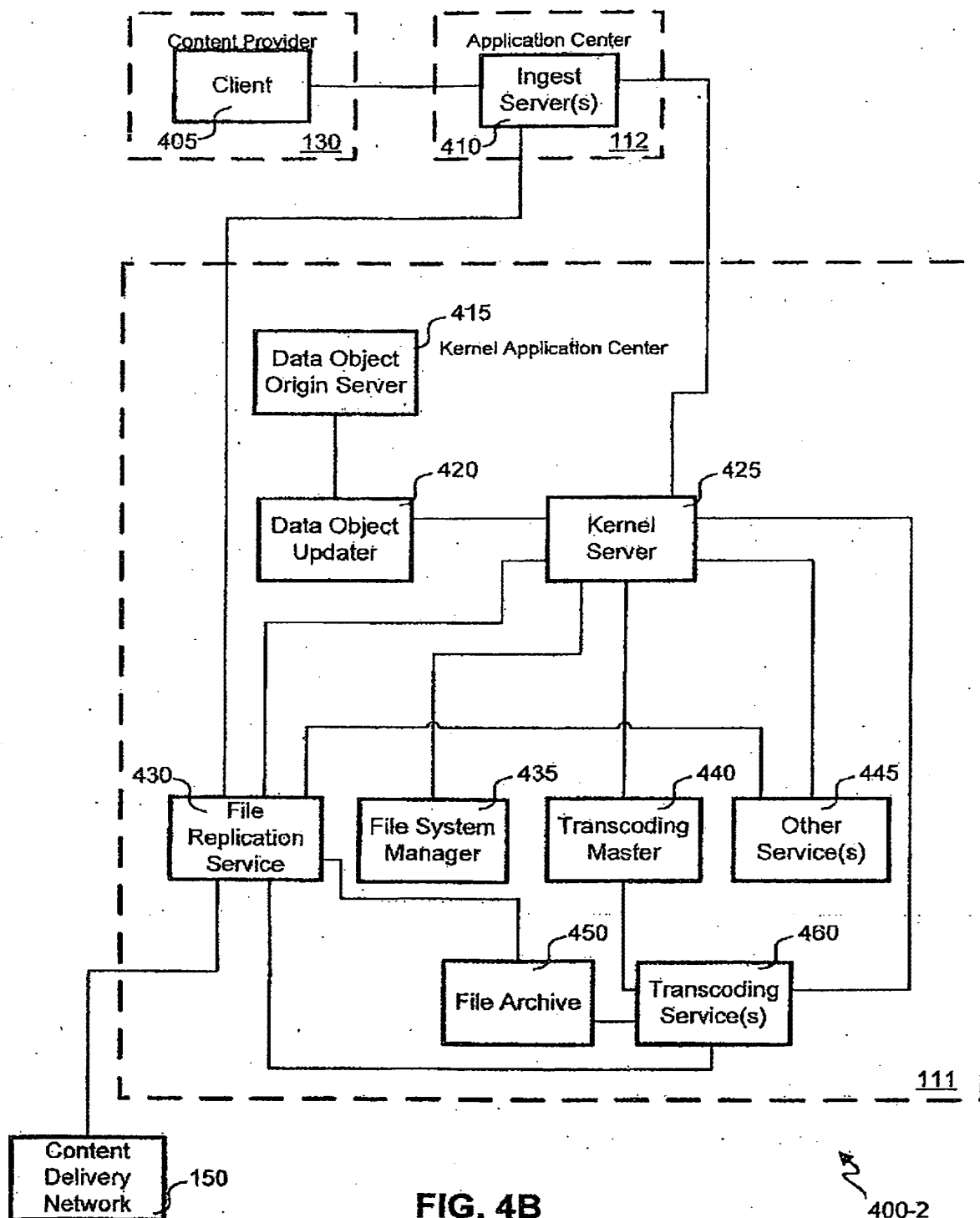
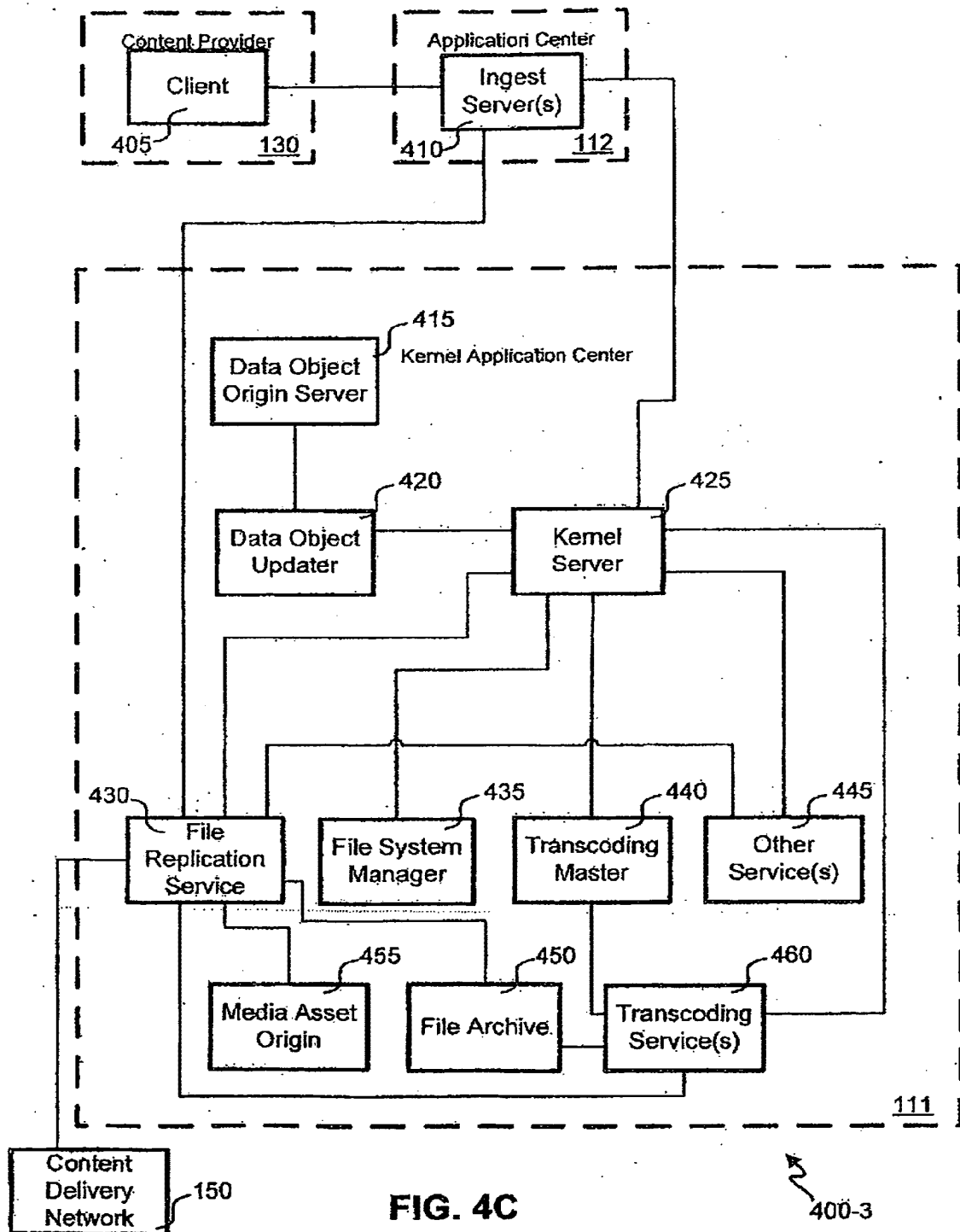


FIG. 4B



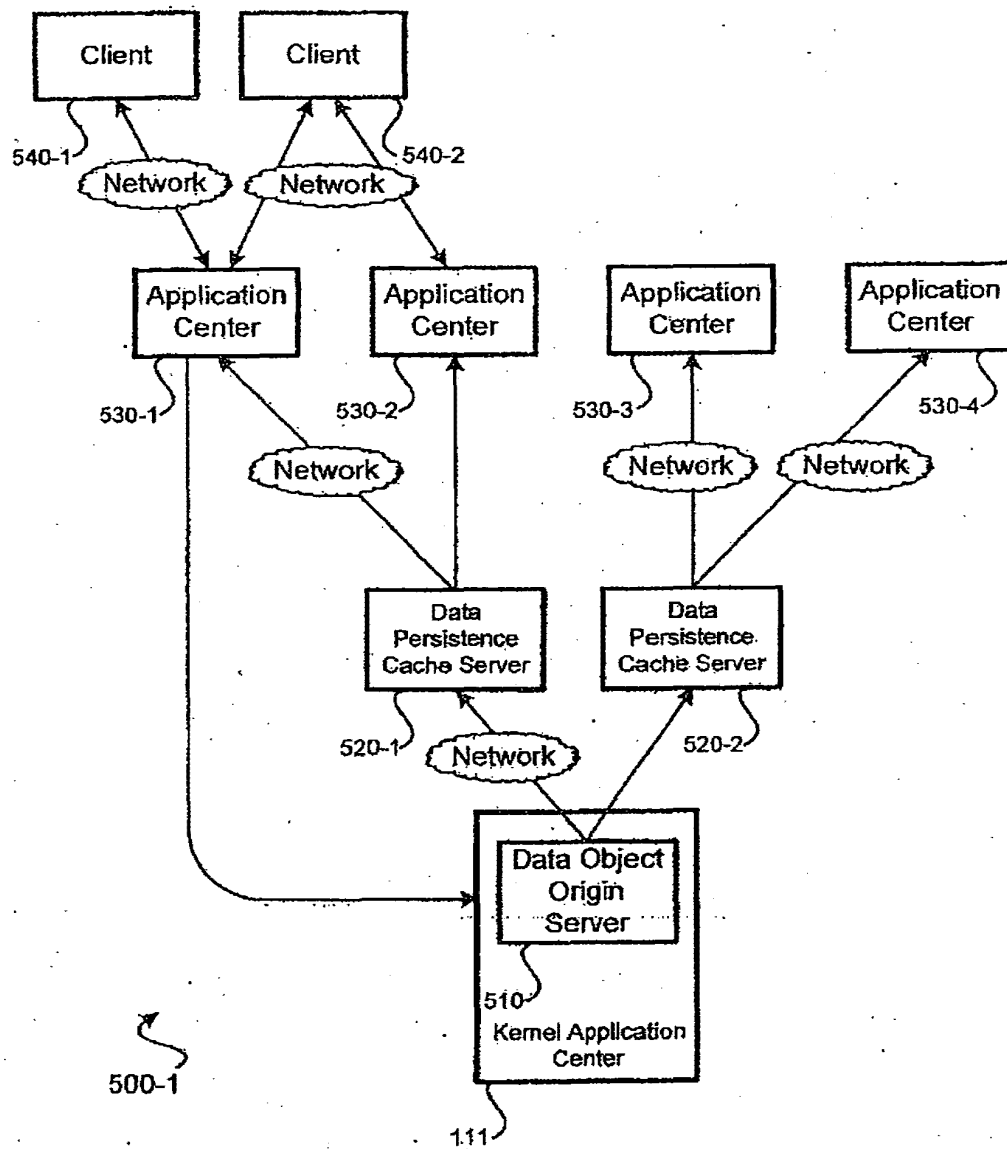


FIG. 5A

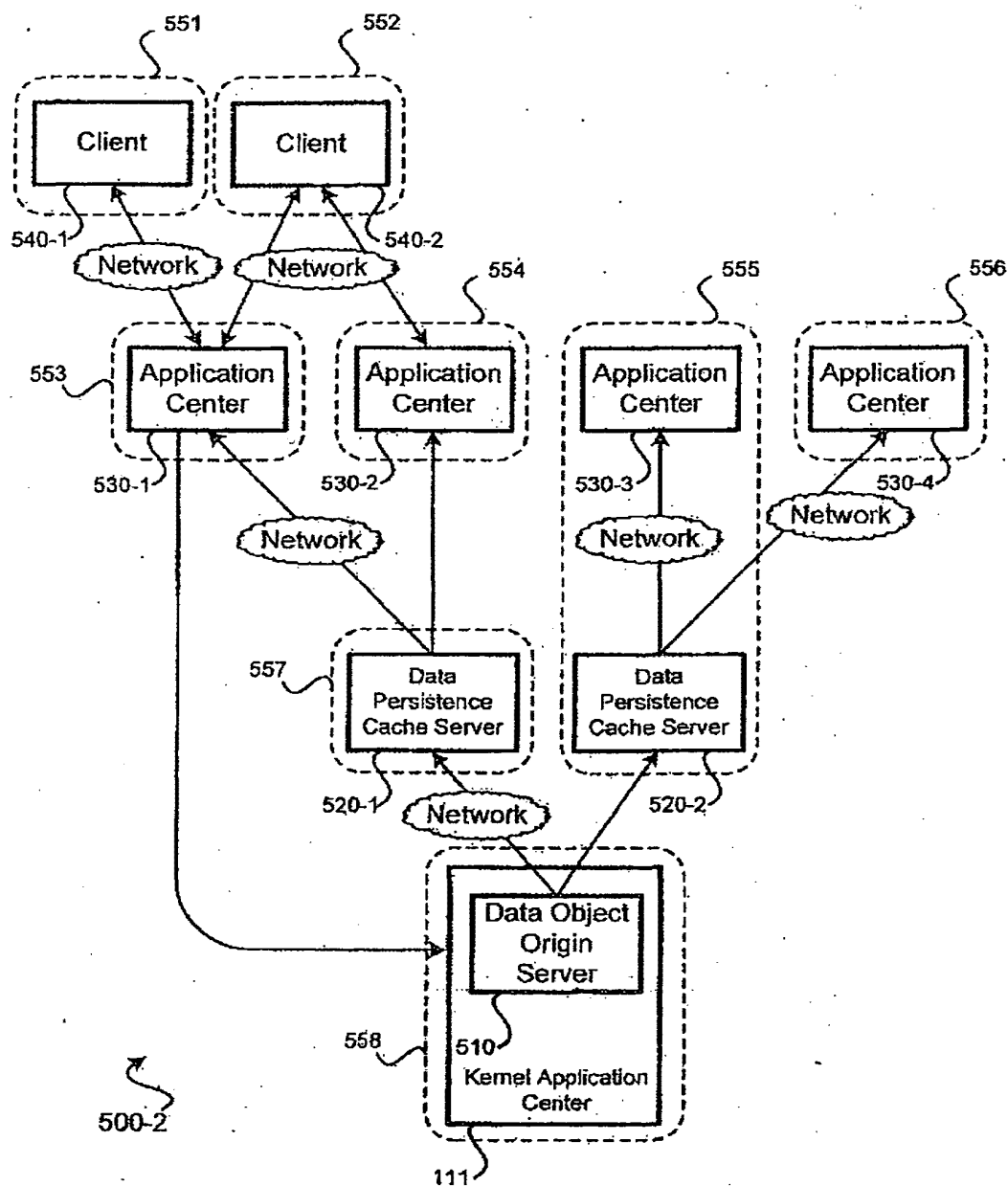
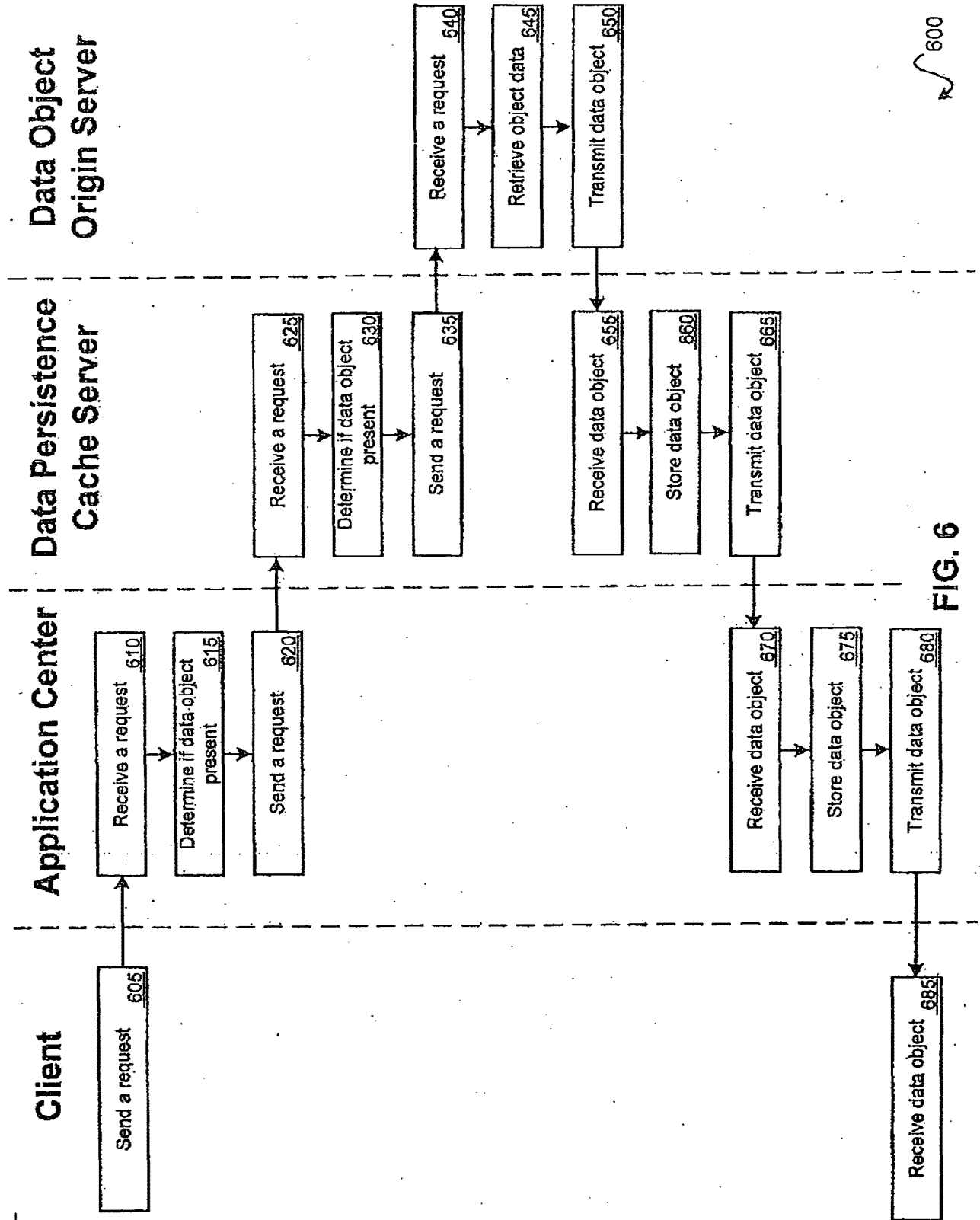
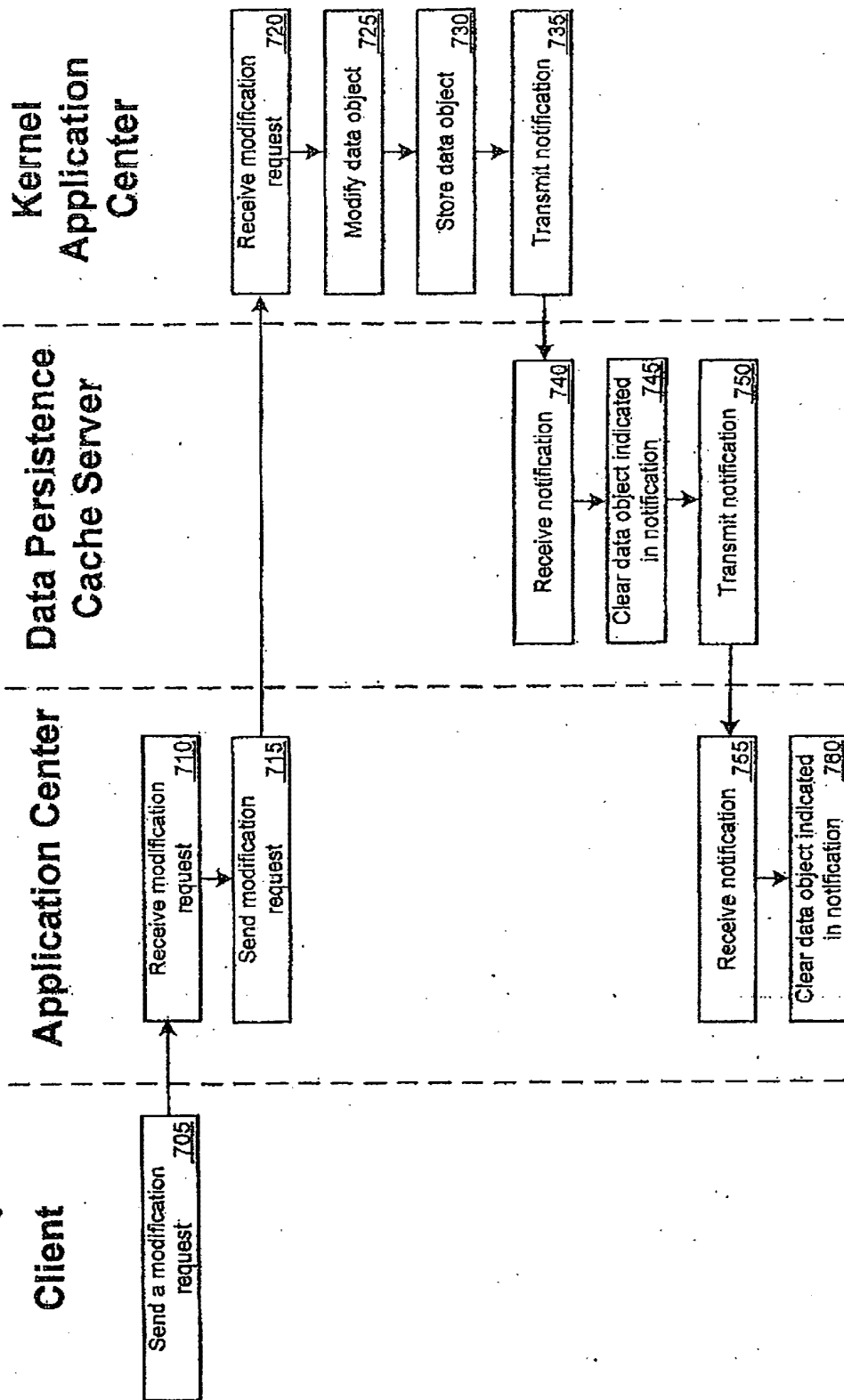


FIG. 5B



600

FIG. 6



700

FIG. 7

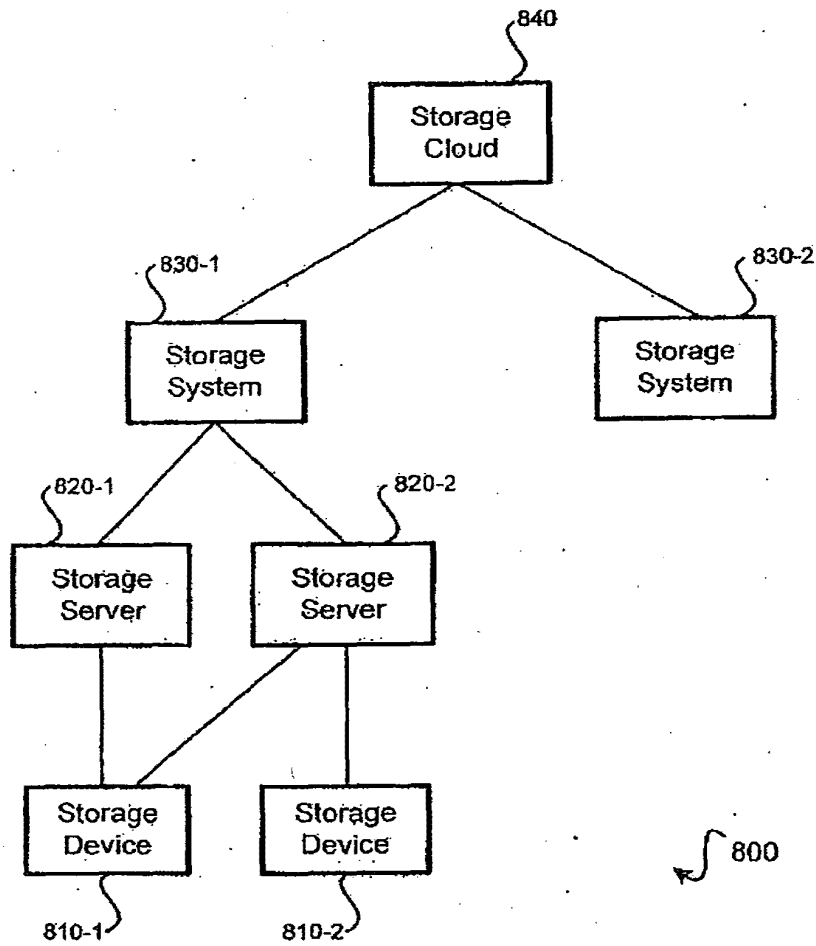


FIG. 8

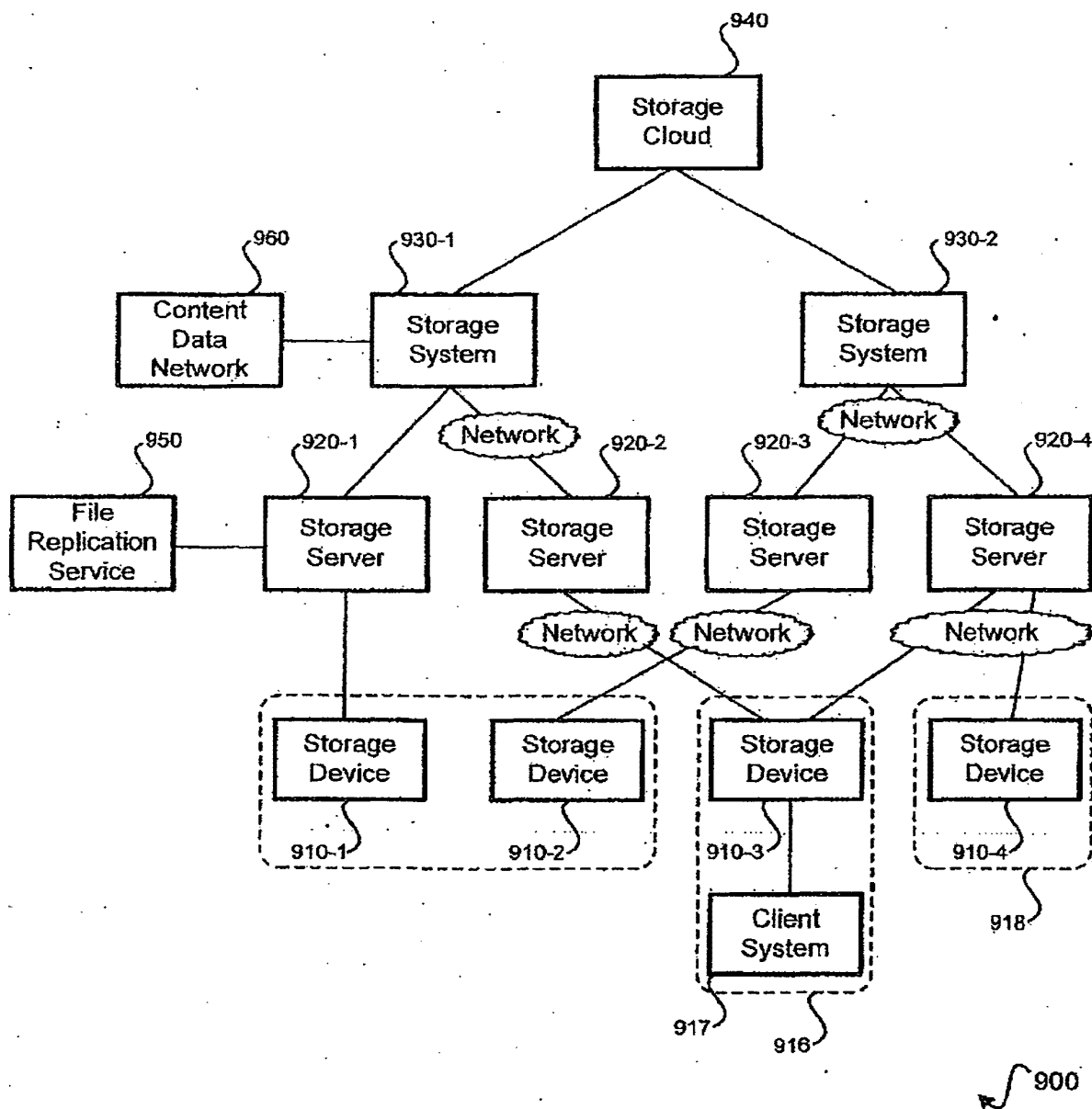


FIG. 9

2010202782 01 Jul 2010

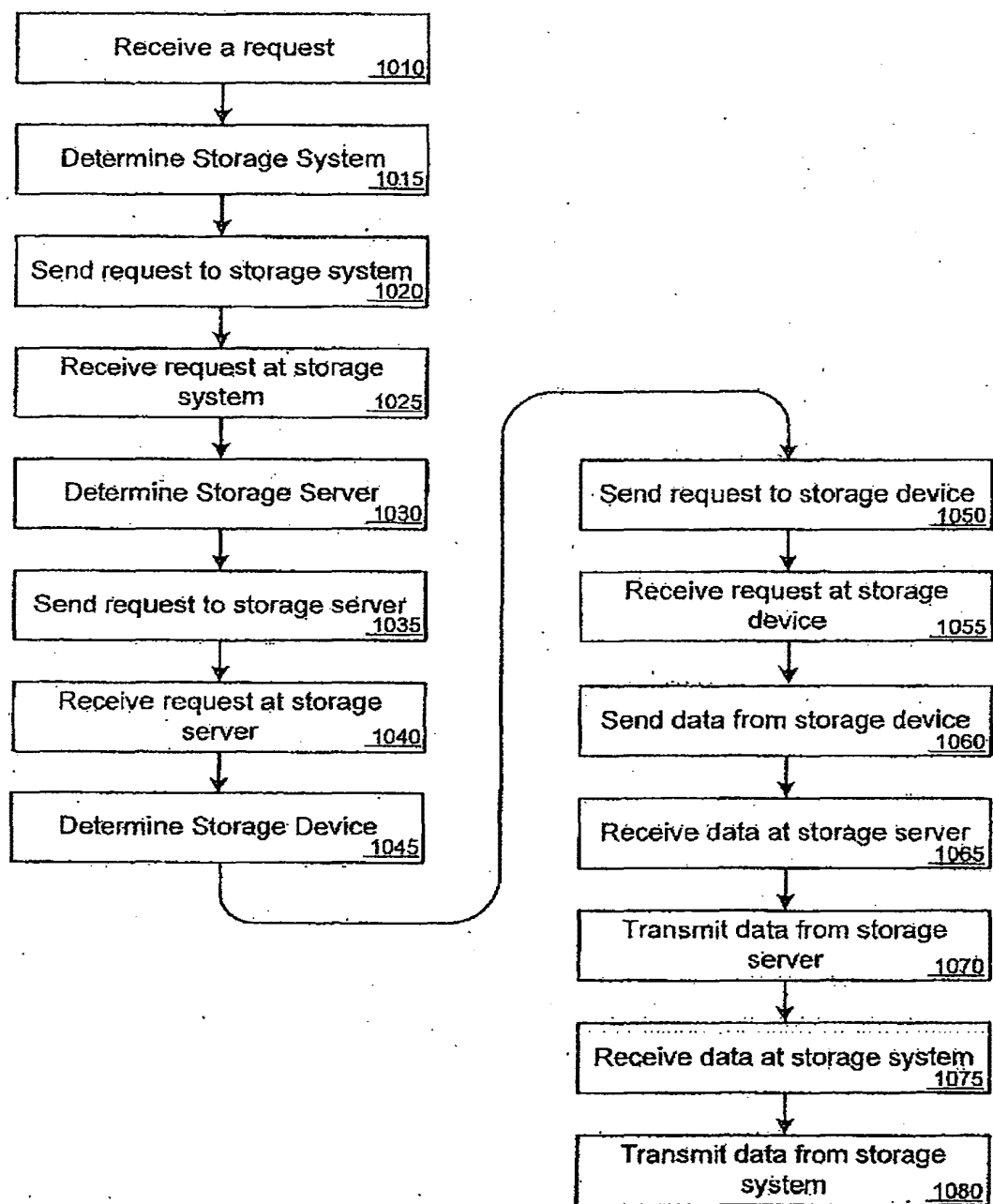


FIG. 10

1000

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**TRANSMITTAL
FORM**

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

1

Application Number

13/017,866

Filing Date

January 31, 2011

First Named Inventor

Albert J. McGowan et al.

Art Unit

2447

Examiner Name

HWANG, JOON H

Attorney Docket Number

92853-798163 (000710US)

ENCLOSURES (Check all that apply)

Fee Transmittal Form



Fee Attached



Amendment/Reply



After Final



Affidavits/declaration(s)



Extension of Time Request



Express Abandonment Request



Information Disclosure Statement



Certified Copy of Priority Document(s)

Reply to Missing Parts/
Incomplete ApplicationReply to Missing Parts
under 37 CFR 1.52 or 1.53

Drawing(s)



Licensing-related Papers



Petition

Petition to Convert to a
Provisional Application

Power of Attorney, Revocation



Change of Correspondence Address



Terminal Disclaimer



Request for Refund



CD, Number of CD(s) _____

☐ Landscape Table on CD

After Allowance Communication to TC

Appeal Communication to Board
of Appeals and InterferencesAppeal Communication to TC
(Appeal Notice, Brief, Reply Brief)

Proprietary Information



Status Letter

Other Enclosure(s) (please identify
below):

Certified Priority Document:

AU 2010202782, Filed 1 July 2010

PostCard

Remarks

The Commissioner is authorized to charge any additional fees to Deposit Account 20-1430

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name

Kilpatrick Townsend & Stockton LLP

Signature

Printed name

Karam J. Saab

Date

December 6, 2011

Reg. No.

64,190

CERTIFICATE OF TRANSMISSION/MAILINGI hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below: PM 548365660

Signature

Typed or printed name

Corina K. Aschenbrenner

Date

December 6, 2011

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	92853-798163

CONFIRMATION NO. 7494

PUBLICATION NOTICE



OC000000052361929

20350
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Title:CLOUD DATA PERSISTENCE ENGINE

Publication No.US-2012-0030313-A1

Publication Date:02/02/2012

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <http://pair.uspto.gov/>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494
20350 7590 02/08/2012 KILPATRICK TOWNSEND & STOCKTON LLP TWO EMBARCADERO CENTER EIGHTH FLOOR SAN FRANCISCO, CA 94111-3834			EXAMINER TIV, BACKHEAN	
			ART UNIT 2451	PAPER NUMBER
			NOTIFICATION DATE 02/08/2012	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

Docket@kilpatricktownsend.com
ipefiling@kilpatricktownsend.com
jlhice@kilpatrick.foundationip.com

Office Action Summary	Application No.	Applicant(s)	
	13/017,866	MCGOWAN ET AL.	
	Examiner	Art Unit	
	BACKHEAN TIV	2451	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 1/31/11.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 5) ☒ Claim(s) 1-20 is/are pending in the application.
- 5a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) ____ is/are allowed.
- 7) ☒ Claim(s) 1-20 is/are rejected.
- 8) ☐ Claim(s) ____ is/are objected to.
- 9) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 1/31/11 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 12) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 13) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. ____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) ☒ Notice of References Cited (PTO-892) | 4) ☐ Interview Summary (PTO-413) |
| 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. ____. |
| 3) ☒ Information Disclosure Statement(s) (PTO/SB/08) | 5) ☐ Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>8/15/11</u> . | 6) ☐ Other: ____. |

Detailed Action

Claims 1-20 are pending in this application.

Drawings

The Drawings filed on 1/31/11 are acceptable.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 8/15/11 has been considered.

Priority

Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Publication 2005/0044242 issued to Stevens et al.(Stevens) in view of US Publication 2012/0005276 issued to Guo et al.(Guo).

As per claims 1, 5, 14 Steven teaches a cloud data persistence system for distribution of data, the cloud data persistence system comprising: a first server which stores a first plurality of data objects and is communicatively coupled with a cache

server(Fig.1, para.0028-0031; web server), wherein: the first server is located at a first location; each of the first plurality of data objects comprise a link to one of a plurality of media assets(Fig.1; web server); and the plurality of media assets are stored at a second location separate from the first location(Fig.1; para.0028-0031, upstream proxy stores objects separate from web server); a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects, wherein: the second server is one of the plurality of cache servers; and the second server is communicatively coupled with a first servers(Fig.1; para.0028-0031, upstream proxy request objects from the web server); and a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fourth plurality of data objects, wherein: the fourth server and the fourth server is configured to receive a request from a first client for the data objectFig.1, para.0028-0031.

Steven however does not teach application servers or a plurality of proxy cache servers or a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects, wherein: the third server is one of the plurality of cache servers; and the third server is communicatively coupled with a second plurality of application servers; a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein: the fifth server is one of the first plurality of

application servers; and the fifth server is configured to receive a request from a second client for the data object; a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects, wherein: the sixth server is one of the second plurality of application servers; and the sixth server is configured to receive a request from a third client for the data object; and a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects, wherein: the seventh server is one of the second plurality of application servers; and the seventh server is configured to receive a request from a fourth client for the data object.

Guo explicitly teaches the use of application proxy deemed to be application server(Fig.1).

Therefore it would have been obvious to one ordinary skill in the art at the time of the invention to modify the teachings of Steven to include the use of application server as taught by Guo in order to quickly update applications through the use of the application proxy, which reduces network traffic(Guo, para.0001).

One ordinary skill in the art would have been motivated to combine the teachings of Steven and Guo to quickly update applications through the use of the application proxy, which reduces network traffic(Guo, para.0001).

Steven in view of Guo does not explicitly teach a plurality of proxy cache servers or a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality

of data objects, wherein: the third server is one of the plurality of cache servers; and the third server is communicatively coupled with a second plurality of application servers; a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein: the fifth server is one of the first plurality of application servers; and the fifth server is configured to receive a request from a second client for the data object; a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects, wherein: the sixth server is one of the second plurality of application servers; and the sixth server is configured to receive a request from a third client for the data object; and a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects, wherein: the seventh server is one of the second plurality of application servers; and the seventh server is configured to receive a request from a fourth client for the data object.

However, this is merely duplication of parts, *In re Harza*, 274 F.2d 669, 124 USPQ 378 (CCPA 1960). Therefore adding multiple cache and application servers to perform the method as taught by Steven and Guo would have been obvious to one ordinary skill in the art.

As per claims 2, the cloud data persistence system of claim 1, wherein the second location comprises a content delivery network(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 3, the cloud data persistence system of claim 1, wherein: data objects in the second plurality of data objects are at least partially determined based on when requests for those data objects were last received at the second server; data objects in the third plurality of data objects are at least partially determined based on when requests for those data objects were last received at the third server; data objects in the fourth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fourth server; data objects in the fifth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fifth server; data objects in the sixth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the sixth server; and data objects in the seventh plurality of data objects are at least partially determined based on when requests for those data objects were last received at the seventh server(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 4, the cloud data persistence system of claim 1, wherein the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth plurality of data objects, and the seventh plurality of data objects(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 6, the cloud data persistence system of claim 5 wherein the first plurality of data objects comprises all data objects present in the second plurality of data

objects and the third plurality of data objects(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 7, the cloud data persistence system of claim 5, wherein the third server stores a fourth plurality of data objects and is configured to request the data object from the first server if the data object is not present in the fourth plurality of data objects, wherein: the third server and the second server are data persistence cache servers; and the third server is communicatively coupled with a sixth and seventh server(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 8, the cloud data persistence system of claim 5, wherein the fifth server stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein: the fifth server and the fourth servers are application servers; and the fifth server is configured to receive requests from a second client for the data object; and the fifth server is geographically separated from the fourth server(Steven, Fig.1; para.0028-0031, Guo, para.0001). Official Notice is taken; The art of having of geographically separated servers is well known in the art(see US Patent 7,373,394 issued to Li, col.1, lines 44-55 for support). Therefore it's obvious to one ordinary skill in the art to modify the teachings to have servers geographically separate in order to provide servers in different locations for user's closest to the location to obtain information. By providing geographic based server, provides quicker responses to request.

As per claims 9, the cloud data persistence system of claim 5 wherein the request from the first client for the data object comprises a universal resource indicator(Steven, Fig.1; para.0005).

As per claims 10, the cloud data persistence system of claim 5 wherein each of the first plurality of data objects comprise a universal resource locator and a title linked to a media asset of the plurality of media assets(Steven, Fig.1; para.0005).

As per claims 11, the cloud data persistence system of claim 5, wherein data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on a frequency of requests for those data objects received from clients(Steven, Fig.1; para.0028-0031, Guo, para.0001). Official Notice is taken; caching frequency requested objects is well known in the art. Therefore it's obvious to one ordinary skill in the art to modify the teaching to include caching objects that are frequently requested in order serve objects quickly as the cache server doesn't need to get the object from the origin/web server.

As per claims 12, the cloud data persistence system of claim 5, wherein data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 13, the cloud data persistence system of claim 5, wherein the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, and the fifth

plurality of data objects(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 15, the method of claim 14 further comprising, after receiving the first data object from the origin server, storing, by the cache server, the first data object(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 16, the method of claim 14 further comprising, after receiving the first data object from the cache server, storing, by the application center, the first data object(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 17, the method of claim 14 wherein the cache server is located at a third location(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 18, the method of claim 14 wherein the first request for the data object comprises a universal resource indicator(Steven, Fig.1; para.0005).

As per claims 19, the method of claim 14, wherein data objects in the first plurality of data objects and the second plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

As per claims 20, the method of claim 14, wherein the third plurality of data objects comprises all data objects in the first and the second pluralities of data

objects(Steven, Fig.1; para.0028-0031, Guo, para.0001). Motivation to combine set forth in claim 1.

Conclusion

Examiner's Note: Examiner has cited particular columns and line numbers in the references as applied to the claims above for the convenience of the applicant.

Although the specified citations are representative of the teachings of the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in its entirety as potentially teaching of all or part of the claimed invention.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BACKHEAN TIV whose telephone number is (571)272-5654. The examiner can normally be reached on M-T 7-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (571) 272-3964. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Backhean Tiv/

Primary Examiner, Art Unit 2451

Notice of References Cited	Application/Control No. 13/017,866		Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.	
	Examiner BACKHEAN TIV		Art Unit 2451	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,852,717 A	12-1998	Bhide et al.	709/203
*	B	US-6,405,240 B1	06-2002	Tsubone et al.	709/203
*	C	US-2008/0294748 A1	11-2008	Brown et al.	709/219
*	D	US-7,516,194 B1	04-2009	Lamkins et al.	709/218
*	E	US-7,613,815 B1	11-2009	Prakash et al.	709/230
*	F	US-7,716,282 B2	05-2010	Takeshima et al.	709/203
*	G	US-7,809,840 B2	10-2010	Jing et al.	709/227
*	H	US-2011/0016197 A1	01-2011	Shiimori et al.	709/219
*	I	US-7,953,820 B2	05-2011	Stevens et al.	709/219
*	J	US-2012/0005276 A1	01-2012	Guo et al.	709/206
*	K	US-2012/0030317 A1	02-2012	Smyth et al.	709/219
*	L	US-2005/0044242	02-2005	Stevens et al.	709/228
*	M	US-7,373,394	05-2008	Li et al.	709/219

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	4	((("20030061272") or ("7373394")).PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; DERWENT; IBM_TDB	OR	OFF	2012/01/26: 09:14
S2	299	proxy near4 cache with (content or orgin) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:12
S3	34	proxy near4 cache with (content or orgin) near4 server same application near3 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:13
S4	118	(user or client) near4 (send\$4 or transmit\$4) near4 request\$4 near5 (object or URL) with application near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:26
S5	5	(user or client) near4 (send\$4 or transmit\$4) near4 request\$4 near5 (object or URL) with application near4 server same proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:26
S6	130	proxy near3 request\$4 near4 (another or different) near4 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:27
S7	579	proxy near3 request\$4 near4 (another or different or second) near4 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:27
S8	2	proxy near3 request\$4 near4 (another or different or second) near4 proxy near4 level and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02: 10:27
S9	44142	(709/219-225).OCLS.	US-PGPUB; USPAT;	OR	OFF	2012/02/02: 10:28

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S10	106	S9 and S7	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:28
S11	2	("20050044242").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:41
S12	4432	application near3 server with (proxy or cache) near3 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:02
S13	1464	application near3 server with (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:02
S14	231	S13 and S9	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S15	1	application near3 server with downstream near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S16	1	application adj2 server with downstream near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S17	708	application adj2 server near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S18	123	S17 and S9	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S19	25	(application adj2 server near3 (proxy or cache) near3 server with (content	US-PGPUB; USPAT;	OR	OFF	2012/02/02 11:08

		or web) near4 server).ab,ti. and (@ad<="20100701" or @rlad<="20100701")	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S20	3095	application adj2 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:13
S21	62	application adj2 proxy with user near3 request\$4 and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:14
S22	10	(application adj2 proxy with user near3 request\$4).ab,ti. and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:15

2/ 2/ 2012 1:47:48 PM
C:\ Users\ btiv\ Documents\ EAST\ Workspaces\ 13017866_cloud_cache.wsp

Substitute for form 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Use as many sheets as necessary)</i>				Complete if Known	
				Application Number	13/017,866
				Filing Date	January 31, 2011
				First Named Inventor	Albert J. McGowan
				Art Unit	2447
				Examiner Name	Not Yet Assigned
Sheet	1	of	1	Attorney Docket Number	92853-798163 (000710US)

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number Kind Code ² (if known)			
	A1	US 2003/0061272 A1	03-27-2003	Krishnamurthy, et al.	
	A2	US 7,373,394 B1	05-13-2008	Li, et al.	

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No. ¹	Foreign Patent Document			Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴	Kind Code ⁵ (if known)				

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials *	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ⁶
	C1	CHANG, F., et al., "Bigtable: A Distributed Storage System for Structured Data," [online], Google Inc., November 2006, [Retrieved from the Internet on July 28, 2011], Retrieved from: http://labs.google.com/papers/bigtable.html , 14 pages.	☐

Examiner Signature	/Backhean Tiv/ (02/02/2012)	Date Considered	02/02/2012
-----------------------	-----------------------------	--------------------	------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kind Codes of U.S. Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

63637149 v1

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /BT



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7494

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.		
13/017,866	01/31/2011	709	2451	92853-798163		
RULE						
APPLICANTS Albert J. McGowan, Phoenix, AZ; Richard L. Carls, Scottsdale, AZ; ** CONTINUING DATA ***** ** FOREIGN APPLICATIONS ***** AUSTRALIA 2010202782 07/01/2010 ** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** ** SMALL ENTITY ** 02/10/2011						
Foreign Priority claimed ☒ Yes ☐ No 35 USC 119(a-d) conditions met ☒ Yes ☐ No Verified and /BACKHEAN TIV/ Acknowledged Examiner's Signature		☐ Met after Allowance Initials	STATE OR COUNTRY AZ	SHEETS DRAWINGS 14	TOTAL CLAIMS 20	INDEPENDENT CLAIMS 3
ADDRESS KILPATRICK TOWNSEND & STOCKTON LLP TWO EMBARCADERO CENTER EIGHTH FLOOR SAN FRANCISCO, CA 94111-3834 UNITED STATES						
TITLE CLOUD DATA PERSISTENCE ENGINE						
FILING FEE RECEIVED 527	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:			☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit		

<i>Index of Claims</i> 	Application/Control No. 13017866	Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.
	Examiner BACKHEAN TIV	Art Unit 2451

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA		☐ T.D.		☐ R.1.47					
CLAIM		DATE									
Final	Original	02/02/2012									
	1	✓									
	2	✓									
	3	✓									
	4	✓									
	5	✓									
	6	✓									
	7	✓									
	8	✓									
	9	✓									
	10	✓									
	11	✓									
	12	✓									
	13	✓									
	14	✓									
	15	✓									
	16	✓									
	17	✓									
	18	✓									
	19	✓									
	20	✓									

Search Notes 	Application/Control No. 13017866	Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.
	Examiner BACKHEAN TIV	Art Unit 2451

SEARCHED			
Class	Subclass	Date	Examiner
709	219-225(limited search)	2/2/12	BT

SEARCH NOTES		
Search Notes	Date	Examiner
Inventor's Name Search PALM	2/2/12	BT
EAST txt Search	2/2/12	BT

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

	/BACKHEAN TIV/ Primary Examiner.Art Unit 2451
--	--



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494
20350 7590 02/28/2012 KILPATRICK TOWNSEND & STOCKTON LLP TWO EMBARCADERO CENTER EIGHTH FLOOR SAN FRANCISCO, CA 94111-3834			EXAMINER TIV, BACKHEAN	
			ART UNIT 2451	PAPER NUMBER
			NOTIFICATION DATE 02/28/2012	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

Docket@kilpatricktownsend.com
ipefiling@kilpatricktownsend.com
jlhice@kilpatrick.foundationip.com

<i>Applicant-Initiated Interview Summary</i>	Application No.	Applicant(s)	
	13/017,866	MCGOWAN ET AL.	
	Examiner	Art Unit	
	BACKHEAN TIV	2451	

All participants (applicant, applicant's representative, PTO personnel):

(1) BACKHEAN TIV. (3) Karam J. Saab(64190).

(2) _____. (4) Tom Franklin(43616).

Date of Interview: 22 February 2012.

Type: ☒ Telephonic ☐ Video Conference
 ☒ Personal [copy given to: ☐ applicant ☒ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☒ No.
 If Yes, brief description: _____.

Issues Discussed ☐101 ☐112 ☐102 ☐103 ☒Others
 (For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: All.

Identification of prior art discussed: prior art of record.

Substance of Interview
 (For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Discussed the invention as it relates, the data object being meta data and not content caching, Fig.5A, para.0049. The applicant distinguished the Stevens from the claimed invention. Further Search and/or consideration is necessary, the examiner will telephone the applicant if there are any suggestions to advance the prosecution of the application..

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☐ Attachment

/Backhean Tiv/ Primary Examiner, Art Unit 2451	
---	--

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record

A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews

Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.

All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

- 1) A brief description of the nature of any exhibit shown or any demonstration conducted,
- 2) an identification of the claims discussed,
- 3) an identification of the specific prior art discussed,
- 4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
- 5) a brief identification of the general thrust of the principal arguments presented to the examiner,
(The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
- 6) a general indication of any other pertinent matters discussed, and
- 7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

I hereby certify that this correspondence is being filed via
EFS-Web with the United States Patent and Trademark Office
on May 18, 2012

PATENT
Attorney Docket No.: 92853-798163
(028406-000700US)

KILPATRICK TOWNSEND & STOCKTON LLP

By: /Stephanie Klepp/
Stephanie Klepp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Albert John McGowan et al

Application No.: 13/017,866

Filed: January 31, 2011

For: CLOUD DATA PERSISTENCE
ENGINE

Customer No. 20350

Confirmation No. 7494

Examiner: Backhean Tiv

Technology Center/Art Unit: 2451

AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

In response to the Office Action mailed February 8, 2012, please enter the following
amendments and remarks:

A **listing of Claims** begins on page 2 of this paper.

Remarks begin on page 9 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Original) A cloud data persistence system for distribution of data, the cloud data persistence system comprising:
 - a first server which stores a first plurality of data objects and is communicatively coupled with a plurality of cache servers, wherein:
 - the first server is located at a first location;
 - each of the first plurality of data objects comprise a link to one of a plurality of media assets; and
 - the plurality of media assets are stored at a second location separate from the first location;
 - a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects, wherein:
 - the second server is one of the plurality of cache servers; and
 - the second server is communicatively coupled with a first plurality of application servers;
 - a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects, wherein:
 - the third server is one of the plurality of cache servers; and
 - the third server is communicatively coupled with a second plurality of application servers; and
 - a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fourth plurality of data objects, wherein:
 - the fourth server is one of the first plurality of application server; and

the fourth server is configured to receive a request from a first client for the data object;

a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein:

the fifth server is one of the first plurality of application servers; and

the fifth server is configured to receive a request from a second client for the data object;

a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects, wherein:

the sixth server is one of the second plurality of application servers; and

the sixth server is configured to receive a request from a third client for the data object; and

a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects, wherein:

the seventh server is one of the second plurality of application servers; and

the seventh server is configured to receive a request from a fourth client for the data object.

2. (Original) The cloud data persistence system of claim 1, wherein the second location comprises a content delivery network.

3. (Original) The cloud data persistence system of claim 1, wherein:

data objects in the second plurality of data objects are at least partially determined based on when requests for those data objects were last received at the second server;

data objects in the third plurality of data objects are at least partially determined based on when requests for those data objects were last received at the third server;

data objects in the fourth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fourth server;

data objects in the fifth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the fifth server;

data objects in the sixth plurality of data objects are at least partially determined based on when requests for those data objects were last received at the sixth server; and

data objects in the seventh plurality of data objects are at least partially determined based on when requests for those data objects were last received at the seventh server.

4. (Original) The cloud data persistence system of claim 1, wherein the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, the fifth plurality of data objects, the sixth plurality of data objects, and the seventh plurality of data objects.

5. (Original) A cloud data persistence system for distribution of data, the cloud data persistence system comprising:

a first server which stores a first plurality of data objects and is communicatively coupled with a second server and a third server, wherein:

the first server is located at a first location;

each of the first plurality of data objects comprises a link to a media asset of a plurality of media assets;

the plurality of media assets is located at a second location; and

the second location is distinct from the first location;

the second server stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects, wherein the second server is communicatively coupled with a fourth server and fifth server; and

the fourth server stores a third plurality of data objects and is configured to request the data object from the second server if the data object is not present in the third

plurality of data objects, wherein the fourth server is configured to receive a request from a first client for the data object.

6. (Original) The cloud data persistence system of claim 5 wherein the first plurality of data objects comprises all data objects present in the second plurality of data objects and the third plurality of data objects.

7. (Original) The cloud data persistence system of claim 5, wherein the third server stores a fourth plurality of data objects and is configured to request the data object from the first server if the data object is not present in the fourth plurality of data objects, wherein:

the third server and the second server are data persistence cache servers;

and

the third server is communicatively coupled with a sixth and seventh server.

8. (Original) The cloud data persistence system of claim 5, wherein the fifth server stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein:

the fifth server and the fourth servers are application servers; and

the fifth server is configured to receive requests from a second client for the data object; and

the fifth server is geographically separated from the fourth server.

9. (Original) The cloud data persistence system of claim 5, wherein the request from the first client for the data object comprises a universal resource indicator.

10. (Original) The cloud data persistence system of claim 5 wherein each of the first plurality of data objects comprise a universal resource locator and a title linked to a media asset of the plurality of media assets.

11. (Original) The cloud data persistence system of claim 5, wherein data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on a frequency of requests for those data objects received from clients.

12. (Original) The cloud data persistence system of claim 5, wherein data objects in the second plurality of data objects and the third plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients.

13. (Original) The cloud data persistence system of claim 5, wherein the first plurality of data objects comprises all data objects in the second plurality of data objects, the third plurality of data objects, the fourth plurality of data objects, and the fifth plurality of data objects.

14. (Original) A method for retrieving data objects using a cloud data persistence system configured for distribution of data, the method comprising:
receiving, by an application center, a first request from a client for a data object,
wherein:

the application center is one of a plurality of application centers that
communicate with a plurality of clients;

the application center is located at a first location;

the data object comprises a link to a media asset; and

the media asset is stored at a second location separate from the first
location;

determining, by the application center, the data object is not stored among a first
plurality of data objects stored at the application center;

transmitting, by the application center, a second request for the data object to a
first cache server, wherein:

the first cache server is communicatively coupled with the plurality of application centers; and

the first cache server is one of a plurality of cache servers;

receiving, by the cache server, the second request for the data object;

determining, by the cache server, the data object is not stored among a second plurality of data objects stored at the cache server;

transmitting, by the cache server, a third request for the data object to an origin server, wherein the origin server is communicatively coupled with the plurality of cache servers;

receiving, by the origin server, the third request for the data object;

locating, by the origin server, the data object among a third plurality of data objects;

transmitting, by the origin server, the data object linked to the third request to the cache server;

receiving, by the cache server, the data object linked to the third request;

transmitting, by the cache server, the data object linked to the second request;

receiving, by the application center, the data object linked to the second request;

and

transmitting, by the application center, the data object linked to the first request to the client.

15. (Original) The method of claim 14 further comprising, after receiving the first data object from the origin server, storing, by the cache server, the first data object.

16. (Original) The method of claim 14 further comprising, after receiving the first data object from the cache server, storing, by the application center, the first data object.

17. (Original) The method of claim 14 wherein the cache server is located at a third location.

18. (Original) The method of claim 14 wherein the first request for the data object comprises a universal resource indicator.

19. (Original) The method of claim 14, wherein data objects in the first plurality of data objects and the second plurality of data objects are at least partially determined based on when requests for those data objects were last received from clients.

20. (Original) The method of claim 14, wherein the third plurality of data objects comprises all data objects in the first and the second pluralities of data objects.

REMARKS

Prior to the entry of this Amendment, the Application included claims 1-20. No claims have been amended, canceled, or added. Hence, after entry of this Amendment, claims 1-20 stand pending for examination.

Claims 1-20 stand rejected under 35 USC § 103(a) as being allegedly unpatentable over U.S. Publication No. 2005/0044242 to Stevens et al. (“Stevens”), in view of U.S. Publication No. 2012/0005276 to Guo et al. (“Guo”).

Examiner Interview

An interview was conducted between Examiner Tiv and Applicants’ Representatives on February 22, 2012. During the interview, Examiner Tiv and Applicants’ Representatives discussed the terms “data object” and “media asset” in reference to the claims of the Application. Applicants and Applicants’ Representatives thank Examiner Tiv for a helpful and productive interview.

§103 Rejections

Independent claims 1, 5, and 14 stand rejected as allegedly obvious over Stevens and Guo. The Applicants respectfully traverse the rejections for at least the reasons detailed herein.

Claim 1 requires that the first server stores a first plurality of data objects at a first location. These *data objects* comprise links to *media assets*, which are stored at a different location than the first location. The remaining independent claims contain similar recitations. A data object contains information about a media asset. As an example, a data object may include fields containing a title, keywords, a date of creation of the media asset, the duration of the media asset, the file formats available for the media asset, and/or which clients are permitted to access the media asset. Each data object that details a media asset is maintained at a location separate from the corresponding media asset. Such an arrangement is not taught, suggested, or otherwise proved obvious by the cited references of Stevens and Guo, whether considered individually or in combination.

Stevens, the primary reference, is generally directed to “supporting retrieval of web content over a meshed communication network.” *Stevens*, Abstract. Stevens discloses the use of a universal resource locator (URL) to retrieve a resource, such as an HTML document, graphic image, or video clip. *Id.*, at ¶28. Stevens describes the use of downstream and upstream proxies, which can be used for pre-fetching. *Id.*, at ¶29. While Stevens is directed to parsing and pre-fetching web content, such as web pages, the independent claims are directed to servers storing, requesting, and retrieving *data objects, which describe a media asset* stored elsewhere. As such, it is not the media assets themselves being accessed and retrieved, but rather the *data objects* which contain information *about* the media assets.

Guo does not cure this deficiency. Guo is generally directed to “reducing application update traffic in cellular networks.” *Guo*, ¶1. The Office Action cites to Guo for allegedly proving obvious the use of an application proxy arrangement. *Office Action*, p. 4. Guo, like Stevens, does not prove obvious an arrangement between data objects and media assets as presented in the independent claims. The independent claims are directed to servers storing, requesting, and retrieving data objects, which each contain information about a media asset stored elsewhere. Again, it is not the media assets themselves being accessed and retrieved, but rather the *data objects* which contain information *about* the media assets. Such recitations are not present in the cited references.

For at least the above reasons, claims 1, 5, and 14 are not proved obvious by the cited references of Stevens and Guo. The remaining independent claims are non-obvious for at least some of the same reasons. Claims 2-4, 6-13, and 15-20 depend, either directly or indirectly, from claims 1, 5, and 14. These claims are non-obvious at least by virtue of their dependence on non-obvious base claims. Withdrawal of the §103 rejections and issuance of a notice of allowance for claims 1-20 is respectfully requested.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested. Further, the Commissioner is hereby authorized to charge any additional fees or credit any overpayment in connection with this paper to Deposit Account No. 20-1430. If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-517-4000.

Respectfully submitted,

Karam J. Saab
Registration No. 64,190

KILPATRICK TOWNSEND & STOCKTON LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 415-576-0200
Fax: 415-576-0300
K4S/s5k
63979459v.1

**PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)
FY 2009***(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*

Docket Number (Optional)

92853-798163 (028406-000700US)

Application Number 13/017,866

Filed January 31, 2011

For CLOUD DATA PERSISTENCE ENGINE

Art Unit 2451

Examiner Joon H. Hwang

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

	<u>Fee</u>	<u>Small Entity Fee</u>	
☒ One month (37 CFR 1.17(a)(1))	\$150	\$75	<u>\$150</u>
☐ Two months (37 CFR 1.17(a)(2))	\$560	\$280	\$ _____
☐ Three months (37 CFR 1.17(a)(3))	\$1270	\$635	\$ _____
☐ Four months (37 CFR 1.17(a)(4))	\$1980	\$990	\$ _____
☐ Five months (37 CFR 1.17(a)(5))	\$2690	\$1345	\$ _____

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☒ Payment by credit card.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to

Deposit Account Number 20-1430.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

☒ attorney or agent of record. Registration Number 64,190

☐ attorney or agent under 37 CFR 1.34.

Registration number if acting under 37 CFR 1.34. _____.

Signature

Karam J. Saab

Typed or printed name

May 18, 2012

Date

303-571-4000

Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Patent Application Fee Transmittal				
Application Number:		13017866		
Filing Date:		31-Jan-2011		
Title of Invention:		CLOUD DATA PERSISTENCE ENGINE		
First Named Inventor/Applicant Name:		Albert J. McGowan		
Filer:		Karam J. Saab/Stephanie Klepp		
Attorney Docket Number:		92853-798163		
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 1 month with \$0 paid	1251	1	150	150

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				150

Electronic Acknowledgement Receipt

EFS ID:	12815064
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab/Stephanie Klepp
Filer Authorized By:	Karam J. Saab
Attorney Docket Number:	92853-798163
Receipt Date:	18-MAY-2012
Filing Date:	31-JAN-2011
Time Stamp:	15:21:55
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 150
RAM confirmation Number	1879
Deposit Account	201430
Authorized User	SAAB,KARAM

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		028406_000700US_AMD_18_May_2012.pdf	444971	yes	11
			3ddb42113960c1253a2da612d98d7a6e6a6cf713		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Amendment/Req. Reconsideration-After Non-Final Reject		1	1	
	Claims		2	8	
	Applicant Arguments/Remarks Made in an Amendment		9	11	
Warnings:					
Information:					
2	Extension of Time	028406_000700US_EXT_18_May_2012.pdf	67889	no	1
			bf32b733e7cf9ba63a5ad5b0e3438c4fd0407e43		
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	30378	no	2
			15b02155bd45042dbe44de86b0168c090ce8717c		
Warnings:					
Information:					
Total Files Size (in bytes):			543238		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 13/017,866		Filing Date 01/31/2011		☐ To be Mailed		
APPLICATION AS FILED – PART I											
(Column 1)			(Column 2)			SMALL ENTITY ☒ OR		OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA	RATE (\$)	FEE (\$)		RATE (\$)	FEE (\$)				
☒ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	82		N/A					
☐ SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A			N/A					
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A					
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X \$	=	OR	X \$	=				
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X \$	=		X \$	=				
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))			If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).								
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))											
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL 82			TOTAL					
APPLICATION AS AMENDED – PART II											
(Column 1)			(Column 2)			(Column 3)		SMALL ENTITY OR		OTHER THAN SMALL ENTITY	
AMENDMENT	05/18/2012	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)		RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	* 20	Minus	** 20	= 0	X \$30 =	0	OR	X \$	=	
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X \$125 =	0	OR	X \$	=	
	☐ Application Size Fee (37 CFR 1.16(s))										
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR			
					TOTAL ADD'L FEE	0	OR	TOTAL ADD'L FEE			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)		RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	X \$	=	OR	X \$	=	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X \$	=	OR	X \$	=	
	☐ Application Size Fee (37 CFR 1.16(s))										
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR			
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.											

 Legal Instrument Examiner:
 /GWENDOLYN MYERS/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

20350 7590 05/30/2012
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER

TIV, BACKHEAN

ART UNIT

PAPER NUMBER

2451

DATE MAILED: 05/30/2012

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494

TITLE OF INVENTION: CLOUD DATA PERSISTENCE ENGINE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	\$870	\$300	\$0	\$1170	08/30/2012

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or **Fax** (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

20350 7590 05/30/2012
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

	(Depositor's name)
	(Signature)
	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494

TITLE OF INVENTION: CLOUD DATA PERSISTENCE ENGINE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	\$870	\$300	\$0	\$1170	08/30/2012

EXAMINER	ART UNIT	CLASS-SUBCLASS
TIV, BACKHEAN	2451	709-219000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____

2 _____

3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

☐ Issue Fee

☐ Publication Fee (No small entity discount permitted)

☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): (**Please first reapply any previously paid issue fee shown above**)

☐ A check is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____

Date _____

Typed or printed name _____

Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494

20350 7590 05/30/2012
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER

TTV, BACKHEAN

ART UNIT	PAPER NUMBER
----------	--------------

2451

DATE MAILED: 05/30/2012

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 0 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 0 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)	
	13/017,866	MCGOWAN ET AL.	
	Examiner	Art Unit	
	BACKHEAN TIV	2451	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 5/18/12.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are 1-20.
4. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☒ All b) ☐ Some* c) ☐ None of the:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. ____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: ____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
 6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date ____.
 - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).**
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|---|
| 1. ☒ Notice of References Cited (PTO-892) | 5. ☐ Notice of Informal Patent Application |
| 2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) | 6. ☐ Interview Summary (PTO-413),
Paper No./Mail Date ____. |
| 3. ☐ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date ____ | 7. ☐ Examiner's Amendment/Comment |
| 4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material | 8. ☒ Examiner's Statement of Reasons for Allowance |
| | 9. ☐ Other ____. |

/Backhean Tiv/
Primary Examiner, Art Unit 2451

Detailed Action

Claims 1-20 are pending in this application. This is a response to the Remarks filed on 5/18/12.

REASONS FOR ALLOWANCE

The following is an examiner's statement of reasons for allowance:

The prior art singly or in combination does not teach, as per claim 1, cloud data persistence system for distribution of data, the cloud data persistence system comprising: a first server which stores a first plurality of data objects and is communicatively coupled with a plurality of cache servers, wherein: the first server is located at a first location; each of the first plurality of data objects comprise a link to one of a plurality of media assets; and the plurality of media assets are stored at a second location separate from the first location; a second server that stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects, wherein: the second server is one of the plurality of cache servers; and the second server is communicatively coupled with a first plurality of application servers; a third server that stores a third plurality of data objects and is configured to request the data object from the first server if the data object is not present in the third plurality of data objects, wherein: the third server is one of the plurality of cache servers; and the third server is communicatively coupled with a second plurality of application servers; and a fourth server that stores a fourth plurality of data objects and is configured to request the data object from the second server if the data

object is not present in the fourth plurality of data objects, wherein: the fourth server is one of the first plurality of application server; and the fourth server is configured to receive a request from a first client for the data object; a fifth server that stores a fifth plurality of data objects and is configured to request the data object from the second server if the data object is not present in the fifth plurality of data objects, wherein: the fifth server is one of the first plurality of application servers; and the fifth server is configured to receive a request from a second client for the data object; a sixth server that stores a sixth plurality of data objects and is configured to request the data object from the third server if the data object is not present in the sixth plurality of data objects, wherein: the sixth server is one of the second plurality of application servers; and the sixth server is configured to receive a request from a third client for the data object; and a seventh server that stores a seventh plurality of data objects and is configured to request the data object from the third server if the data object is not present in the seventh plurality of data objects, wherein: the seventh server is one of the second plurality of application servers; and the seventh server is configured to receive a request from a fourth client for the data object.

As per claim 5, cloud data persistence system for distribution of data, the cloud data persistence system comprising: a first server which stores a first plurality of data objects and is communicatively coupled with a second server and a third server, wherein: the first server is located at a first location; each of the first plurality of data objects comprises a link to a media asset of a plurality of media assets; the plurality of media assets is located at a second location; and the second location is distinct from

the first location; the second server stores a second plurality of data objects and is configured to request a data object from the first server if the data object is not present in the second plurality of data objects, wherein the second server is communicatively coupled with a fourth server and fifth server; and the fourth server stores a third plurality of data objects and is configured to request the data object from the second server if the data object is not present in the third plurality of data objects, wherein the fourth server is configured to receive a request from a first client for the data object.

As per claim 14, method for retrieving data objects using a cloud data persistence system configured for distribution of data, the method comprising: receiving, by an application center, a first request from a client for a data object, wherein: the application center is one of a plurality of application centers that communicate with a plurality of clients; the application center is located at a first location; the data object comprises a link to a media asset; and the media asset is stored at a second location separate from the first location; determining, by the application center, the data object is not stored among a first plurality of data objects stored at the application center; transmitting, by the application center, a second request for the data object to a first cache server, wherein: the first cache server is communicatively coupled with the plurality of application centers; and the first cache server is one of a plurality of cache servers; receiving, by the cache server, the second request for the data object; determining, by the cache server, the data object is not stored among a second plurality of data objects stored at the cache server; transmitting, by the cache server, a third request for the data object to an origin server, wherein the origin server is

communicatively coupled with the plurality of cache servers; receiving, by the origin server, the third request for the data object; locating, by the origin server, the data object among a third plurality of data objects; transmitting, by the origin server, the data object linked to the third request to the cache server; receiving, by the cache server, the data object linked to the third request; transmitting, by the cache server, the data object linked to the second request; receiving, by the application center, the data object linked to the second request; and transmitting, by the application center, the data object linked to the first request to the client.

In particular the prior art does not teach the claims as it relates to Fig.5A,B, and supporting para.52-68, see also Remarks filed on 5/18/12.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BACKHEAN TIV whose telephone number is (571)272-5654. The examiner can normally be reached on M-T 7-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (571) 272-3964. The fax phone

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Backhean Tiv/
Primary Examiner, Art Unit 2451

Notice of References Cited	Application/Control No. 13/017,866		Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.	
	Examiner BACKHEAN TIV		Art Unit 2451	Page 1 of 2

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,852,717 A	12-1998	Bhide et al.	709/203
*	B	US-6,405,240 B1	06-2002	Tsubone et al.	709/203
*	C	US-2006/0265427 A1	11-2006	Cohen et al.	707/200
*	D	US-2007/0083537 A1	04-2007	Martinez, Ronald	707/101
*	E	US-2007/0294295 A1	12-2007	Finkelstein et al.	707/104.1
*	F	US-2008/0294748 A1	11-2008	Brown et al.	709/219
*	G	US-7,516,194 B1	04-2009	Lamkins et al.	709/218
*	H	US-7,613,815 B1	11-2009	Prakash et al.	709/230
*	I	US-7,716,282 B2	05-2010	Takeshima et al.	709/203
*	J	US-7,809,840 B2	10-2010	Jing et al.	709/227
*	K	US-2010/0287159 A1	11-2010	ABAJIAN et al.	707/723
*	L	US-2011/0016197 A1	01-2011	Shiimori et al.	709/219
*	M	US-2011/0029883 A1	02-2011	Lussier et al.	715/738

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Notice of References Cited	Application/Control No. 13/017,866		Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.	
	Examiner BACKHEAN TIV		Art Unit 2451	Page 2 of 2

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,921,116 B2	04-2011	Finkelstein et al.	707/746
*	B	US-7,953,820 B2	05-2011	Stevens et al.	709/219
*	C	US-2012/0005276 A1	01-2012	Guo et al.	709/206
*	D	US-2012/0030317 A1	02-2012	Smyth et al.	709/219
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes 	Application/Control No. 13017866	Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.
	Examiner BACKHEAN TIV	Art Unit 2451

SEARCHED			
Class	Subclass	Date	Examiner
709	219-225(limited search)	2/2/12	BT

SEARCH NOTES		
Search Notes	Date	Examiner
Inventor's Name Search PALM	2/2/12	BT
EAST txt Search	2/2/12	BT
Update Search	5/23/2012	BT

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
709	219-225(limited search)	5/23/2012	BT

	/BACKHEAN TIV/ Primary Examiner.Art Unit 2451
--	--

<i>Index of Claims</i> 	Application/Control No. 13017866	Applicant(s)/Patent Under Reexamination MCGOWAN ET AL.
	Examiner BACKHEAN TIV	Art Unit 2451

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☒ Claims renumbered in the same order as presented by applicant				☐ CPA		☐ T.D.		☐ R.1.47		
CLAIM		DATE								
Final	Original	02/02/2012	05/23/2012							
	1	✓	=							
	2	✓	=							
	3	✓	=							
	4	✓	=							
	5	✓	=							
	6	✓	=							
	7	✓	=							
	8	✓	=							
	9	✓	=							
	10	✓	=							
	11	✓	=							
	12	✓	=							
	13	✓	=							
	14	✓	=							
	15	✓	=							
	16	✓	=							
	17	✓	=							
	18	✓	=							
	19	✓	=							
	20	✓	=							

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L5	1071	meta\$data with proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/05/23 10:02
L6	78	stor\$4 adj2 meta\$data with proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/05/23 10:04
L7	0	stor\$4 adj2 meta\$data with proxy with link with (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:05
L8	0	stor\$4 adj2 meta\$data same proxy with link with (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:05
L9	0	stor\$4 adj2 meta\$data same proxy with link same (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:05
L10	3	stor\$4 adj2 meta\$data same proxy same link same (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:05
L11	51	stor\$4 adj2 meta\$data and proxy same link same (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:06
L12	15	stor\$4 adj2 meta\$data and proxy same link with (media or multi\$media or video or audio) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:06
L13	13	stor\$4 adj2 meta\$data with (database or server or proxy) same (search\$4)	US-PGPUB; USPAT;	OR	ON	2012/05/23 10:11

		with (access\$4) with (media or multi\$media or audio or video) and (@ad<="20100701" or @rlad<="20100701")	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
L14	8989	(cache or proxy) with application near3 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:21
L15	54	(cache or proxy) with application near3 server same ("data object" or meta\$data) with (URL or link) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2012/05/23 10:22
S1	4	((("20030061272") or ("7373394")).PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; DERWENT; IBM_TDB	OR	OFF	2012/01/26 09:14
S2	299	proxy near4 cache with (content or orgin) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:12
S3	34	proxy near4 cache with (content or orgin) near4 server same application near3 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:13
S4	118	(user or client) near4 (send\$4 or transmit\$4) near4 request\$4 near5 (object or URL) with application near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:26
S5	5	(user or client) near4 (send\$4 or transmit\$4) near4 request\$4 near5 (object or URL) with application near4 server same proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:26
S6	130	proxy near3 request\$4 near4 (another or different) near4 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:27
S7	579	proxy near3 request\$4 near4 (another or different or second) near4 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:27
S8	2	proxy near3 request\$4 near4 (another or different or second) near4 proxy	US-PGPUB; USPAT;	OR	OFF	2012/02/02 10:27

		near4 level and (@ad<="20100701" or @rlad<="20100701")	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S9	44142	(709/219-225).OCLS.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:28
S10	106	S9 and S7	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:28
S11	2	("20050044242").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 10:41
S12	4432	application near3 server with (proxy or cache) near3 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:02
S13	1464	application near3 server with (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:02
S14	231	S13 and S9	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S15	1	application near3 server with downstream near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S16	1	application adj2 server with downstream near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S17	708	application adj2 server near3 (proxy or cache) near3 server with (content or web) near4 server and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:07
S18	123	S17 and S9	US-PGPUB; USPAT;	OR	OFF	2012/02/02 11:07

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S19	25	(application adj2 server near3 (proxy or cache) near3 server with (content or web) near4 server).ab,ti. and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:08
S20	3095	application adj2 proxy and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:13
S21	62	application adj2 proxy with user near3 request\$4 and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:14
S22	10	(application adj2 proxy with user near3 request\$4).ab,ti. and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2012/02/02 11:15

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	46449	(709/219-225).CCLS.	US-PGPUB; USPAT; UPAD	OR	OFF	2012/05/23 09:59
L16	54	(cache or proxy) with application near3 server same ("data object" or meta\$data) with (URL or link) and (@ad<="20100701" or @rlad<="20100701")	US-PGPUB; USPAT; UPAD	OR	ON	2012/05/23 10:27

5/ 23/ 2012 10:41:07 AM
C:\ Users\ btiv\ Documents\ EAST\ Workspaces\ 13017866_cloud_cache.wsp

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

20350 7590 05/30/2012
KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	01/31/2011	Albert J. McGowan	92853-798163	7494

TITLE OF INVENTION: CLOUD DATA PERSISTENCE ENGINE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	\$870	\$300	\$0	\$1170	08/30/2012

EXAMINER	ART UNIT	CLASS-SUBCLASS
TIV, BACKHEAN	2451	709-219000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Kilpatrick Townsend & Stockton LLP

2 _____

3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.111. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Unicorn Media, Inc.

Tempe, AZ

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

- ☒ Issue Fee
- ☒ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)

- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature /Karam J. Saab/

Date August 30, 2012

Typed or printed name Karam J. Saab

Registration No. 64,190

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Electronic Patent Application Fee Transmittal

Application Number:	13017866			
Filing Date:	31-Jan-2011			
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE			
First Named Inventor/Applicant Name:	Albert J. McGowan			
Filer:	Karam J. Saab/Rita Christensen			
Attorney Docket Number:	92853-798163			
Filed as Small Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	2501	1	870	870
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1170

Electronic Acknowledgement Receipt

EFS ID:	13630471
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Karam J. Saab
Filer Authorized By:	
Attorney Docket Number:	92853-798163
Receipt Date:	30-AUG-2012
Filing Date:	31-JAN-2011
Time Stamp:	17:31:11
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 1170
RAM confirmation Number	4555
Deposit Account	201430
Authorized User	SAAB,KARAM

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	ISSUEFEE.pdf	1782691	no	1
			581d516e61513c05ed5de403aa0d75e789a37a9e		

Warnings:

Information:

2	Fee Worksheet (SB06)	fee-info.pdf	31831	no	2
			915d1c539a1f10cc29638b05234852f166a16c4b		

Warnings:

Information:

Total Files Size (in bytes):			1814522
------------------------------	--	--	---------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/017,866	10/02/2012	8280987	92853-798163	7494

20350 7590 09/12/2012

KILPATRICK TOWNSEND & STOCKTON LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Albert J. McGowan, Phoenix, AZ;
Richard L. Carls, Scottsdale, AZ;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

PATENT
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

December 22, 2017

Patent No. 8,280,987

Issue Date: October 2, 2012

Title: CLOUD DATA PERSISTENCE ENGINE

Application No.: 13/017,866

Filed: January 31, 2011

Inventor(s): Albert J. McGowan; Richard L. Carls

Attorney Docket No. 092853-000700US-0798163

Notification of Loss of Entitlement to Small Entity Status Pursuant to 37 C.F.R. §1.27(g)(2)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This communication hereby notifies the United States Patent and Trademark Office that small entity status is no longer applicable for the above-identified patent; the correct status for the referenced patent is **large**. The current owner of the subject patent, Brightcove, Inc., now meets the requirements for large entity status. Accordingly, Applicant requests the Office update its records to reflect the change of entity status from small to large for the instant U.S. Patent.

Date: December 22, 2017

Respectfully submitted,

/Scott L. McMillan/
Scott L. McMillan
USPTO Reg. No. 62079

Electronic Acknowledgement Receipt

EFS ID:	31326592
Application Number:	13017866
International Application Number:	
Confirmation Number:	7494
Title of Invention:	CLOUD DATA PERSISTENCE ENGINE
First Named Inventor/Applicant Name:	Albert J. McGowan
Customer Number:	20350
Filer:	Scott Lee McMillan/Shannon Serrano
Filer Authorized By:	Scott Lee McMillan
Attorney Docket Number:	92853-798163
Receipt Date:	22-DEC-2017
Filing Date:	31-JAN-2011
Time Stamp:	17:36:21
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Notification of loss of entitlement to small entity status	092853-0798163_Notification.pdf	83714 39860820041fa6cf1a7417435aab46ede2d71197	no	1

Warnings:

Information:	
Total Files Size (in bytes):	83714
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>	

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	13/017,866
Filing Date	2011-01-31
First Named Inventor	Albert J. McGowan
Title	CLOUD DATA PERSISTENCE ENGINE
Art Unit	2451
Examiner Name	BACKHEAN TIV
Attorney Docket Number	40BC7US-NP

SIGNATURE of Applicant or Patent Practitioner

Signature	/Thomas D. Franklin/	Date (Optional)	2023-04-18
Name	Thomas D. Franklin	Registration Number	43616
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.



*Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☒ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above:

191546

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☒ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name

Address

City

State

Zip

Country

Telephone

Email

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Brightcove Inc.

- ☐ Inventor or Joint Inventor (title not required below)
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)
- ☒ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature / Shakir Hussaini / Date (Optional) 02/07/2023

Name Shakir Hussaini

Title Commercial Counsel

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☒ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc ID: c3dd6e3f22d4c3018a8efa9bd29780087b43b55

Petitioner Disney Ex-1004, 0254

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #
13/017,866

RECEIPT DATE / TIME
04/19/2023 12:29:04 PM ET

ATTORNEY DOCKET #
92853-798163

Title of Invention

CLOUD DATA PERSISTENCE ENGINE

Application Information

APPLICATION TYPE Utility - Nonprovisional Application
under 35 USC 111(a)

PATENT # 8280987

CONFIRMATION # 7494

FILED BY Dana Montgomery

PATENT CENTER # 61941436

FILING DATE 01/31/2011

CUSTOMER # 20350

FIRST NAMED INVENTOR Albert J. McGowan

CORRESPONDENCE ADDRESS -

AUTHORIZED BY Thomas Franklin

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
40BC7US-NP_CoverSheetPOA.pdf	3	Power of Attorney	1959 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
40BC7US-NP_CoverSheetPOA.pdf	37BB38A3A91E1AE7E6B12B35A16DEBEDF3AA3E8EC71AA5F8 911D41AF8CCB5313530EAD EB08B0A7D19BD76B903A80058F8 56F427C4E0A91EAE99F43E7518270E3

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized

by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	40BC7US-NP

CONFIRMATION NO. 7494

POA ACCEPTANCE LETTER



OC000000137815976

Date Mailed: 04/27/2023

191546
Mughal IP P.C.
1500 NW Bethany Blvd Ste 200
Beaverton, OR 97006-5208

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/19/2023.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/tmwilliams/



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	92853-798163

CONFIRMATION NO. 7494

POWER OF ATTORNEY NOTICE



OC000000137815975

20350
Kilpatrick Townsend & Stockton LLP - West Coast
Mailstop: IP Docketing - 22
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309

Date Mailed: 04/27/2023

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/19/2023.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/tmwilliams/



UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #
13/017,866

RECEIPT DATE / TIME
07/11/2024 12:56:29 PM Z ET

ATTORNEY DOCKET #
40BC7US-NP

Title of Invention

CLOUD DATA PERSISTENCE ENGINE

Application Information

APPLICATION TYPE Utility - Nonprovisional Application
under 35 USC 111(a)

PATENT # 8280987

CONFIRMATION # 7494

FILED BY Carolina Rodriguez

PATENT CENTER # 66303916

FILING DATE 01/31/2011

CUSTOMER # 191546

FIRST NAMED INVENTOR Albert J. McGowan

CORRESPONDENCE ADDRESS -

AUTHORIZED BY Kevin Rieffel

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
001511-0003-101 373c Statement.pdf	2	Assignee showing of ownership per 37 CFR 3.73	119 KB
001511-0003-101 Power of Attorney.pdf	2	Power of Attorney	150 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
001511-0003-101 373c Statement.pdf	C832B08E7844CE2060C34CE3FB6F4591B3841BF0A5D4002338 0A76933676909C0C212AE8DA77F96C906886E7D49B4A8D05210

2A571D307138EB9FB72B91066D9A

001511-0003-101 Power of
Attorney.pdf

39B3571A0BC996BFE69DE2EFA2106043D0CC492C389155A67
D3B24B587E5196F7365A55D6139C65E826CF738E138136A0F0
36C793F4B883941691D1A0AB7DF7E

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)First Named Inventor/Patent Owner: ADEIA MEDIA HOLDINGS LLCApplication No./Patent No.: 13/017,866 (Patent No. 8,280,987) Filed/Issue Date: January 31, 2011 (Issue Date October 2, 2012)Titled: CLOUD DATA PERSISTENCE ENGINEADEIA MEDIA HOLDINGS LLC, a Limited Liability Company

(Name of Assignee)

(Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☒ The assignee of the entire right, title, and interest.
2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):
- ☐ The extent (by percentage) of its ownership interest is ____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.
- ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (e.g., bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

- A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached*.
- B. ☒ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Albert J. McGowan To: UNICORN MEDIA, INC.

The document was recorded in the United States Patent and Trademark Office at

Reel 026230, Frame 0734, or for which a copy thereof is attached*.2. From: UNICORN MEDIA, INC. To: CACTI ACQUISITION LLC

The document was recorded in the United States Patent and Trademark Office at

Reel 032792, Frame 0728, or for which a copy thereof is attached*.

[Page 1 of 2]

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0031. Public burden for this form is estimated to average 12 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

STATEMENT UNDER 37 CFR 3.73(c)3. From: CACTI ACQUISITION LLC To: BRIGHTCOVE INC.

The document was recorded in the United States Patent and Trademark Office at

Reel 034745, Frame 0158, or for which a copy thereof is attached*.4. From: BRIGHTCOVE INC. To: ADEIA MEDIA HOLDINGS LLC

The document was recorded in the United States Patent and Trademark Office at

Reel 067609, Frame 0435, or for which a copy thereof is attached*.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).☒ *As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Kevin A. Rieffel/2024-07-02

Signature

Date

Kevin A. Rieffel71,318

Printed or Typed Name

Title or Registration Number

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA/82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	13/017,866 (Patent No. 8,280,987)
Filing Date	January 31, 2011 (Issue Date October 2, 2012)
First Named Inventor	Albert J. McGowan
Title	CLOUD DATA PERSISTENCE ENGINE
Art Unit	2451
Examiner Name	Backhean Tiv
Attorney Docket Number	001511-0003-101

SIGNATURE of Applicant or Patent Practitioner

Signature	/Kevin A. Rieffel/	Date (Optional)	2024-07-02
Name	Kevin A. Rieffel	Registration Number	71,318
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.



*Total of 1 forms are submitted.

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0035. Public burden for this form is estimated to average 3 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.** If filing this completed form by mail, send to: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☒ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above:

75563

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☒ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name

Address

City

State

Zip

Country

Telephone

Email

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

ADEIA MEDIA HOLDINGS LLC

- ☐ Inventor or Joint Inventor (title not required below)
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)
- ☒ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	Paul Rodriguez	Date (Optional)	4/25/2024
Name	Paul Rodriguez		
Title	Senior Vice President, Head of IP Portfolio Management		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☒ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	001511-0003-101

CONFIRMATION NO. 7494

POA ACCEPTANCE LETTER



OC000000075461964

75563
Haley Guiliano LLP (Adeia)
75 Broad Street
Suite 1000
New York, NY 10004-3226

Date Mailed: 07/24/2024

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 07/11/2024.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/mnguyen/



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/017,866	01/31/2011	Albert J. McGowan	40BC7US-NP

CONFIRMATION NO. 7494

191546
MUGHAL GAUDRY & FRANKLIN PC
1500 NW Bethany Blvd Ste 200
Beaverton, OR 97006-5208

POWER OF ATTORNEY NOTICE



OC000000075461852

Date Mailed: 07/24/2024

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 07/11/2024.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at (571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/mnguyen/