(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification:
H04L 12/66 (2006.01)

(21) International Application Number:
PCT/US2006/029571

(22) International Filing Date: 28 July 2006 (28.07.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/703,687   28 July 2005 (28.07.2005)   US

(71) Applicant *(for all designated States except US)*: POR-TAL SOFTWARE, INC. [US/US]; 10200 South De Anza Blvd., Cupertino, CA 95014 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: LABUDA, David, Scott [US/US]; 555 Bryant Street, #604, Palo Alto, CA 94301 (US). KRISHNAMOORTHY, Jayaprakash [US/US]; 2252 Lenox Pl., Santa Clara, CA 95054 (US). HADDOCK, James, R. [US/US]; 692 Clipper Street, San Francisco, CA 94114 (US). ROCKEL, Alexander

[DE/DE]; Luttenredder 32, 22457 Hamburg (DE). BRE-FCZYNSKI, Keith, M. [US/US]; 5636 Stevens Creek Blvd., Apt. 381, Cupertino, CA 95014 (US). DOUGLAS, Giles [GB/US]; 96 El Dora Dr, Mountain View, CA 94041 (US).

(74) Agent: LEVINE, David, A.; Levine Bagade Han LLP, 2483 East Bayshore Road, Suite 100, Palo Alto, CA 94303 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

(54) Title: REVENUE MANAGEMENT SYSTEM AND METHOD
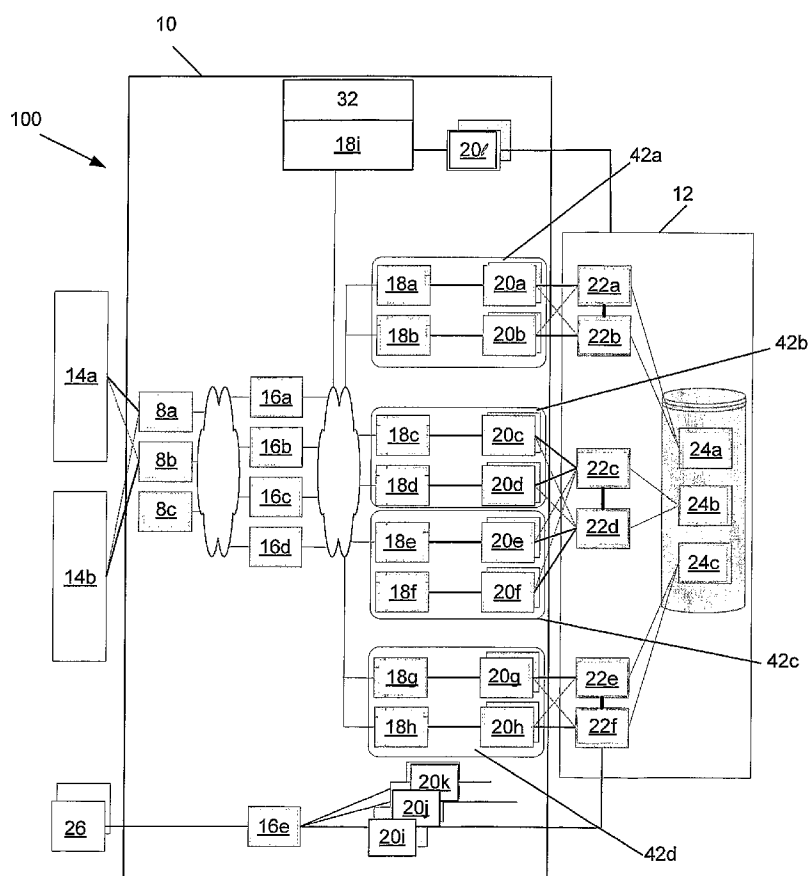
(57) Abstract: A real-time customer relation management system is disclosed. The system can provide increased availability, reduced internal latencies, and reduced data processing and transfer. The system can provide real time processing and batch processing. The system architecture can have an in-memory write-through cache. The cache can store data that would have otherwise been sent to a database. The system can have a backup in-memory write-through cache. The system can use a warm standby, for example, to enhance data backup efficiency.

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

Published:

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

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

1   TITLE OF THE INVENTION
2   REVENUE MANAGEMENT SYSTEM AND METHOD
3
4   CROSS-REFERENCE TO RELATED APPLICATIONS
5   [0001] This application claims the benefit of U.S. Provisional Application No.
6   60/703,687 filed 28 July 2005 which is herein incorporated by reference in its
7   entirety.
8
9   BACKGROUND OF THE INVENTION
10   [0002] This invention relates to a revenue management system that has an in-memory
11   write-through cache.
12   [0003] Prepaid phone accounts are tracked in real-time by billing and time
13   management hardware and software architectures in communication with the phone
14   network switch. The architecture approves the customer's call if there are sufficient
15   funds in the customer's prepaid account. If the customer runs out of time on his
16   prepaid account during a call, the architecture acts to terminate the call.
17   [0004] These architectures are known as customer relation management (CRM)
18   systems. CRM systems are also used for non-prepaid scenarios, such as for
19   generating bills. CRM systems are also used for other telecommunications, and other
20   network management scenarios.
21   [0005] Prepaid account CRM systems need to have the ability to track accounts in
22   real-time. Available real-time architectures for managing prepaid customer accounts
23   have some existing limitations.
24   [0006] First, the available architectures require high performance and data
25   throughput, thereby leading to relatively high hardware requirements. These
26   architectures, along with their storage and maintenance can be expensive and time-
27   consuming.
28   [0007] Second, requirements for very low system response latencies are difficult to
29   achieve. Transactions in existing architectures involve several round-trips to the disk-
30   based storage subsystem. The data must be processed by a comparatively large
31   software stack to transform from a relational representation into a physical storage
32   format.
33   [0008] Third, in an available architecture, the data is transferred several times from
34   component to component of the system to retrieve the data, map it from a relational

1   format to an object format, process it with the desired business logic, and then transfer

2   the response to the client.

3   [0009] Fourth, currently available architectures can not provide desired levels of data

4   availability to the public phone network (e.g., the switch). The close connection of a

5   prepaid CRM system to the public network increases the data availability

6   requirements. Being part of the public network, some parts of the system need to

7   have carrier-grade availability.

8   [0010] Also, no single product accomplished both batch processing and real-time

9   processing for telecommunications CRM (e.g., billing) purposes.

10

11   BRIEF SUMMARY OF THE INVENTION

12   [0011] A system and method for managing any numerical account information is

13   disclosed. For example, the system and method can be used for managing revenue for

14   telecommunications system. The system and method can be used to manage account

15   balances, such as user accounts for the telecommunications system. Management of

16   account balances can include altering the balance of the account during use, and/or

17   querying the account (e.g., by the account holder or a customer service

18   representative), and/or querying the account to produce a billing statement or perform

19   other accounting features, and/or querying the account to determine whether to

20   authorize use of the account.

21   [0012] The system can be used with an account with an existing balance (e.g.,

22   prepaid), an account with a maximum use limit (e.g., capped), a current payment

23   account (e.g., now-pay, for example through the use of a credit card), other types of

24   balance management accounts, or combinations thereof.

25   [0013] The system architecture can be configured to increase performance, and

26   availability and decrease latency. The system and method can manage accounts, for

27   example, for the prepaid wireless markets handling services such as GSM, GPRS and

28   SMS.

29   [0014] The system can have a rating engine, a billing engine, and a first, high-speed,

30   memory (e.g., transaction in memory object store (TIMOS)). The first memory can

31   be a virtual database cache. The first memory can be a typical on-board RAM storage

32   location.

1   [0015] The first memory can be a smart cache. The smart cache can treat different

2   object types different ways. For example, the smart cache can treat reference objects,

3   database-only objects, and transient object differently.

4   [0016] Reference objects can be owned by the database and never updated by the first

5   memory. Reference objects can include dynamic reference objects (e.g., an account

6   balance) that change each call, and static reference objects (e.g., the billing rate for

7   different types of calls) that never or rarely change. Database-only objects can be

8   objects that change one-time or rarely during the call and are not referred to by the

9   connection manager. Transient objects can exist, for example, only in-memory (e.g.,

10   in TIMOS). Transient objects can be unwritten to the database. Transient objects can

11   be written to the database, for example, at the end of the call (e.g., credit balance).

12   [0017] The database can have a data dictionary. The data dictionary can by written

13   by the users. The data dictionary can define an object type and what type of object

14   each other is. Customers can edit the data dictionary if so desired.

15   [0018] The new revenue management system can have a high availability. The

16   system can have a warm standby operation by referring to any data remaining in

17   TIMOS. During warm standby, in the case of a loss of data, the system can recreate

18   data from the switch and/or TIMOS when the switch sends re-authorization data (e.g.,

19   during long calls) or end-of-call data.

20   [0019] A known failure protection scheme with a high availability (monitor) regularly

21   checking the status of the control manager, TIMOS, data manager, the database

22   manager, and the database is also disclosed.

23   [0020] A self-container failure protection system is disclosed. Each component of the

24   system can check on the status of its immediately downstream component. If the

25   downstream component has failed, or is passing along a failure message regarding a

26   further downstream component failure, the system can take appropriate action,

27   including alerting a user.

28

29   BRIEF DESCRIPTION OF THE DRAWINGS

30   [0021] Figure 1 illustrates a variation of the revenue management system architecture

31   connected to a switch over a network.

32   [0022] Figure 2 illustrates a variation of the revenue management system architecture

33   connected to a switch over a public network.

1   [0023] Figures 3 through 5 illustrate variations of the revenue management system

2   architecture.

3   [0024] Figure 6 illustrates a variation for a method for using the revenue management

4   system.

5   [0025] Figure 7 illustrates process flows for variations for methods for using the

6   revenue management system.

7   [0026] Figure 8 illustrates a variation for a method for using the revenue management

8   system.

9   [0027] Figure 9 illustrates process flows for variations for methods for using the

10   revenue management system.

11   [0028] Figure 10 illustrates a variation for a method for using the revenue

12   management system.

13   [0029] Figure 11 illustrates process flows for variations for methods for using the

14   revenue management system.

15   [0030] Figures 12a through 14 illustrate variations of the revenue management

16   system.

17

18   DETAILED DESCRIPTION

19   [0031] A computer-based system and method for managing any numerical account

20   information is disclosed. For example, the system and method can be used for

21   managing revenue for telecommunications system. The system and method can be

22   used to manage account balances, such as user accounts for the telecommunications

23   system. The management of account balances can include altering the balance of the

24   account during use, and/or querying the account (e.g., by the account holder or a

25   customer service representative), and/or querying the account to produce a billing

26   statement or perform other accounting features, and/or querying the account to

27   determine whether to authorize use of the account.

28   [0032] Figure 1 illustrates a telecommunication device 2, such as a phone, computer,

29   or fax machine, that can be connected through a public telephone network 4 to a

30   switch 6. The telecommunication device 2 can be communicating with a second

31   telecommunication device through the switch 6. The switch 6 can communicate

32   across a network and through a gateway 8 (e.g., having a protocol translator) to the

33   revenue management system 100. The gateway 8 can communicate directly with a

34   business logic module 10 or business logic application (e.g., Portal Infranet, Portal

1   Software, Inc. Cupertino, CA). The business logic module 10 can communicate with

2   a database system 12 to determine whether the telecommunication device 2 connected

3   to the switch 6 has permission to connect and/or stay on the line. The database

4   system 12 can have a highly-available Oracle RAC database cluster. The system 100

5   can utilize Oracle transaction management functionality.

6   [0033] Figure 2 illustrates that the gateway 8 can be a part of the revenue

7   management system 100. The gateway 8 can interface between the business logic

8   module 10 and an intelligent network (IN) service control point system (SCP) 14.

9   The service control point system 14 can facilitate communication between the switch

10   6 and the gateway 8. The SCP system 14 can be software or a remote computer

11   database within the network that receives queries, for example from service switching

12   points (SSP), in order to process applications, such as 800 and LNP number lookups

13   and calling card verification. The SCP system 14 can process the applications

14   utilizing the customer management system 100. The gateway 8 can be a high-speed

15   protocol translator from the IN SCP to the remainder of the revenue management

16   system 100.

17   [0034] Figure 3 illustrates that the business logic module 10 can have one or more

18   rating connection managers (CM) 16a and 16b, a first memory data manager 18 (e.g.,

19   TIMOS Data Manager (DMT) from Portal Software, Inc.), and one or more second

20   memory data managers 20a and 20b (e.g., Oracle Data Manager (DM Oracle). The

21   second memory data managers 20a and 20b can communicate with the database

22   system 12 or other second memory system. The database system 12 can have one or

23   more database clusters 22a and 22b (e.g., Oracle Real Application Clusters), for

24   example, providing high availability and scalability for databases running on the

25   cluster. The database clusters 22a and 22b can support one or more databases 24.

26   [0035] The business logic module can be accessed via the gateway 8 and/or via a

27   manual access application 26. The manual access application 26 can be operated

28   manually or automatically. The manual access application 26 can be configured, for

29   example, to be used by billing software to generate invoices, and/or by a customer

30   service representative to check on account status, and/or by the account-holder to

31   check account status.

32   [0036] The revenue management system 100 can have a first memory (e.g., TIMOS)

33   and a second memory (e.g., database). The first memory can be, for example, in

34   and/or in communication with the first memory data manager 18. The first memory

1   can be configured to have faster, slower, and/or the same read, and/or write, and/or re-

2   write speeds (e.g., access speeds) as the second memory. The first memory can be an

3   in-memory data store and database cache dedicated to high-speed rating and

4   authorization requirements.

5   [0037] The first memory can be solid state memory, such as system memory (e.g.,

6   RAM) or one or more hard drives, for example with fast access speeds. Requests for

7   data in the first memory can be processed faster than requests for data in a second

8   memory.

9   [0038] The first data in a first data object can be stored in the first memory in the

10   format used by the business logic module 10 (e.g., Portal Infranet, Portal Software,

11   Inc.). The first data can be left untranslated before storage in the first memory. The

12   internal search and storage algorithms can be optimized for in-first-memory data.

13   Storing the first data in the first memory can, for example, eliminate the round trip to

14   the second memory (e.g., one or more databases, such as on database servers), and

15   can speed the process of storing, editing and/or querying the first data. Object

16   creation or updates for the first data objects can require no access of the second

17   memory. Updates for the first data objects can be performed in the first memory. The

18   system can have, for example, a reduced throughput and/or latency.

19   [0039] The first memory data objects (e.g., transient objects) can be stored in the first

20   memory and/or the second memory. For example, the first memory data objects can

21   be stored not in the database and not be persisted in the first memory. The first

22   memory objects can, for example, exist only in the process heap memory of the first

23   memory. The first memory objects can be, for example, managed in a transactional

24   manner (e.g., like the other memory objects).

25   [0040] First memory data objects can be removed from the first memory by shutdown

26   of the first memory or the business logic executing a delete operation on the first

27   memory data object. The store for first memory data objects can be a fixed size, for

28   example, determined during startup of the first memory process.

29   [0041] The first memory data manager 18 can be configured to improve access times

30   and latency on moving and/or writing and/or editing and/or deleting and/or querying

31   objects.

32   [0042] The second memory can be in and/or in communication with the second

33   memory data manager 20. Requests for second memory data objects can be sent to

34   the second memory data manager 20. The second memory, for example, can be a

1   disk-based (e.g., on one or more hard drives) database. The database can be a

2   relational database (RDBMS).

3   [0043] The system can have low access second memory data objects (e.g., database-

4   only objects). The low access second memory data objects can be stored primarily

5   and/or exclusively in the second memory (e.g., one or more databases). The low

6   access second memory data objects can be stored in the first memory none of the

7   time, or some of the time.

8   [0044] The first memory data manager can access the low access second memory data

9   object type via a pass-through mode. For example, requests can be forwarded to the

10   second memory data manager (e.g., DM_Oracle), and responses can be forwarded

11   back to the first memory data manager.

12   [0045] The high access second memory data objects (e.g., reference objects) can be

13   updated seldom and not during high-speed session processing. The high access

14   second memory data objects can be stored (cached) in a first memory reference object

15   cache (ROC). The high access second memory data objects can grow in number in

16   relation to growth in the subscriber base.

17   [0046] The high access second memory data objects can exist in the first memory an

18   equal amount of time as length of the first memory process. A newly started first

19   memory instance can contain no high access second memory data objects.

20   [0047] Updating and creating high access second memory data objects can be

21   performed in the second memory and in the first memory. in the high access second

22   memory data objects can be updated or created asynchronously or synchronously in

23   the second memory and the first memory.

24   [0048] The high access second memory data objects can be static or dynamic. The

25   static high access second memory data objects can be queried, updated, created, or

26   deleted at irregular intervals. The static high access second memory data objects can

27   be, for example, subscriber information such as the list of subscribed services and the

28   chosen tariff plans.

29   [0049] The dynamic high access second memory data objects can be touched (e.g.,

30   queried, updated, created, deleted) after the completion of each session. The dynamic

31   high access second memory data objects can be, for example, the monetary and non-

32   monetary balances belonging to a subscriber account.

33   [0050] A standby-first memory (e.g., for a high availability variation that can have an

34   active first memory and a backup, standby first memory) can preload the static high

1   access second memory data objects. Changes of static reference objects can be

2   propagated from the active first memory to the standby first memory.

3   [0051] The gateway 8 can directly communicate with the business logic module 10.

4   For example, during a customer's use of the telecommunication network, the gateway

5   8 can communicate with a first connection manager (CM) 16a. The gateway 8 can

6   pass requests to the CM 16a, for example, calling the appropriate business logic

7   routines depending on the type of request that is indicated from the IN SCP 14. The

8   gateway 8 can be nearly stateless. The gateway 8 can provide fast failover

9   capabilities, for example, accompanied by a degraded mode of operation that is used

10   when the lower architecture layers become unavailable. The gateway 8 can perform

11   authentication, authorization and accounting procedures.

12   [0052] Events received by the CM can be rated via an embedded rating engine using

13   the data provided from the first memory data manager 18 (e.g., DM TIMOS cache)

14   and the database system 12. The rating engine can produce rates for customer use of

15   the telecommunications network under the specific conditions that apply (e.g., time of

16   day, day of week, network used). The rating engine can cache pricing objects itself,

17   for example, in order to reduce the number of network roundtrips necessary to

18   complete the rating phase. The rating engine can perform zoning and discounting

19   rating.

20   [0053] Based on the object type, the first memory data manager 18 can pass the

21   request to the database system 12, query the first memory data manager 18 reference

22   object cache or accesses the first memory (e.g., in-memory store) for transient objects.

23   The object types and their locations can be defined in a business logic database (e.g.,

24   Infranet Data Dictionary by Portal Software, Inc.), which can be in the database

25   system 12. Traffic for objects not in the first memory can be allowed to bypass the by

26   accessing the database manager 20, for example, in the same way a commonly used

27   system without the first data manager would be configured. The data integrity of the

28   first memory can be ensured by a platform-managed synchronization mechanism that

29   can propagate the necessary updates to the first memory. The first memory can have

30   one or more caches.

31   [0054] Figures 4a and 4b illustrate variations of the revenue management system 100.

32   The gateway 8 can act as a high-speed protocol translator as well as an SLA monitor

33   with fallback capabilities. The CM 16 can receive requests from the gateway 8. The

34   CM 16 can have the authorization, authentication and accounting business logic (e.g.,

1   for delivery to the gateway 8). The CM 16 can call operational codes on the first data

2   manager 18. The CM 16 can be replaced with another client, such as a migration tool.

3   The CM 16 can have a realtime pipeline (RTP) 28. The RTP 28 can be configured to

4   adjust the rating, for example by discounting and zoning the rate. The RTP 28 can be

5   optionally used by the CM 16 while rating.

6   [0055] The first data manager 18 can have a data migratory subsystem 30. The data

7   manager subsystem 30 can be used to fill the high access second memory object cache

8   after start or fail over.

9   [0056] The first data manager 18 can have a directory server 32. The directory server

10   32 can be configured to identify the correct first memory/second memory

11   combinations in scaled scenarios with more second memory instances than first

12   memory instances or more first memory instances than second memory instances.

13   The directory server 32 can enable the gateway instances and CM instances to be

14   independent of the number of first memory instances. The number and location of

15   gateway and CM processes can be flexibility and scalability with respect to the

16   number and location of first memory instances.

17   [0057] Figure 4a shows that the elements of the architecture of the business logic

18   module can all be standalone. Figure 4b illustrates that the numerous elements of the

19   architecture can be integrated.

20   [0058] Figure 5 illustrates that the first memory 102 can have a reference object cache

21   (ROC) 34 and a transient object store (TOC) 36. The ROC 34 can be managed by a

22   separate set of rules than the TOC 36. The ROC 34 and the TOC 36 can be in the

23   same or different parts of the first memory 102. The first memory 102 can be part of,

24   or separate but in communication with, the first memory data manager 18. The ROC

25   34 can be configured to cache high access second memory data objects (e.g.,

26   reference objects). The TOC 36 can be configured to store first memory data objects

27   (e.g., transient objects).

28   [0059] Figure 6 illustrates a method for accessing a first memory data in the TOC 36.

29   The CM 16 can send, shown by arrow 38, a request to the first data manager 18. The

30   request can apply to the first memory data. The first data manager 18 can analyze the

31   request 38. The first data manager 18 can conclude that the request applies to the first

32   memory. The first data manager 18 can apply or execute the request on the TOC 36.

33   The first data manager 18 can generate a reply and send, shown by arrow 40, the reply

34   to the CM 16.

1   [0060] Figure 7 illustrates flows of various requests from the CM 16 and the replies to

2   the requests. The instructions are shown as create, update, delete and search/read

3   (i.e., query). The request from the CM 16 can be, respectively, create the first data

4   object, update the first data object, delete the first data object, and search/read the first

5   data object. (The numbers of the requests and replies illustrate an exemplary

6   chronological order.) The first data manager 18 can convert or otherwise translate the

7   request from the CM 16 to a first data manager instruction, such as add the first data

8   object, change the first data object, remove the first data object, and find the first data

9   object, respectively. The first data manager 18 can apply or execute the first data

10   manager instruction on the TOC 36. The first data manager 18 can then return a

11   reply. The replies can include the data searched, and/or confirmation that the task was

12   completed successfully, and/or an error code and or error explanation.

13   [0061] Figure 8 illustrates a method for accessing high access memory data in the

14   ROC 34 and in the database system 12. The CM 16 can send, shown by arrow 38, a

15   request to the first data manager 18. The request can apply to the high access second

16   memory data. The first data manager 18 can analyze the request 38. The first data

17   manager 18 can conclude that the request applies to the high access second memory

18   data. The first data manager 18 can determine whether the high access second

19   memory data is in the ROC 34. If the first data manager 34 determines that the high

20   access second memory data is in the ROC 34, the first data manager 18 can apply or

21   execute the request on the high access second data in the ROC 34. The first data

22   manager 18 can send the request to the second data manager 20. The second data

23   manager 20 can apply or execute the request on the high access second data in the

24   database system 12. The database system 12 and/or the second data manager 20

25   and/or the first data manager 18 can generate one or more replies. The replies can be

26   sent, shown by arrow 40, directly or via the first data manager 18 to the CM 16.

27   [0062] Figure 9 illustrates flows of various requests from the CM and the replies to

28   the requests. The exemplary instructions are shown as create, update, delete, simple

29   and complex searches/reads (i.e., query). The request from the CM 16 can be,

30   respectively, create the first data object, update the first data object, delete the first

31   data object, and search/read the first data object. (The numbers of the requests and

32   replies illustrate an exemplary chronological order.) The first data manager 20 can

33   convert or otherwise translate the request from the CM 16 to a first data manager

34   instruction, such as add the first data object, change the first data object, remove the

1   first data object, and find the first data object, respectively. The first data manager

2   can then apply or execute the translated request on the high access second data in the

3   ROC 34.

4   [0063] The first data manager 18 can send the request to the second data manager 20

5   and/or the CM 16 can send the request directly to the second data manager 20. The

6   second data manager 20 can convert or otherwise translate the request to a second

7   data manager instruction, such as insert the row of data, update the row of data, delete

8   the row of data, and select the row or rows of data, respectively (with no response

9   shown for a simple search/read, although the second data manager can perform simple

10   searching). The second data manager 20 can apply or execute the request on the high

11   access second data in the database system 12. The second data manager 20, and/or

12   the database system 12 and/or the first data manager 20 can then return a reply. The

13   replies can include the data searched, and/or confirmation that the task was completed

14   successfully, and/or an error code and or error explanation.

15   [0064] Figure 10 illustrates a method for accessing low access memory data in the

16   database system 12. The CM 16 can send, shown by arrow 38, a request to the first

17   data manager 18, and/or directly to the second data manager 18. The request can

18   apply to the high access second memory data. The first data manager 18 can analyze

19   the request 38. The first data manager 18 can conclude that the request applies to the

20   low access second memory data. The first data manager 18 can send the request to

21   the second data manager 20. The second data manager 20 can apply or execute the

22   request on the high access second data in the database system 12. The database

23   system 12 and/or the second data manager 20 can generate one or more replies. The

24   replies can be sent, shown by arrow 40, directly or via the first data manager 18 to the

25   CM 16.

26   [0065] Figure 11 illustrates flows of various requests from the CM and the replies to

27   the requests. The exemplary instructions are shown as create, update, delete, and

28   searches/read (i.e., query). The request from the CM 16 can be, respectively, create

29   the first data object, update the first data object, delete the first data object, and

30   search/read the first data object. (The numbers of the requests and replies illustrate an

31   exemplary chronological order.) The first data manager 18 can send the request to the

32   second data manager 20. The second data manager 20 can convert or otherwise

33   translate the request to a second data manager instruction, such as insert the row of

34   data, update the row of data, delete the row of data, and select the row or rows of data,

1   respectively. The second data manager 20 can apply or execute the request on the

2   high access second data in the database system 12. The second data manager 20,

3   and/or the database system 12 and/or the first data manager 20 can then return a reply.

4   The replies can include the data searched, and/or confirmation that the task was

5   completed successfully, and/or an error code and or error explanation.

6   [0066] The CM 16 can send requests directly to the desired data manager 18 or 20

7   and/or the CM 16 can tag the request and the first data manager 18 can analyze the tag

8   to determine whether to apply and/or execute the request and/or whether to send the

9   request to the second data manager. The tag can be the substance of the request (i.e.,

10   the requested action) and/or additional data solely to communicate the desired final

11   location of the request.

12   [0067] The first memory data objects of this category can be created, updated or

13   deleted in the high-speed access path of the revenue management system 100.

14   Examples of the first memory data objects include active session objects and resource

15   reservation objects.

16   [0068] The first memory data objects can be analyzed using, for example, logical

17   predicates (e.g., equals, not equals). Queries executed on first memory data can

18   specify an index to use to satisfy the query. The index can be a hash to enable fast

19   value lookup. The index can be a single column index. Predicates on other columns

20   can be supported by filtering the result set to find matches.

21   [0069] Requests for the first memory data objects can be passed to a standard heap

22   memory area. The requests can be created, changed and deleted within transactions.

23   [0070] The first memory data objects can be limited to particular object, such as

24   business object types.

25   [0071] The high access second memory data objects can be accessed only in a read-

26   only mode in the high-speed access path. An example of the high access second

27   memory data objects is customer account information.

28   [0072] The ROC 34 can be filled on demand. This means that requests can be

29   redirected to the database system 12 if the high access second memory data object is

30   not found in the ROC 34. If the request is a read of an entire object, the ROC 34 can

31   be filled or cached by the reply (e.g., as the reply passes through the first data manger

32   18 on the reply's route back to the CM 16 from the second data manager 20). Partial

33   object requests ('read_fields') of the high access second memory data objects can be

34   cached in a similar manner to that performed for the entire object.

1   [0073] The high access second memory data objects can be fully queried. Simple

2   queries involving basic logical operators (e.g., equals, not equals) can be performed

3   by the first data manager 18 on the high access second memory data objects in the

4   ROC 34. Complex queries (e.g., involving joins to other objects, or operators such as

5   'like' or 'in') can be performed by the second data manager 20 on the high access

6   second memory data objects in the database system 12.

7   [0074] The dynamic high access second memory data objects can be loaded by the

8   data migrator 30 after a failover.

9   [0075] The static high access second memory data objects can be loaded by the data

10   migrator 30 immediately after the backup first memory system has been started. The

11   static high access second memory data objects can be synchronized with the database

12   via the first memory synchronization system.

13   [0076] The low access second memory data objects can be absent from the first

14   memory. Requests for the low access second memory data objects can be routed

15   directly from first data manager 18 to the second data manager 20. The low access

16   second memory data objects can be fully queried.

17   [0077] The first data manager can allow reading of the first data values during a write

18   operation. The first data manager can have the write operation take place on a

19   scratchpad of data that is only visible to the writing transaction. The first data

20   manager can serialize the first data while the update is moved to main memory at the

21   commit time.

22   [0078] The first data manager can have a read committed isolation. The read

23   committed isolation makes all committed updates available to transactions even if the

24   commit takes place after the transaction is started. Read committed isolation can

25   prevent "dirty" reads (i.e., the first data manager preserves the earlier first data value

26   for reading during pending changes to the first data value).

27   [0079] The first data manager can support or not support statement or transaction

28   level consistent reads.

29   [0080] The revenue management system 100 can be configured to route any traffic

30   not related to session handling can be routed to and/or away from the first memory

31   data manager 18. A synchronization system can be used to send updates to the first

32   memory data manager 18. The synchronization system can automatically propagate

33   changes affecting objects stored in the first memory to all the first memory instances

34   caching the particular object or object type.

1   [0081] The revenue management system 100 can have a convergence system. The

2   convergence system can load batch data via the first data manager 18 into the revenue

3   management system 100, for example, to share any data of batch origin, such as

4   balances between prepaid and postpaid accounts.

5   [0082] The data capacity of a first memory instance can be lower than data the

6   capacity of a second memory (e.g., database) instance. One second memory instance

7   can support several shared-nothing instances of the first memory. (The commonly

8   used term is m:n (m – first memory instances / n – second memory instances)).

9   [0083] The first memory data manager 18 can reduce the latency for objects first

10 memory data manager 18 handles, and at the same time enabling increased throughput

11 of the system 100.

12 [0084] For installation of first memory data manager 18, the first memory data

13 manager 18 can be configured to be inserted between the CM 16 component and

14 second memory data manager 20 component. The introduction of the first memory

15 data manager 18 can change the access characteristics of some object types for a pre-

16 existing revenue management system that did not have the first memory data manager

17 18. Installation of the first memory data manager 18 can be configured to be

18 transparent (e.g., not change object types). The system 100 can be configured so that

19 the higher-level business logic architecture layers cannot tell first memory data

20 manager 18 is present. However, the business logic can be changed to utilize the first

21 memory data manager 18. These changes can be ignored by the system 100 if the

22 first memory data manager 18 is not present.

23 [0085] The first memory data manager 18 can be installed in an existing revenue

24 management system. For example, the first memory data manager 18 can be

25 physically installed (e.g., mounting hardware and/or loading software onto the

26 appropriate computer-readable medium) and the base software can be configured.

27 [0086] After the installation of the base software, the first data can then be migrated

28 into the first memory data manager 18. The first memory data object residencies

29 stored in the data dictionary can take effect, loading the first data onto the first data

30 manager 18 during use. The residencies can be part of the default business logic

31 module 10 installation (having no effect when the first memory data manager 18 is

32 not present) or can be loaded onto the business logic module 10 during the installation

33 of the first memory data manager 18.

1 [0087] Reference objects can be migrated by loading into the first data manager 18

2 when accessed for the first time and/or pushed into the first memory data manager 18

3 by the data migratory 30.

4 [0088] Data objects can be redefined as first memory data objects (or low access or

5 high access second memory data objects) by deploying the data object via the normal

6 mechanism, and then updating the residency type in the data dictionary.

7 [0089] After a process startup, the first memory data manager 18 can have an empty

8 ROC 34. A separate data migration thread can push all high access second memory

9 data in the ROC 34. The static high access second memory data objects can be loaded

10 into the ROC 34. For example, a first memory data manager 18 in backup mode can

11 load only the static high access second memory data objects into the ROC 34.

12 [0090] The data migrator 30 can provide a notification hook to signal the end of the

13 migration and/or startup phase to other processes. The first memory data manager 18

14 can be operational immediately after start (e.g., before the migratory tool sends the

15 notification hook), for example, with an empty cache. The first request after startup

16 for a specific first data object can trigger that first data object to be loaded into the

17 cache (e.g., if the data migrator 30 has not yet loaded the desired first data object

18 already).

19 [0091] Upon a system shutdown, the first data manager 18 can close the process log

20 file, and release used memory. The high access second memory data objects can be

21 unaffected by shutdown (e.g., remaining stored on the second memory).

22 [0092] The revenue management system 100 can provide hooks to verify and monitor

23 performance. The revenue management system 100 can log performance data on a

24 regular basis and/or make performance data available via an embedded web server. A

25 signal can be sent to the second memory data manager to collect desired data (e.g., for

26 some parts of the system).

27 [0093] The revenue management system 100 can create system logs that can monitor

28 operation of the revenue management system 100. A log monitoring GUI (e.g.,

29 Pipeline log viewer) can be used. Business logic style pin-logging can, for example,

30 aid debugging and diagnosis.

31 [0094] The first memory data manager 18 can have a pipeline framework tracing

32 model. Additional trace information can be collected from subsystems of the revenue

33 management system 100 on a case by case basis.

1   **[0095]** Figures 12a, 12b and 12c illustrate that the revenue management system 100

2   can be scalable to large scale expansion.

3   **[0096]** The revenue management system 100 can have multiple second memory

4   locations (e.g., databases 24a, 24b and 24c). The revenue management system 100

5   can have separate instances of the first memory data manager 18a-18i, and the second

6   memory data manager 20a-20l. Pairs of sets of first memory data managers and

7   second memory data manager, for example, 18a, 18b, 20a, and 20b can be formed

8   into high availability (HA) pairs 42. The HA pairs 42 can have active and backup

9   first data managers 18a and 18b, respectively, for example, and active and backup

10   second data managers 20a and 20b, respectively, for example.

11   **[0097]** The revenue management system 100 can have a capacity partitioning scheme.

12   **[0098]** Each second memory (e.g., database 24) instance can be associated (i.e., in

13   communication) with one or more first memory data manager 18 instances. The

14   revenue management system can be configured so no data is stored in overlapping

15   second memory instances (n Timos instances : 1 database). The business logic

16   module can have several, independent databases (m). The combination of

17   TIMOS/databases can be referred to as $m:n$ configuration.

18   **[0099]** The CM 16a-16e can lookup in the directory server 32 to identify the first

19   memory data manager 18 and second memory data manager 20 (or database 24)

20   combination applicable for a certain object.

21   **[0100]** The revenue management system 100 can have account migration tools. The

22   account migration tools can move subscriber data from one first and/or second

23   memory location (e.g., database 24 and/or first memory data manager 18) to another

24   first and/or second memory location.

25   **[0101]** Multiple second memory data managers 20 can communicate with the same

26   database clusters 22. All the databases can be managed with one database cluster 22

27   (e.g., one RAC cluster) (not shown).

28   **[0102]** The SCPs 14, gateways 8 and CMs 16 can each be associated to multiple first

29   memory data managers 18. The gateway 8 can support load balancing over several

30   CMs 16. The CMs 16 can use the directory server 32 to route the requests to the

31   correct first memory data managers 18.

32   **[0103]** The database system 12 can run multiple database schemes in one RAC cluster

33   22. The revenue management system 100 can be configured to associate a dedicated

34   set of resources to just one SCP 14 or group of SCPs 14.

1   [0104] Figure 12b illustrates that the first memory data managers 18c and 18d can

2   communicate directly with the database system 12, for example with the database

3   clusters 22a and 22b.

4   [0105] Figure 12c illustrates that the revenue management system can have two or

5   more database systems 12a and 12b.

6   [0106] Figure 13 illustrates a failure protection scheme with a high availability

7   monitor 44 regularly checking the status of the control manager, the first memory in

8   the first memory data manager 18, the second memory data manager 20, the database

9   cluster , and the database 24.

10   [0107] Figure 14 illustrates a self-contained failure protection system. Each

11   component of the revenue management system 100 can check on the status of its

12   immediately downstream component (e.g., the second memory data manager 20 is

13   immediately downstream of the first memory data manager 18). If the immediately

14   downstream component has failed, or is sending a failure message regarding a further

15   downstream component failure, the revenue management system can take appropriate

16   action, including alerting a user that a failure has occurred. The revenue management

17   system 100 can be absent of a separate monitor component checking for system

18   failures.

19   [0108] The revenue management system 100 can have a high availability. The

20   revenue management system 100 can have a warm standby operation by referring to

21   any data remaining in the first memory (e.g., TIMOS). During warm standby, in the

22   case of a loss of data (e.g., during a system failure), the revenue management system

23   100 can recreate data from the switch 6 and/or the first memory when the switch 6

24   sends re-authorization data (e.g., during long calls) or end-of-call data.

25   [0109] The database system 24 can store the latest static high access second memory

26   data before a loss of data. The static high access second memory data can be

27   recovered to the first memory from the database system 24 after a loss of data in the

28   first memory.

29   [0110] In the revenue management system 100, higher layers (i.e., more stable during

30   a system failure, such as the database) of architecture with very high availabilities can

31   partially or completely backup lower layers (i.e., less stable during a system failure,

32   such as a solid state RAM variation of the first memory) with lesser availabilities in

33   case of failures.

1   [0111] The revenue management system 100 can have spare, unused hardware and

2   software such as backup data managers in the high availability pairs 42, as shown in

3   Figures 12a through 12c. The high availability pair 42 can have active and a backup

4   first data managers 18 and active and backup second data managers 20. The backup

5   data managers can copy from the respective active data managers, for example during

6   a period of no other activity with the active data manager and/or from a sketchpad,

7   and/or the last available data from the active data manager. In case of failure of an

8   element, the backup or other inactive elements will be able to restore data and/or take

9   over the additional load.

10   [0112] U.S. Patent Application Nos. 10/394,409 filed 21 March 2003, and 11/478,558

11   filed 28 June 2006 are incorporated by reference herein in their entireties.

12   [0113] Accessing can include querying, updating, creating, deleting and combinations

13   thereof. Querying, updating, creating, and deleting for any data can be interchanged

14   with each other as disclosed.

15   [0114] It is apparent to one skilled in the art that various changes and modifications

16   can be made to this disclosure, and equivalents employed, without departing from the

17   spirit and scope of the invention. System and architecture are used as interchangeable

18   terms, both referring to one or more hardware and software components in

19   communication. All elements shown herein can be software and/or hardware

20   components. Elements shown with any embodiment are exemplary for the specific

21   embodiment and can be used on other embodiments within this disclosure.

1   CLAIMS

2   We claim:

3   1. A computer-based telecommunications network account management system

4   comprising:

5   a first memory having a first memory access speed,

6   a second memory having a second memory access speed,

7   a first memory manager configured to route a first request for a first object to

8   the first memory, and configured to route a second request for a second object not to

9   the first memory,

10   wherein the first memory has a faster access speed than the second memory.

11

12   2. The system of Claim 1, further comprising a second memory manager configured

13   to route the second object to the second memory.

14

15   3. The system of Claim 1, wherein the second memory comprises a hard drive.

16

17   4. The system of Claim 1, wherein the first memory comprises solid state memory.

18

19   5. The system of Claim 1, wherein the first memory comprises RAM.

20

21   6. The system of Claim 1, wherein the second memory comprises a relational

22   database.

23

24   7. The system of Claim 1, further comprising a database cluster.

25

26   8. The system of Claim 1, further comprising a gateway configured to interface with

27   a telecommunications network.

28

29   9. The system of Claim 1, wherein the first request comprises a query.

30

31   10. The system of Claim 9, wherein the second request comprises a query.

32

33   11. The system of Claim 1, wherein the first request comprises a request to write.

34

1   12. The system of Claim 11, wherein the second request comprises a request to write.

2

3   13. A computer-based telecommunications network account management system

4   comprising:

5   a first in-memory object store, wherein the in-memory object store comprises

6   solid state memory, and wherein the memory is configured to store data, and

7   a first hard drive configured to store a database.

8

9   14. The system of Claim 13, further comprising a second in-memory object store,

10   wherein the second in-memory object store is configured to substantially backup the

11   data on the first in-memory object store.

12

13   15. The system of Claim 13, further comprising a second hard drive configured to

14   substantially backup the database stored on the first hard drive.

15

16   16. A method of tracking telecommunications network use, using an account

17   management system comprising a first memory and a second memory wherein the

18   first memory has a first access speed and the second memory has a second access

19   speed and the first access speed is faster than the second access speed, the method

20   comprising:

21   requesting a first action in the first memory.

22

23   17. The method of Claim 16, wherein requesting comprises sending a request to a

24   first memory data manager, and wherein the first memory data manager analyzes the

25   request.

26

27   18. The method of Claim 17, wherein requesting further comprises routing the

28   request by the first memory data manager after the first data manager analyzes the

29   request.

30

31   19. The method of Claim 18, wherein routing comprises performing the request.

32

33   20. The method of Claim 19, wherein routing comprises sending the request to a

34   second data manager.

1

2   21. The method of Claim 16, wherein the first action comprises querying.

3

4   22. The method of Claim 16, wherein the first action comprises writing.

5

6   23. The method of Claim 16, further comprising requesting a second action in the

7   second memory.

8

9   24. The method of Claim 16, wherein the second action comprises querying.

10

11   25. The method of Claim 16, wherein the second action comprises writing.

12

13   26. A method of tracking telecommunications network use, using an account

14   management system comprising a first memory and a second memory wherein the

15   first memory has a first access speed and the second memory has a second access

16   speed and the first access speed is faster than the second access speed, the method

17   comprising:

18   storing first data on the first memory;

19   storing second data on the second memory; and

20   backing up the first data on a first backup memory.

21

22   27. The method of Claim 26, further comprising; backing up the second data on a

23   second backup memory.

24

25   28. A method of tracking telecommunications network use, using an account

26   management system comprising:

27   storing a first data in a first memory location, wherein the first memory

28   location is subject to memory loss during a system failure;

29   storing a second data in the first memory location;

30   storing the first data in a second memory location, wherein the first data can

31   be recovered from the second memory location after a system failure;

32   recovering the first data from the second memory location after a system

33   failure.

34

1   29. The method of Claim 28, further comprising not storing the second data in the

2   second memory location.

3

4   30. A method of tracking telecommunications network use, using an account

5   management system comprising a first system component, a second system

6   component, and a third system component, the method comprising:

7   a first checking for a failure, wherein the first checking for a failure is

8   performed by a first system component on a second system component; and

9   a second checking for a failure, wherein the second checking for a failure is

10   performed by the second system component on a third system component.

11

12   31. The method of Claim 30, wherein the first system component comprises a

13   gateway.

14

15   32. The method of Claim 31, wherein the second system component comprises a first

16   data manager.

17

18   33. The method of Claim 32, wherein the third system component comprises a

19   second data manager.
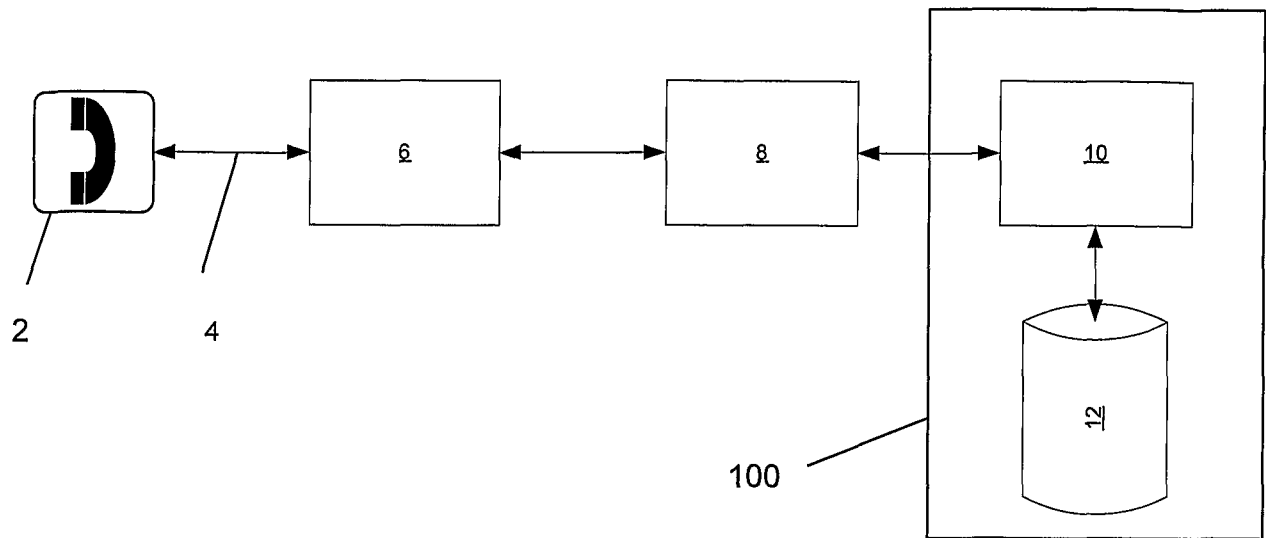
Figure 1
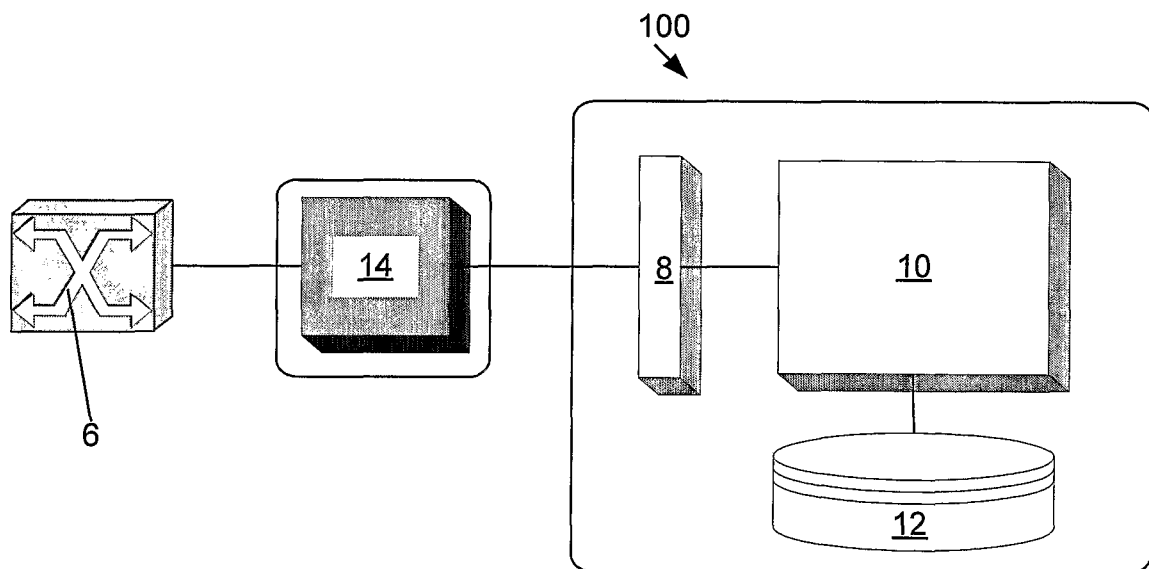
Figure 2

Figure 3

Figure 4a

10

100

Figure 4b

10

100

102

Figure 5

10

Figure 6

100

Figure 7

10

Figure 8

100
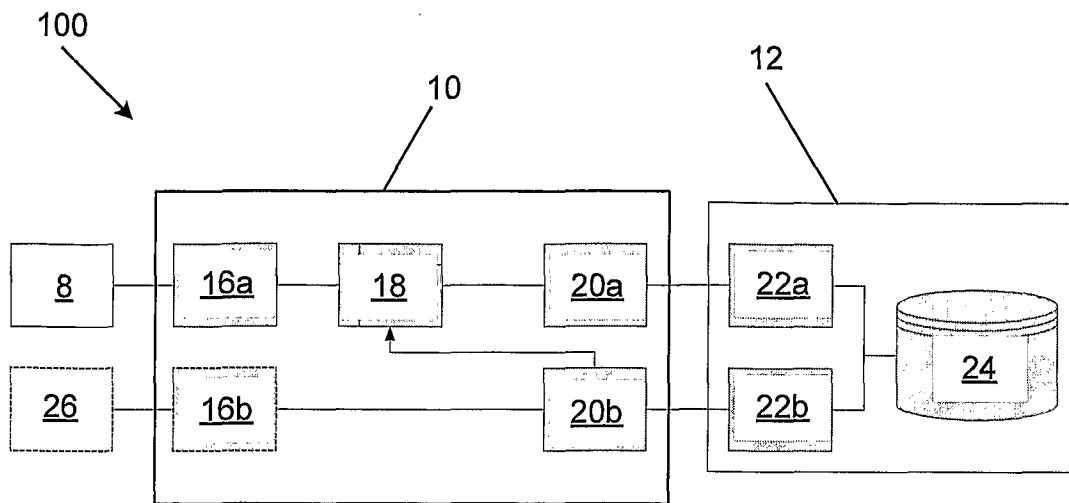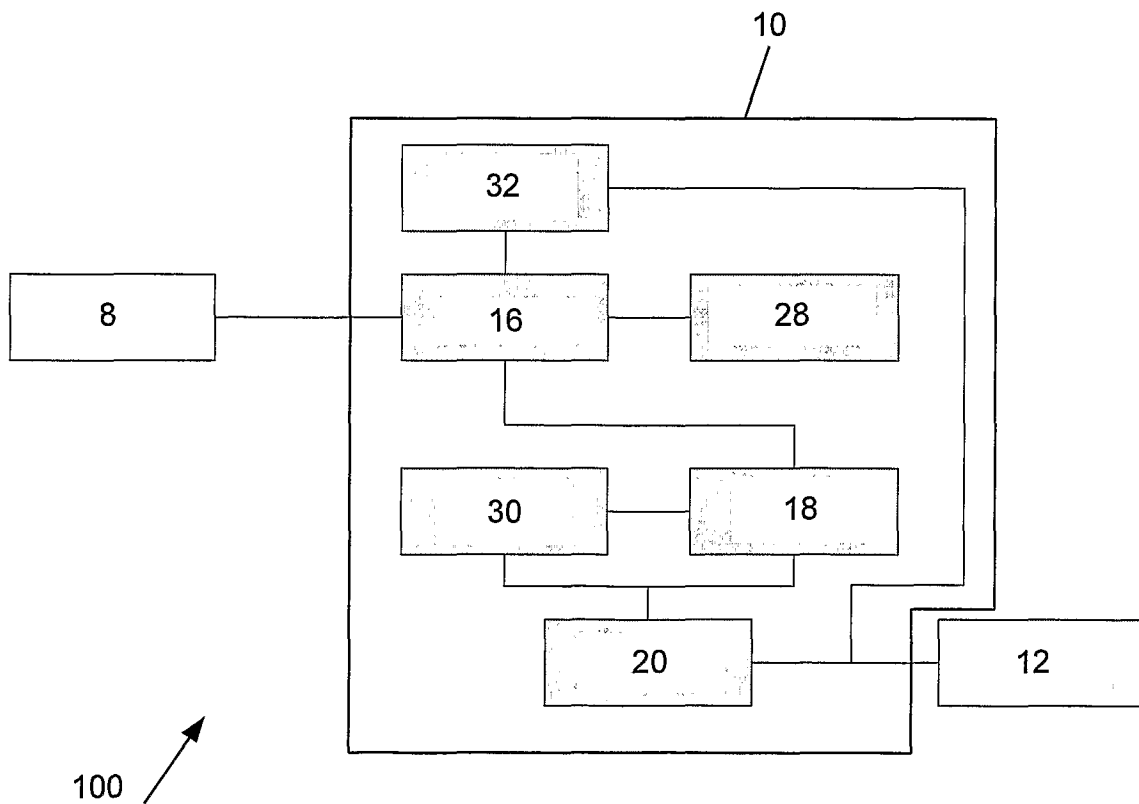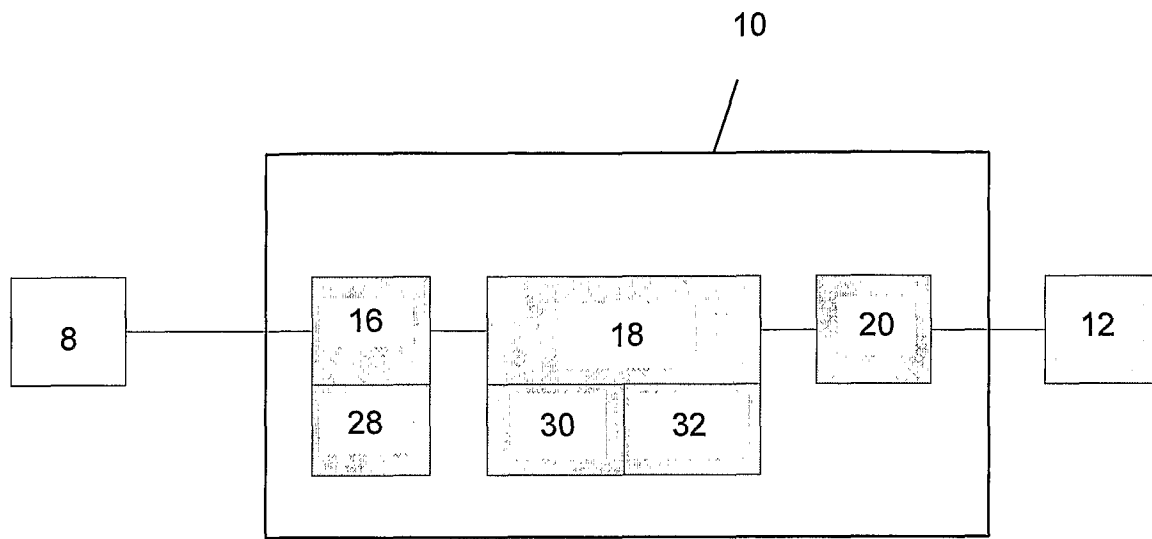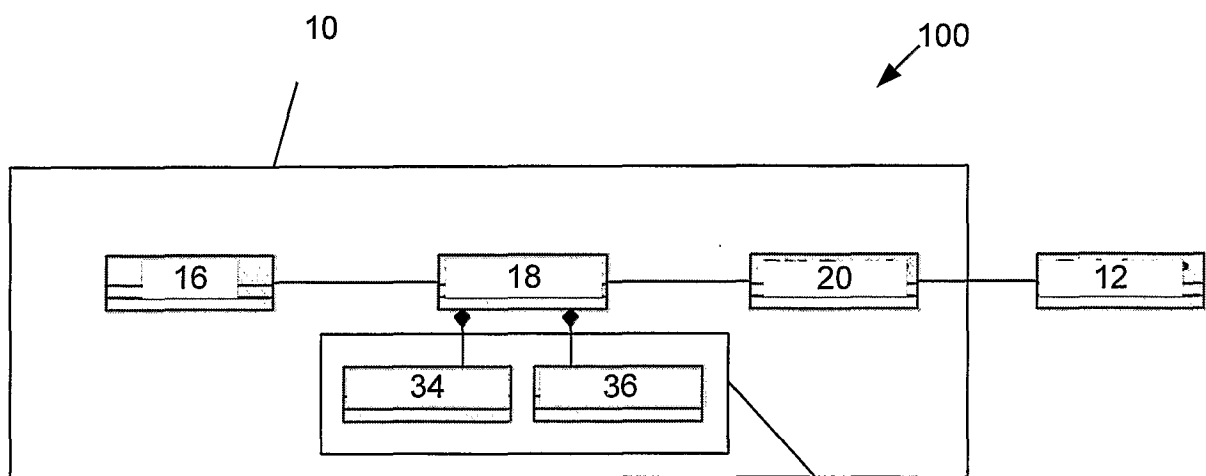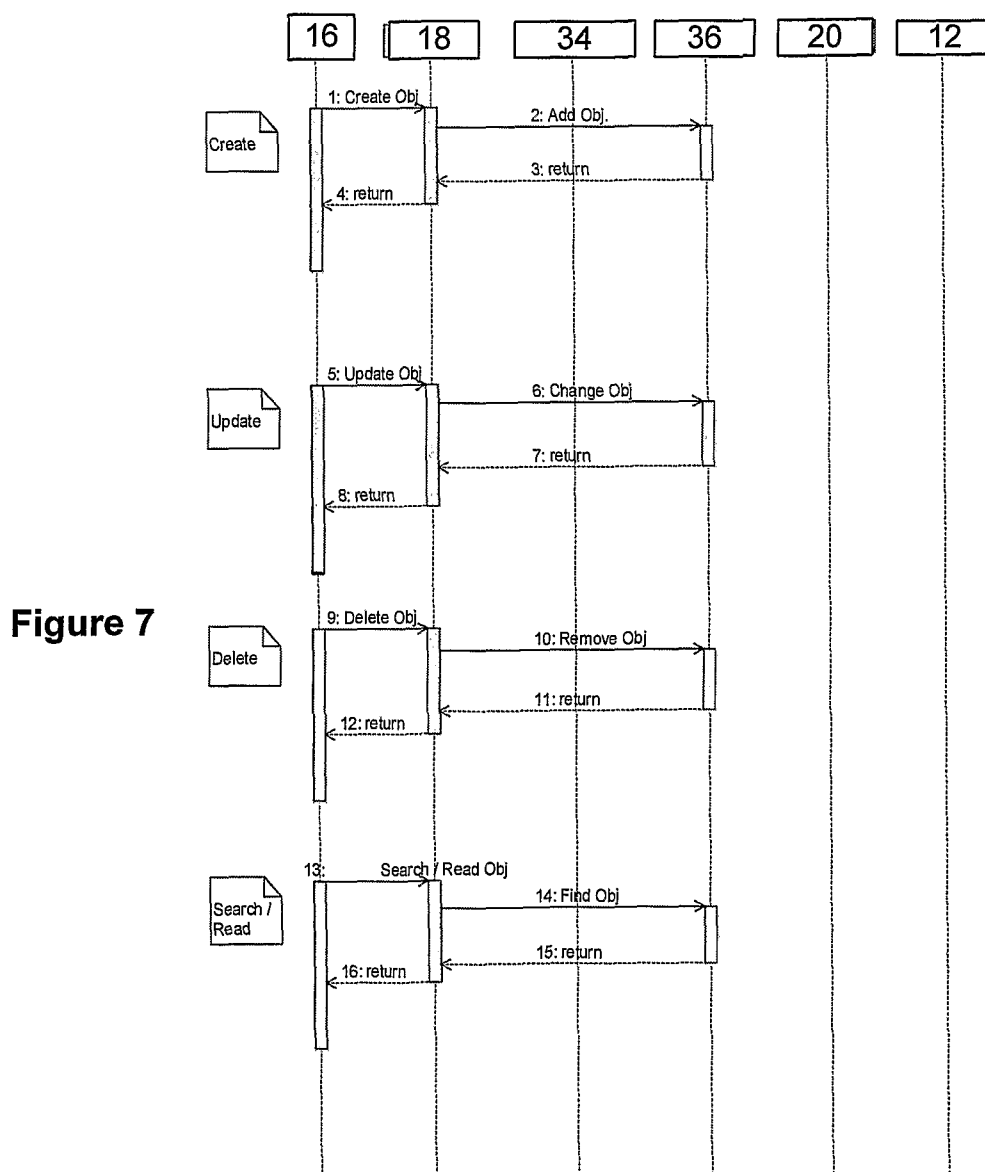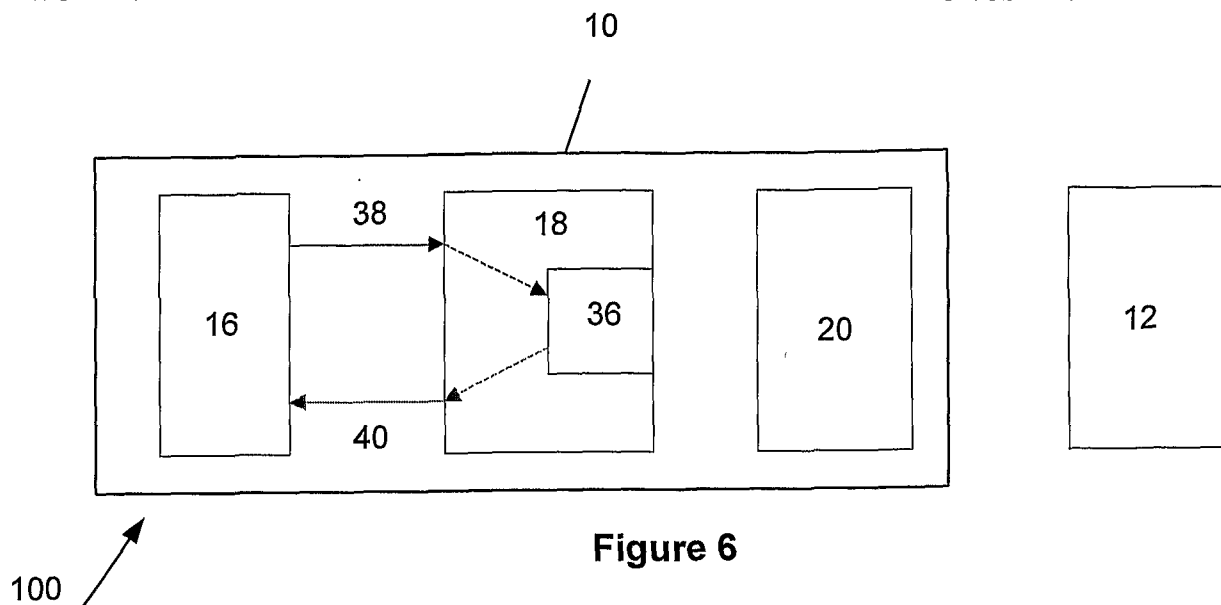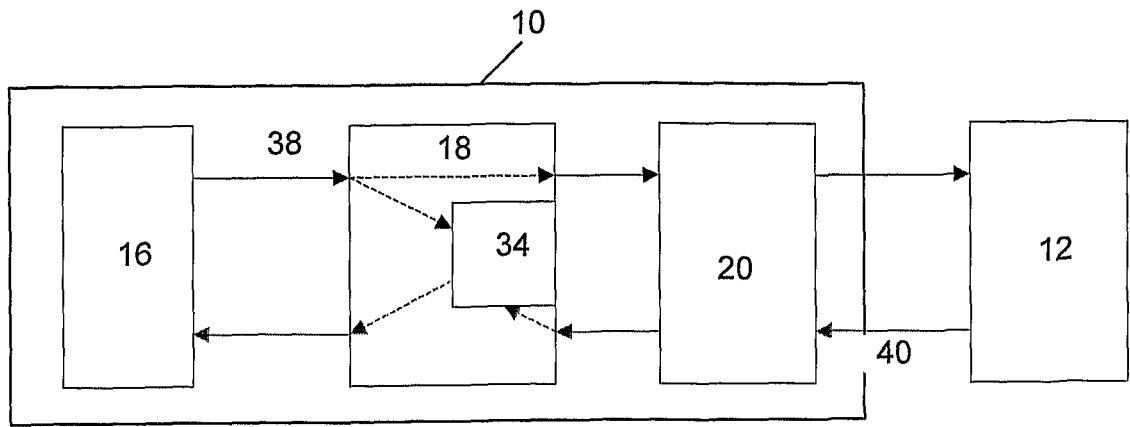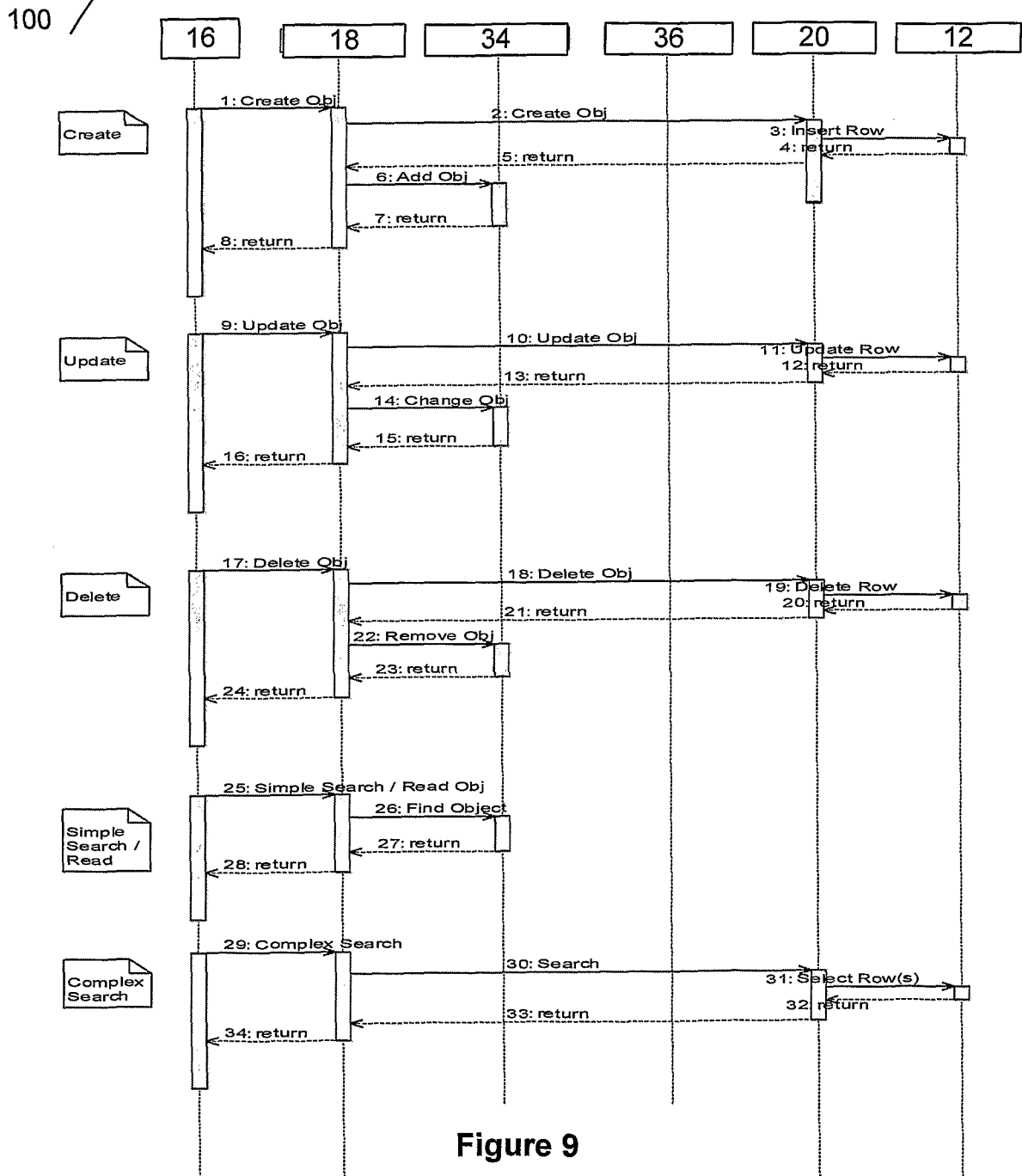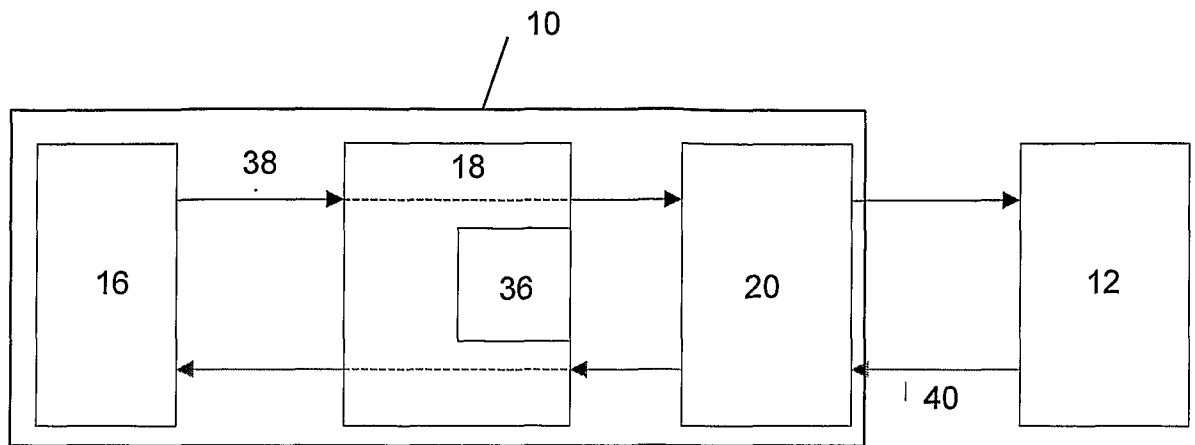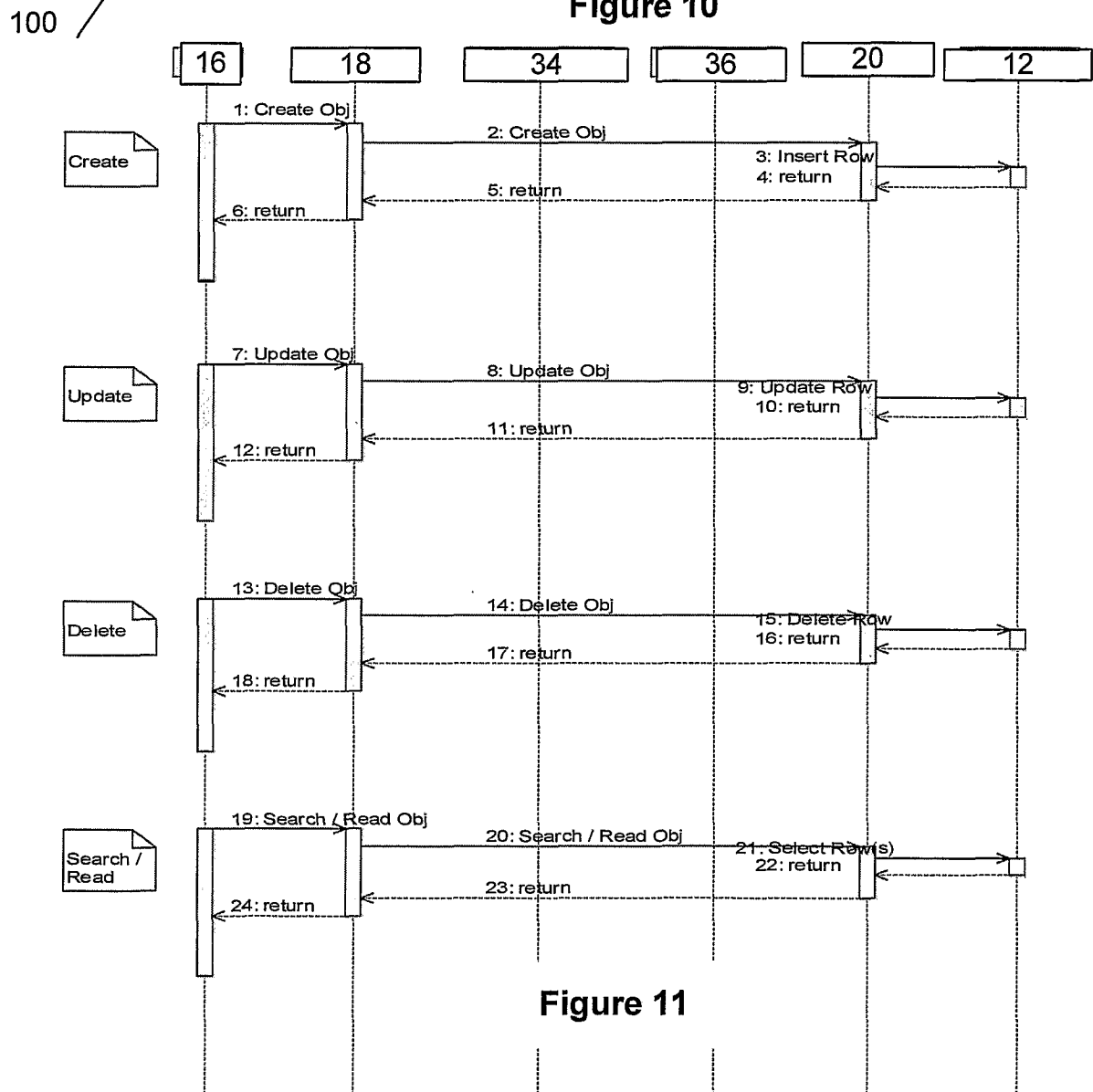
Figure 9

Figure 10

Figure 11
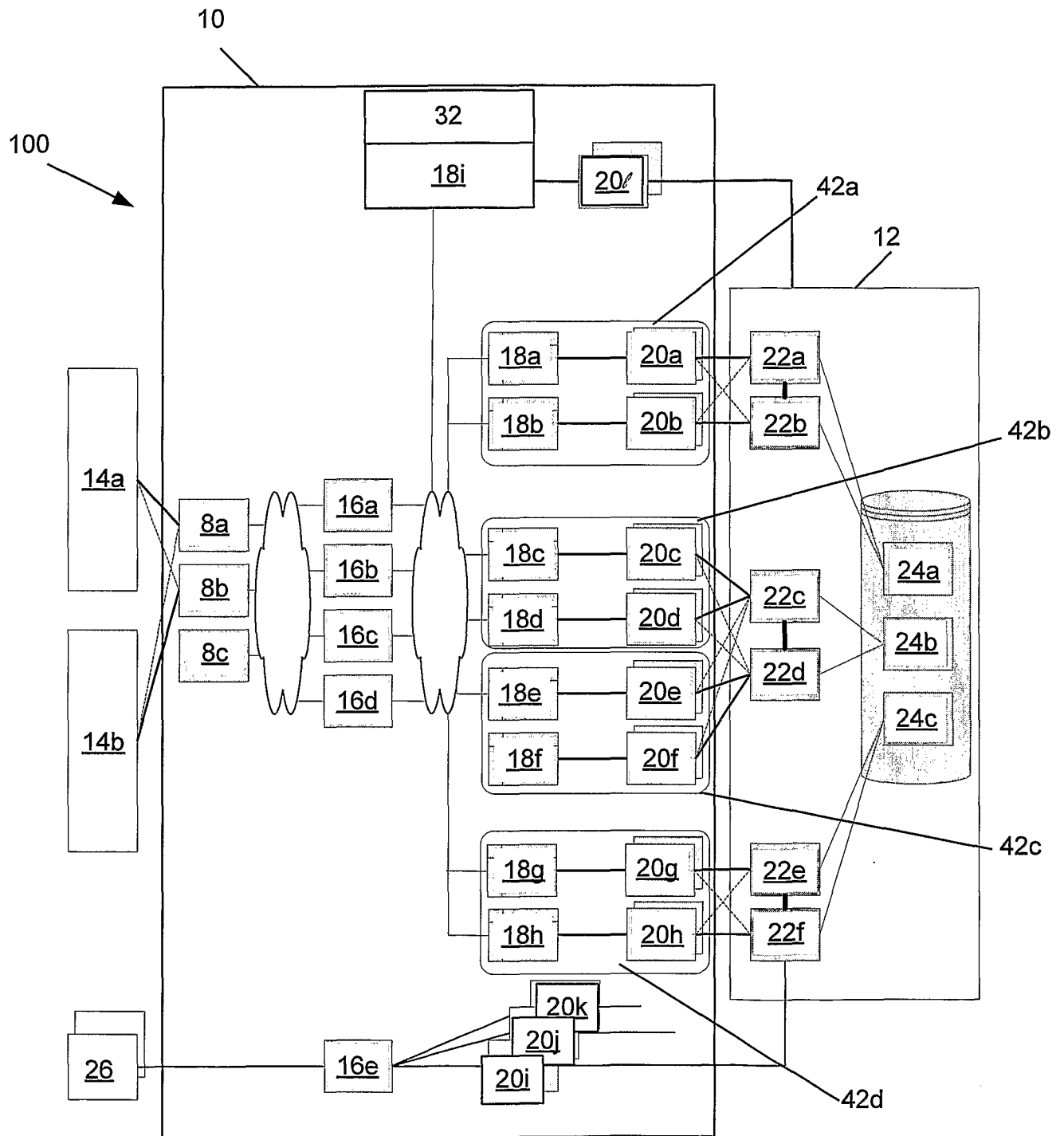
Figure 12a
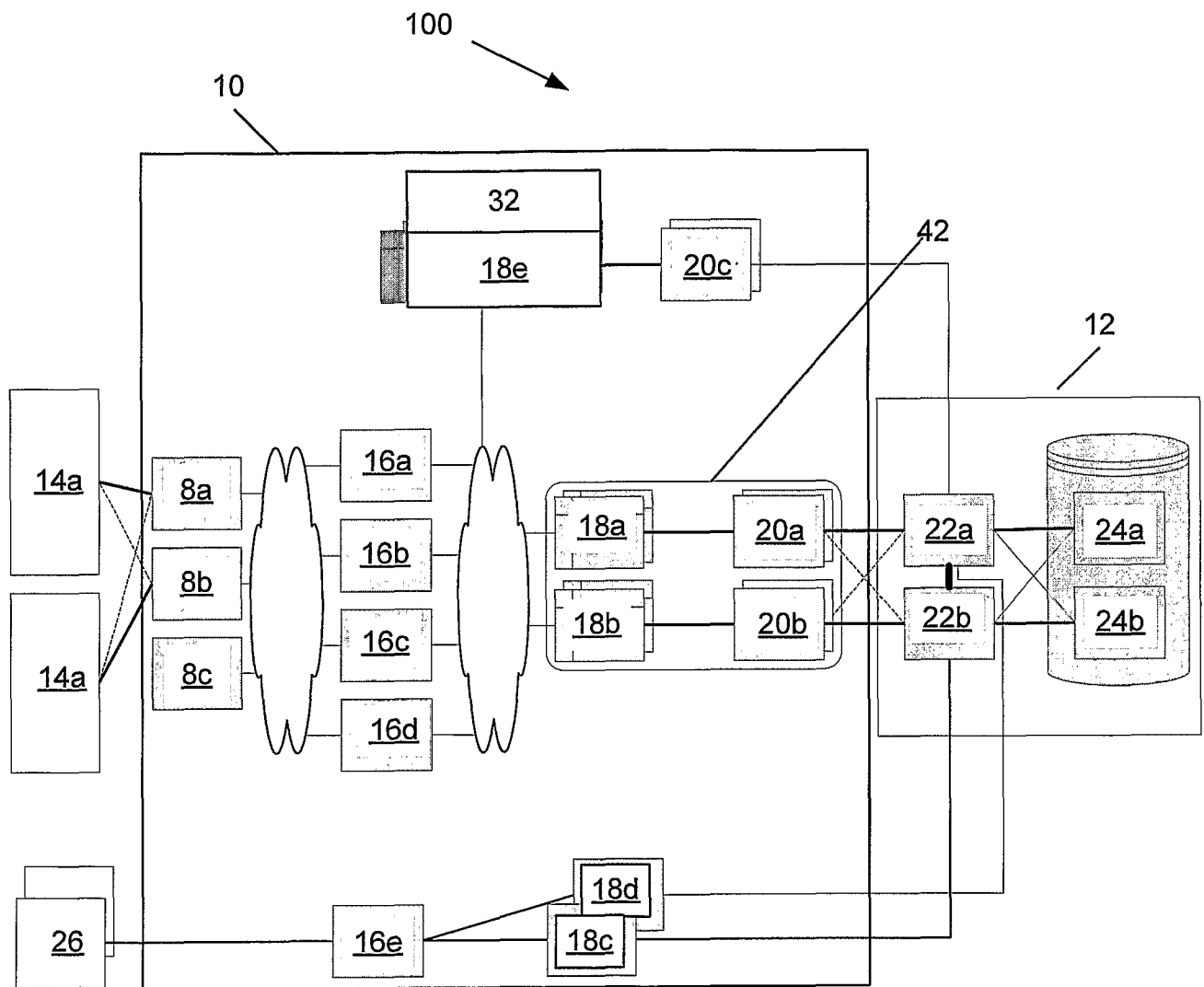
Figure 12b
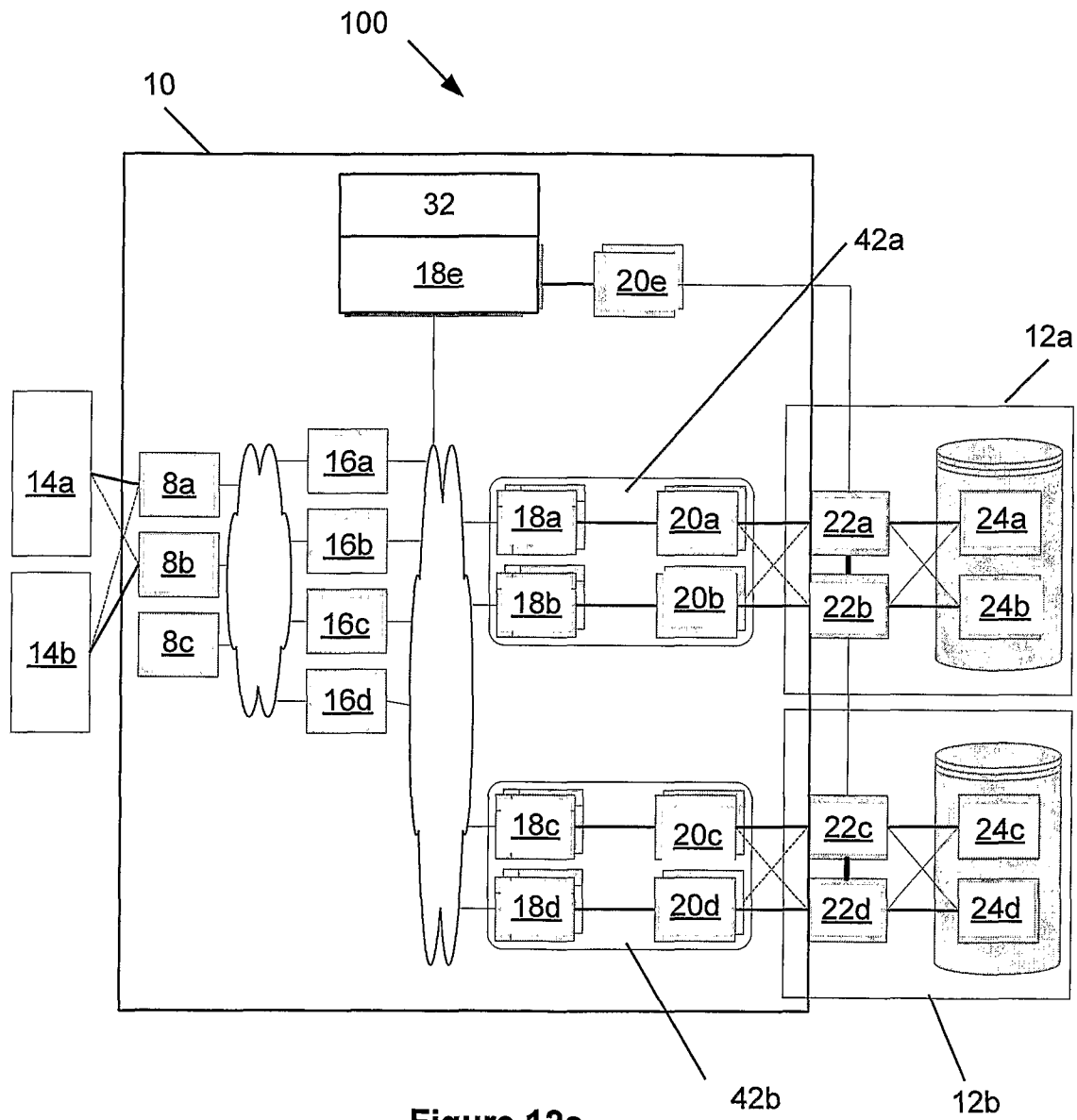
Figure 12c

Figure 13

Figure 14