Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2457706 A1 2003/02/27

(21) 2 457 706

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2002/08/15
(87) Date publication PCT/PCT Publication Date: 2003/02/27
(85) Entrée phase nationale/National Entry: 2004/02/13
(86) N° demande PCT/PCT Application No.: US 2002/026035
(87) N° publication PCT/PCT Publication No.: 2003/017170
(30) Priorité/Priority: 2001/08/15 (60/312,737) US

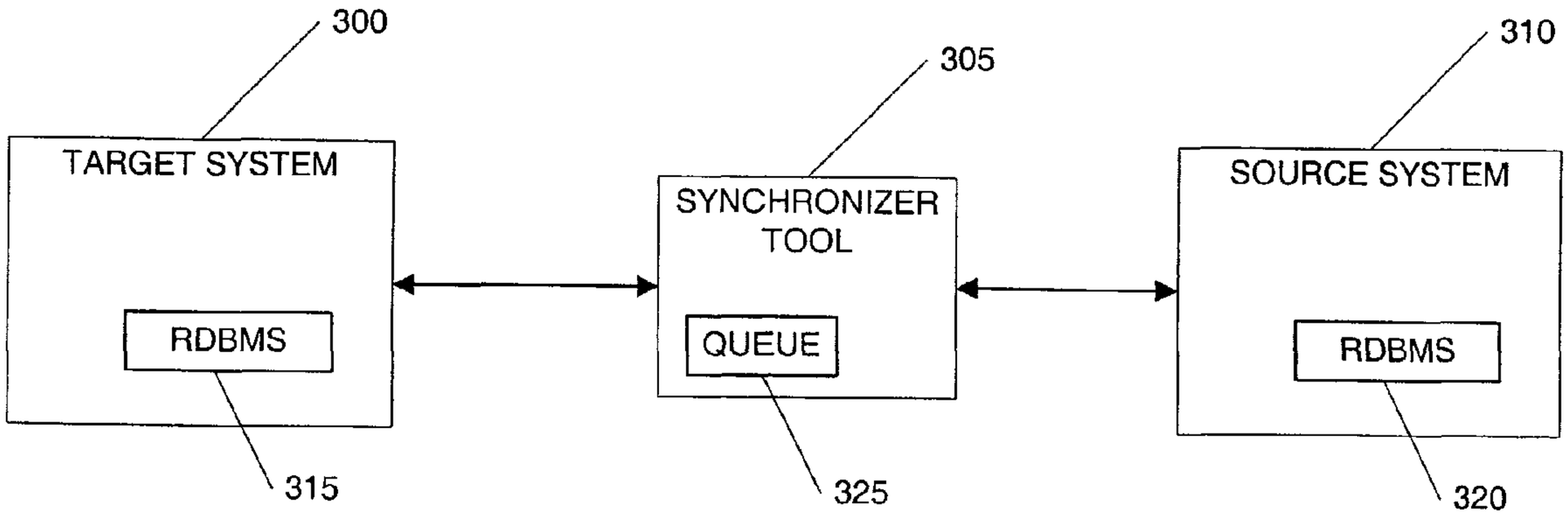
(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/60

(71) Demandeur/Applicant:
QUALCOMM INCORPORATED, US

(72) Inventeurs/Inventors:
HOREL, JERRY, CA;
TRUITT, ROBERT, US;
YU, JULIE, US

(74) Agent: SMART & BIGGAR

(54) Titre : INTERFACE DE SYNCHRONISATION DE DONNEES
(54) Title: DATA SYNCHRONIZATION INTERFACE



(57) Abrégé/Abstract:
Systems and methods consistent with the present invention provide an interface to facilitate the communication between systems. A source system (310) translates data into an XML format and transmits the data to a synchronizer interface tool. The synchronizer tool stores the data in a persistent intermediate storage, such as a queue (325), allowing the source system to go offline or perform other tasks while the data is transmitted to the target system. The synchronizer tool initiates the transmission of the data to the target system. If the target system indicates that it received the data, then the synchronizer tool sends an acknowledgement to the source system indicating the data was received. If the target system does not receive the transmission, the synchronizer tool maintains the data in the present intermediate storage andreinitiates transmission of the data to the target system (300).

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 February 2003 (27.02.2003)

PCT

(10) International Publication Number
WO 03/017170 A1

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US02/26035

(22) International Filing Date: 15 August 2002 (15.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/312,737 15 August 2001 (15.08.2001) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121 (US).

(72) Inventors: **HOREL, Jerry**; 6500 Torin Road, Brentwood Bay, British Columbia V8M2H5 (CA). **TRUITT, Robert**; 10198 Wateridge Circle #158, San Diego, CA 92121 (US). **YU, Julie**; 48700 Algonquin Court, San Diego, CA 92130 (US).

(74) Agents: **WADSWORTH, Philip R.** et al.; QUALCOMM Incorporated, 5775 Morehouse Drive, San Diego, CA 92121 (US).

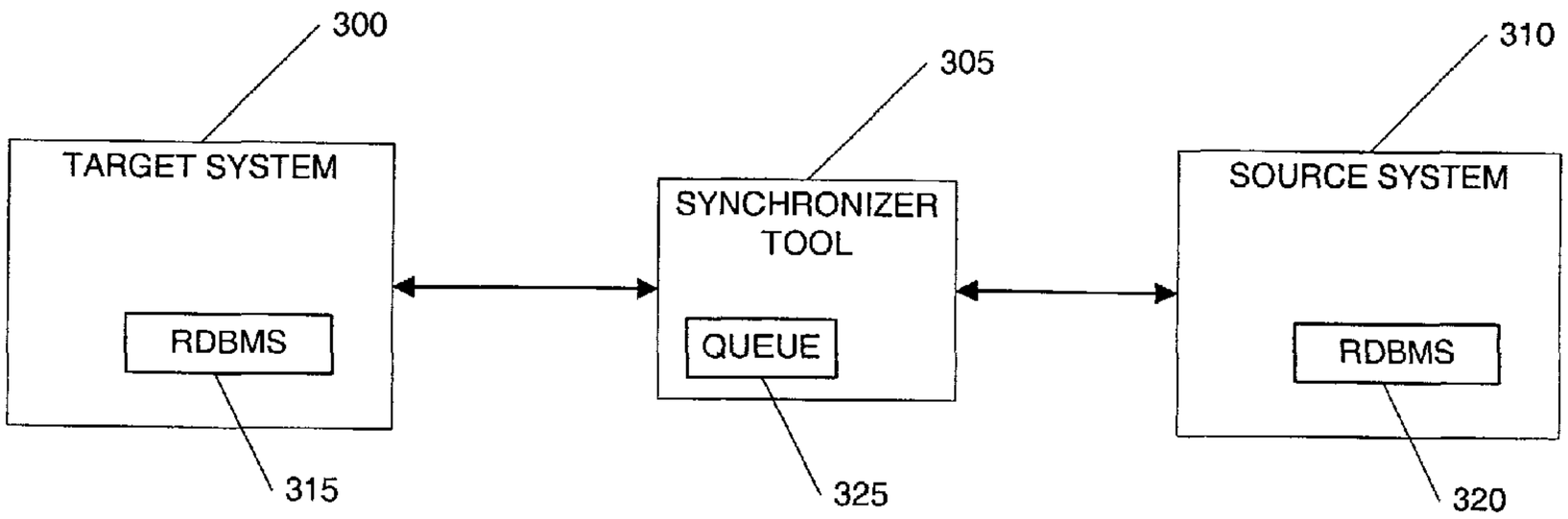
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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

(54) Title: DATA SYNCHRONIZATION INTERFACE



(57) Abstract: Systems and methods consistent with the present invention provide an interface to facilitate the communication between systems. A source system (310) translates data into an XML format and transmits the data to a synchronizer interface tool. The synchronizer tool stores the data in a persistent intermediate storage, such as a queue (325), allowing the source system to go offline or perform other tasks while the data is transmitted to the target system. The synchronizer tool initiates the transmission of the data to the target system. If the target system indicates that it received the data, then the synchronizer tool sends an acknowledgement to the source system indicating the data was received. If the target system does not receive the transmission, the synchronizer tool maintains the data in the present intermediate storage andreinitiates transmission of the data to the target system (300).

WO 03/017170 A1

DATA SYNCHRONIZATION INTERFACE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of Provisional Application No. 60/312,737, filed August 15, 2001, pending, which application is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

I. Field of the Invention

[0002] The present invention generally relates to communication between systems. More particularly, the invention relates to a communication interface to aid transferring of data and information between systems and/or databases.

II. Description of the Related Art

[0003] Wireless devices, such as cellular telephones, personal digital assistants ("PDAs"), pagers, laptops with wireless connectivity, etc., communicate packets including voice and data over a wireless network. These wireless devices have installed application programming interfaces ("APIs") onto their local computer platform that allow software developers to create software applications that operate on the wireless device. The API sits between the wireless device system software and the software application, making the wireless device functionality available to the application without requiring the software developer to have the specific wireless device system source code.

[0004] These wireless devices typically need to communicate with other systems and databases within the other systems. Unfortunately, the wireless device may lose a signal during communication or be otherwise unavailable when transmitting information to other systems. This may cause errors when attempting to access a database when the signal is lost. The wireless device may be required to reinitiate the database access and resubmit the database request when the signal is reacquired.

[0005] Extended beyond wireless devices, often wire-based systems need to communicate with each other but do not share a similar "language" for communication. For example, as with wireless devices, one system may need to communicate with the database in another system to receive or insert data. To communicate with the database, the system must be aware of the database language, record and field structures, and

formats in order to access and store information in the database. While current technologies provide for the ability for the interface between the system and the database to include the language, structure and format of the database, this becomes more complex when multiple databases, possibly requiring multiple unique database languages, etc., need to be accessed.

[0006] Furthermore, when data is to be sent to multiple systems, or conversely received from multiple systems, a common interface does not exist to integrate across the multiple systems to simplify the data transmission. This is problematic for systems communicating with several other systems.

[0007] Therefore, what is needed in the art is an interface that simplifies the communication between one or multiple databases and provides reliable and secure transfer of information between multiple systems.

[0008] Current methods in the art do not address this need. Database replication services and custom built database interfaces can become very complex and unwieldy if multiple databases need to be accessed. In addition, all systems that access the custom database must have that interface. Also, custom built databases are required to stay online for transactions to occur.

[0009] Electronic Data Interchange (EDI) only addresses the need for pre-defined message types and content specific to electronic commerce. The EDI message formats do not address the data exchange needs as it relates to wireless services, system integration considerations and billing specificity.

SUMMARY OF THE INVENTION

[0010] Systems and methods consistent with the present invention overcome the shortcomings of existing systems by allowing an application access to a device's resources based on a set of permissions associated with the application.

[0011] In one embodiment, the present invention provides a method of processing information in an interface comprising receiving data in a first format from a source system, storing the data in a persistent intermediate storage, initiating the transmission of the information to the target system while retaining the information, receiving an indication that the transmission of information to the target system was unsuccessful, and reinitiating the transmission of the information to the target system. The method may further include storing data received from multiple source systems associated with

multiple target systems, and initiating the transmission of the data from multiple source systems to the associated multiple target systems. Also, the method may include receiving an source system acknowledgement indicating the acknowledgement was received and removing the data from the persistent intermediate storage.

[0012] Other objects, advantages, and features of the present invention will become apparent after review of the hereinafter set forth Brief Description of the Drawings, Detailed Description of the Invention, and the Claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention. In the drawings:

[0014] Figure 1 is a representative diagram of a wireless network and the computer hardware and wireless devices that can be used in an exemplary embodiment of the present invention;

[0015] Figure 2 is a block diagram of the hardware components of the wireless network providing communication between different wireless devices, an application download server, and a database in an exemplary embodiment of the present invention;

[0016] Figure 3 is a block diagram detecting the architecture of the synchronizer tool interface in an exemplary embodiment of the present invention; and

[0017] Figure 4 is a flowchart depicting the process of the synchronizer tool interface in an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0018] Reference will now be made in detail to the presently exemplary and preferred embodiments of the invention as illustrated in the accompanying drawings, in which like reference characters designate like or corresponding parts throughout the several drawings. The nature, objectives and advantages of the present invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.

[0019] Figure 1 illustrates an architecture in which one embodiment of the present invention may be implemented using one or more wireless devices, such as cellular telephone **12**, in communication across a wireless network **14** with at least one network server, such as application download server **16**, that selectively downloads or provided access to software applications or other data to the wireless devices across a wireless communication portal or other data access to the wireless network **14**. As shown here, the wireless device can be a cellular telephone **12**, with a graphics display **13**, a personal digital assistant **18** with PDA screen **19**, a pager **20** with a graphics display **21**, which is shown here as a two-way text pager, or even a separate computer platform **22** that has a wireless communication portal and a display **23**, and may otherwise have a wired connection **24** to a network or the Internet. The system **10** can include any form of remote computer module including a wireless communication portal, including without limitation, wireless modems, PCMCIA cards, access terminals, personal computers, access terminals, telephones without a display or keypad, or any combination or sub-combination thereof.

[0020] The application download server **16** is shown here on a local server-side network **26** with other computer elements in communication with the wireless network **14**, such as a database **28** with stored applications and data that contains software applications and data that are accessible and downloadable to the wireless devices **12**,**18**,**20**,**22**.

[0021] Figure 2 is a block diagram of the hardware components of the wireless network providing communication between different wireless devices, an application download server, and a database in an exemplary embodiment of the present invention. The wireless network **14** is merely exemplary and can include any system whereby remote modules, such as wireless devices **12**,**18**,**20**,**22**, communicate over-the-air between and among each other and/or between and among components of a wireless network **14**, including, without limitation, wireless network carriers and/or servers. The application download server ("ADS") **16** and the stored applications database **28** will be present on the cellular data network with any other components that are needed to provide cellular telecommunication services. The server 32 may provide other functions used by one or other components, such as application management functions for the application download server. The server 32 and ADS may use a synchronizer interface tool (not shown) to communicate data between the systems. The server or other components of the system may interface with other systems not shown to transmit or receive data. The

synchronizer tool may be used in supporting communication with these other systems as well.

[0022] The application download server **16**, and/or other servers communicate with a carrier network **40**, through a data link, such as the Internet, a secure LAN, WAN, or other network. The carrier network **40** controls messages (generally being data packets) sent to a messaging service controller ("MSC") **42**. The carrier network **40** communicates with the MSC **42** by a network, the Internet and/or POTS ("plain ordinary telephone system"). Typically, the network or Internet connection between the carrier network **40** and the MSC **42** transfers data, and the POTS transfers voice information. The MSC **42** is connected to multiple base stations ("BTS") **44**. In a similar manner to the carrier network, the MSC **42** is typically connected to the BTS **44** by both the network and/or Internet for data transfer and POTS for voice information. The BTS **44** ultimately broadcasts messages wirelessly to the wireless devices, such as cellular telephone **12**, by short messaging service ("SMS"), or other over-the-air methods known in the art.

[0023] The wireless device, such as cellular telephone **12**, has a computer platform **50** that can receive and execute software applications and display data transmitted from the application download server **16**. The computer platform **50** also allows the wireless device to interact with data and applications resident on network servers. The computer platform **50** includes, among other components, a display driver **52** that drives the graphics display **13** and renders images on the graphics display **13** based upon graphics data received at the computer platform **50**. The computer platform **50** also includes an application-specific integrated circuit ("ASIC") **54**, or other processor, microprocessor, logic circuit, or other data processing device. The ASIC **52** is installed at the time of manufacture of the wireless device and is not normally upgradeable. The ASIC **52** or other processor executes the application programming interface ("API") layer **56** that interfaces with any resident programs in the memory **58** of the wireless device. The memory can be comprised of read-only or random-access memory (RAM and ROM), EPROM, EEPROM, flash cards, or any memory common to computer platforms. The computer platform **50** also includes a local database **60** that can hold the software applications not actively used in memory **58**, such as the software applications downloaded from the application download server **16**. The local database **60** is typically comprised of one or more flash memory cells, but can be any secondary or

tertiary storage device as known in the art, such as magnetic media, EPROM, EEPROM, optical media, tape, or soft or hard disk.

[0024] The wireless device, such as cellular telephone **12**, can access and download many types of applications, such as games and stock monitors, or simply data such as news and sports-related data. The downloaded data can be immediately displayed on the display or stored in the local database **60** when not in use. The software applications can be treated as a regular software application resident on the wireless device **12**,**18**,**20**,**22**, and the user of the wireless device can selectively upload stored resident applications from the local database **60** to memory **58** for execution on the API **56**. The end-user of the wireless device **12**,**18**,**20**,**22** can also selectively delete a software application from the local database **60**.

[0025] Figure 3 is a block diagram depicting the architecture of a synchronizer tool interface in an exemplary embodiment of the present invention. In one embodiment, the synchronizer tool **305** enables tables in one RDBMS system to be synchronized with a destination system via a standard XML interface. It will be recognized by those skilled in the art that the XML language is an implementation choice and other languages may be used. Furthermore, it will also be recognized that the synchronizer tool may be used to synchronize or communicate other information, not just information from RDBMS tables, between computer systems or subsystems. In addition, the connections between the interface and the target and source systems may be by any communication media available, such as wireless, including RF, satellite and infrared communication, and wire-based communication methods. In one embodiment, the communications medium is IP based.

[0026] Data received from the RDBMS **320** in the source system **310** is stored in the queue **325** of the synchronizer tool **305**. This data is translated into a common file format such as XML. By having a storage mechanism, such as a queue, the synchronizer tool can maintain the data for the RDBMS **315** even if the source system **310** is off line. The synchronizer tool **305** may also be implemented to provide an acknowledgement to the source system **310** when the data sent to the target system **315** is replicated in the RDBMS **315** (the target system may provide an acknowledgement to the synchronizer tool 305 indicating this replication has occurred, acknowledge the receipt of the data, or acknowledge other processing was performed).

[0027] The source system **310** is connected to the synchronizer tool **305**. This connection may be by any communication mechanism, including wireless and/or wire-based connection (or combination thereof). The synchronizer tool **305** may be local to the source system **310** or it may be remote to it. The source system transfers information to the synchronizer via this connection. This information may be data for replication to a remote database, or may be any type of information destined for another system, such as a message request, data request or other data transfer. In one embodiment, the source system **310** sends this information to the synchronizer tool **305** using an XML format.

[0028] The target system receives the information from the synchronizer tool **305** and may respond with an acknowledgement to the synchronizer tool **305**. Alternatively, the synchronizer tool may interact directly with a subsystem in the target system **300** (such as a RDBMS **315**) to perform a task such as data replication.

[0029] Note the definition of a target system **300** and source system **310** is somewhat arbitrary. In one instance a system may be the source system sending data to a target system while in another instance the same system is acting as the target system receiving data from another system.

[0030] Figure 4 is a flowchart depicting the process of the synchronizer tool interface in an exemplary embodiment of the present invention. The method begins by having the source system translate data into a XML format (Step **400**) or some other common format. This data may be in the form of scalar data for the target system, a request for information from the target system or other information to be transferred for processing by the target system. Next the data is transmitted to and received by the synchronizer tool (Step **405**).

[0031] The data is stored in a persistent intermediate storage (Step **405**), such as a queue within a synchronizer tool. This allows some independence between the source and target systems. The source system may make the request, have it stored in the queue and can go offline while the request gets transmitted to and/or processed by the target system. In addition, the synchronizer tool may send an acknowledgement to the source system that it received the data.

[0032] The data in XML format is then transmitted from the synchronizer tool to the target system (Step **415**). Note that the "data" can be any piece of information desired to be sent to the target system. It may be data used for replication into an RDBMS of

the target system. Alternatively, it may be any type of command or information sent to the target system for possible processing by the target system. In addition, there may be multiple target systems receiving this data. The synchronization tool may queue multiple data transmission requests for multiple destinations, including multi-target system destination requests.

[0033] The method continues to initiate the translated command on a RDBMS (Step **415**). After translating the command into the appropriate language for the RDBMS, it initiates this command by direct access to the target system's RDBMS or by sending the command to a processor, such as one in the target system, for execution.

[0034] If the data is received at the target (Step **420**), the "Yes" branch is followed and an acknowledgement is sent to the source system (Step **425**). The source system formats this acknowledgement into a format readable by the source system, such as in XML. If the source system is not online, the acknowledgement may be queued and further attempts may be made to the source system to inform it that the data was transmitted to the target system. The synchronizer tool may determine if the data was received by receiving an acknowledgement from the target system.

[0035] If data is received at the target as determined in Step **420**, such as may occur when the target system or RDBMS is offline, then the "No" branch is followed and the data remains in the queue and will retry (Step **430**) to initiate a transmission of the data to the target system as described in Step **415**. There are many queuing algorithms and time parameters possible to determine when to initiate another transmission to the target system. The preferred algorithm and time is based on the processing capacity, efficiency, available resources (such as available queuing memory) as well as other implementation factors of the systems involved.

[0036] The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled

in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM.

[0037] In addition, it should be noted that various changes and modifications could be made herein without departing from the scope of the invention as defined by the appended claims and their equivalents. Furthermore, although elements of the invention may be described or claimed in the singular, the plural is contemplated unless limitation to the singular is explicitly stated.

CLAIMS

What is claimed is:

1. A method of processing information, comprising:

receiving the information from a source system in a common format;

storing the information;

initiating the transmission of the information to the target system while retaining the information;

receiving an target system acknowledgement indicating the target system received the information; and

sending an acknowledgement to the source system upon receiving a target system acknowledgement.

2. The method of claim 1, wherein the common format is an XML format.

3. The method of claim 1, wherein the target system is a database and the information includes data for replication in the database.

4. The method of claim 1, wherein the information is a command directed toward the target system and the sending the target system acknowledgement indicates the command was processed.

5. A method of processing information in an interface, comprising:

receiving data in a first format from a source system;

storing the data in a persistent intermediate storage;

initiating the transmission of the information to the target system while retaining the information;

receiving an indication that the transmission of information to the target system was unsuccessful; and

reinitiating the transmission of the information to the target system.

6. The method of claim 5 further comprising:

storing data received from multiple source systems associated with multiple target systems; and

initiating the transmission of the data from multiple source systems to the associated multiple target systems.

7. The method of claim 5 further comprising:

receiving a source system acknowledgement indicating the acknowledgement was received; and

removing the data from the persistent intermediate storage.

8. An interface, comprising:

a connection to a source system and a target system;

a storage to store data from the source system;

a processor configured to receive data from the source system, to store the data in the storage, to initiate the transmission of the data to the target system, to receive a target system acknowledgement from the target system, and to send an acknowledgement to the source system.

9. The interface from claim 8, wherein the processor is further configured to reinitiate the transmission of data to the target system.

10. A system for processing information in an interface, comprising:

means for receiving the information from a source system in a common format;

means for storing the information;

means for initiating the transmission of the information to the target system while retaining the information;

means for receiving an target system acknowledgement indicating the target system received the information; and

means for sending an acknowledgement to the source system upon receiving a target system acknowledgement.

11. A computer-readable medium containing computer-executable instructions for allowing access to a device resource that when executed perform a method, comprising:

receiving the information from a source system in a common format;

storing the information;

initiating the transmission of the information to the target system while retaining the information;

receiving an target system acknowledgement indicating the target system received the information; and

sending an acknowledgement to the source system upon receiving a target system acknowledgement.

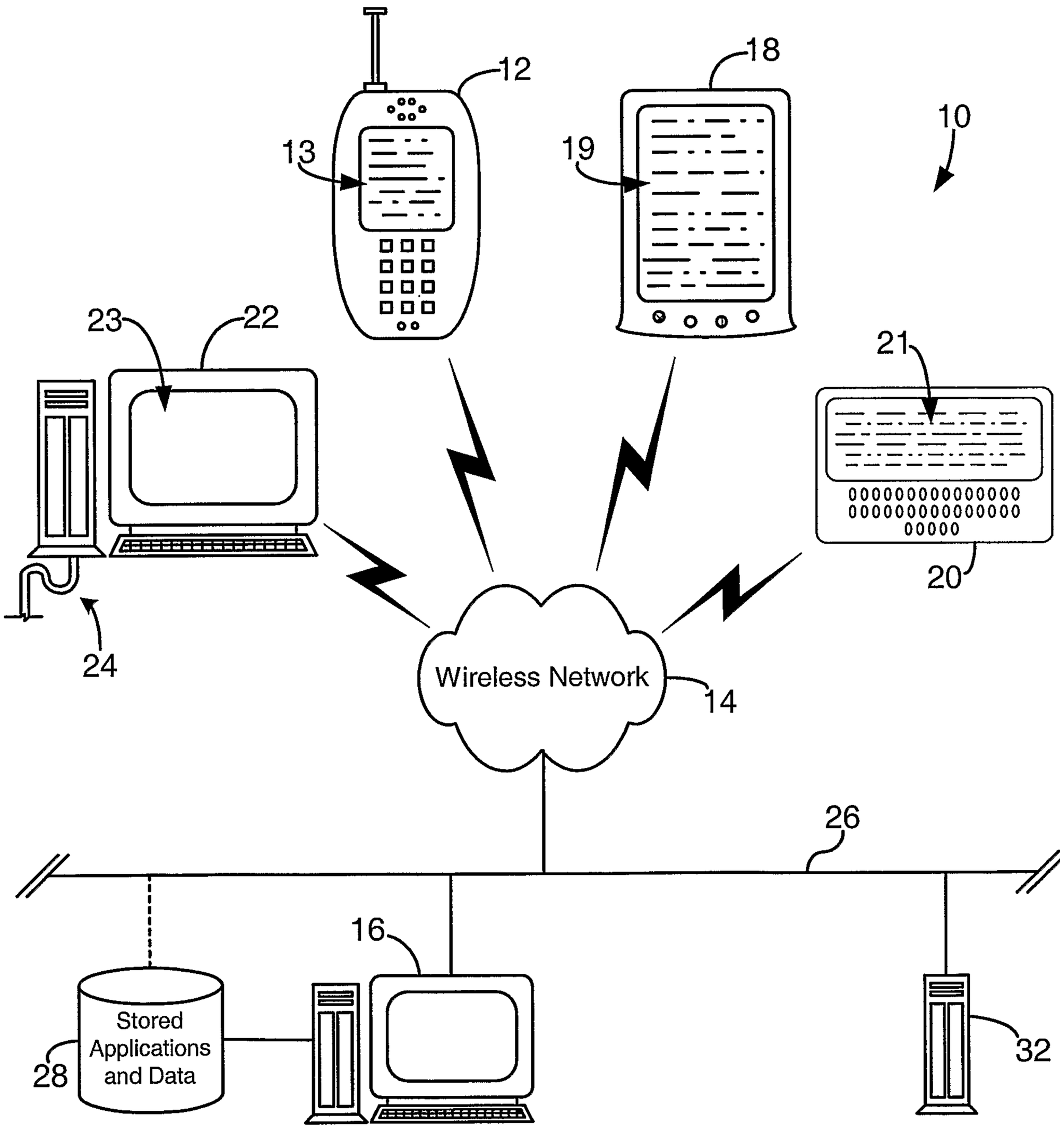

12
13
18
19
10
23
22
21
00000000000000000
00000000000000000
00000
20
24
Wireless Network
14
26
16
Stored
Applications
and Data
28
32

Figure 1

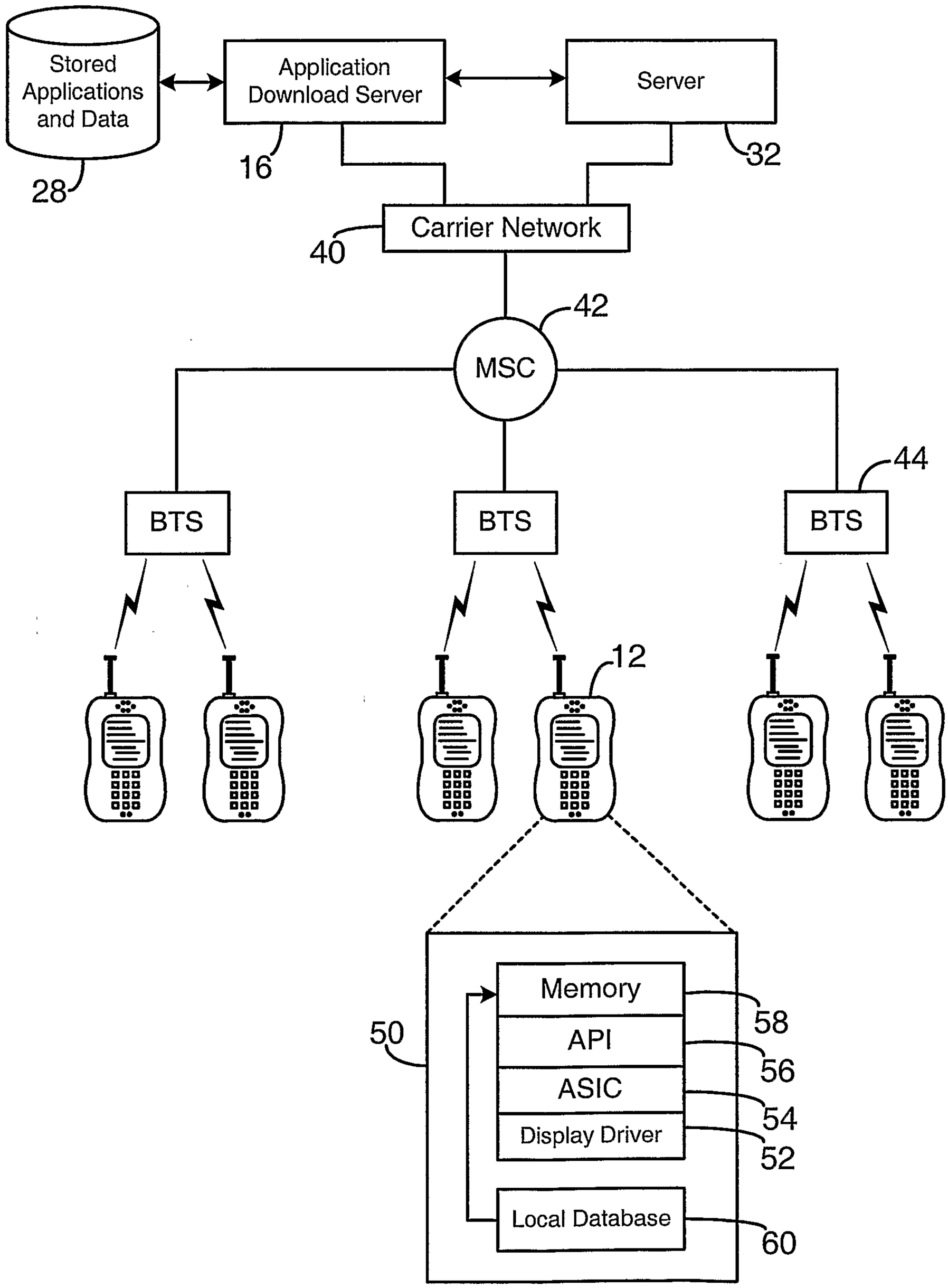

Stored Applications and Data
28
Application Download Server
16
Server
32
Carrier Network
40
MSC
42
BTS
BTS
BTS
44
12
Memory
API
ASIC
Display Driver
58
56
54
52
50
Local Database
60

Figure 2

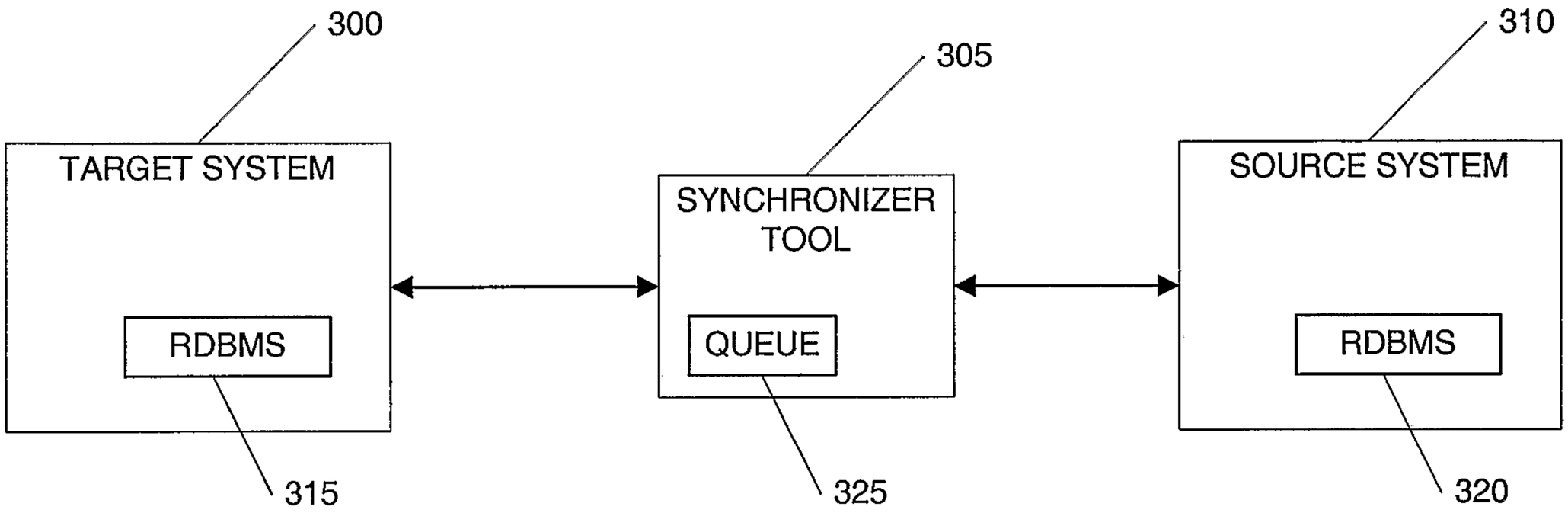

300
TARGET SYSTEM
RDBMS
315
305
SYNCHRONIZER TOOL
QUEUE
325
310
SOURCE SYSTEM
RDBMS
320

Figure 3

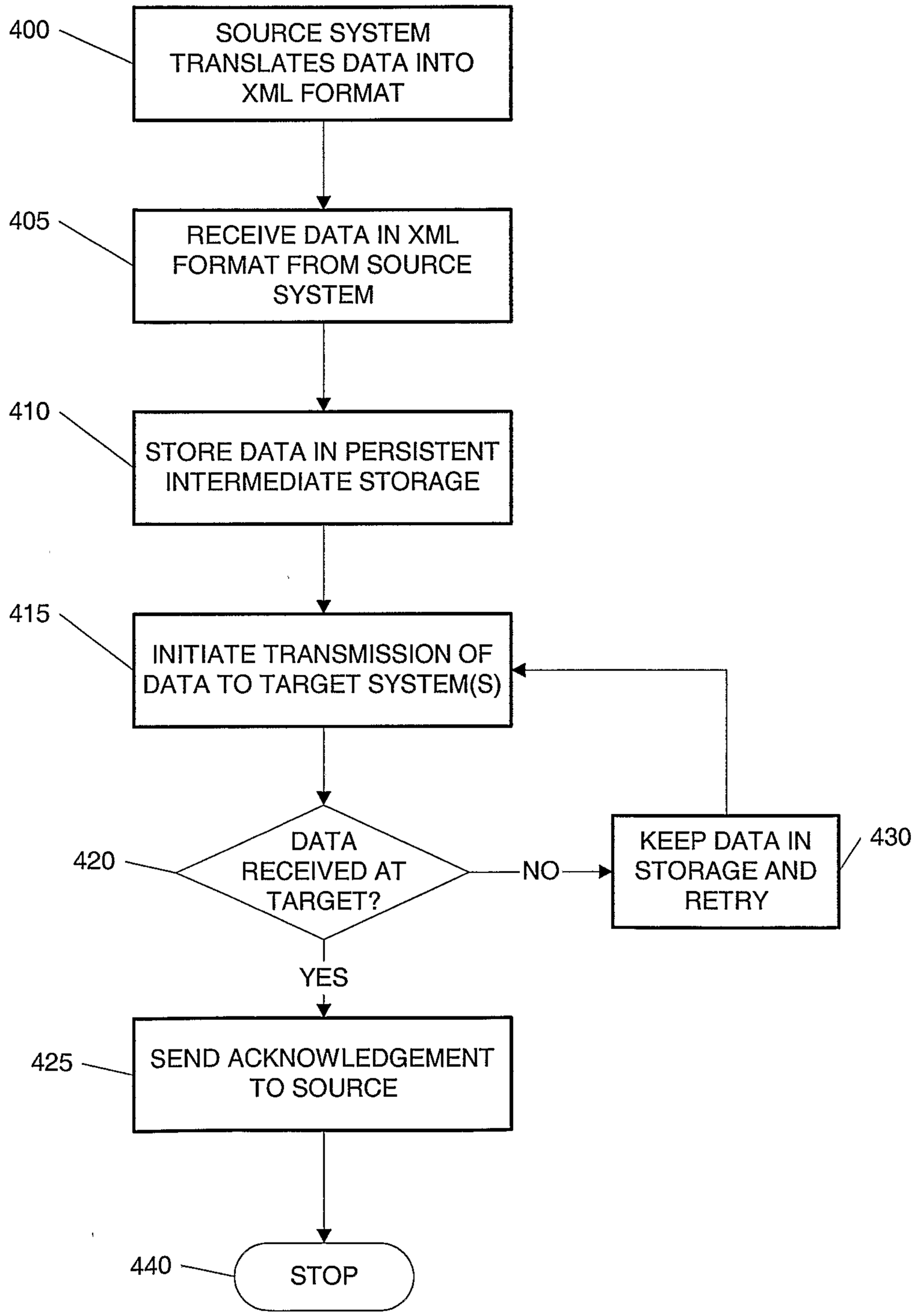
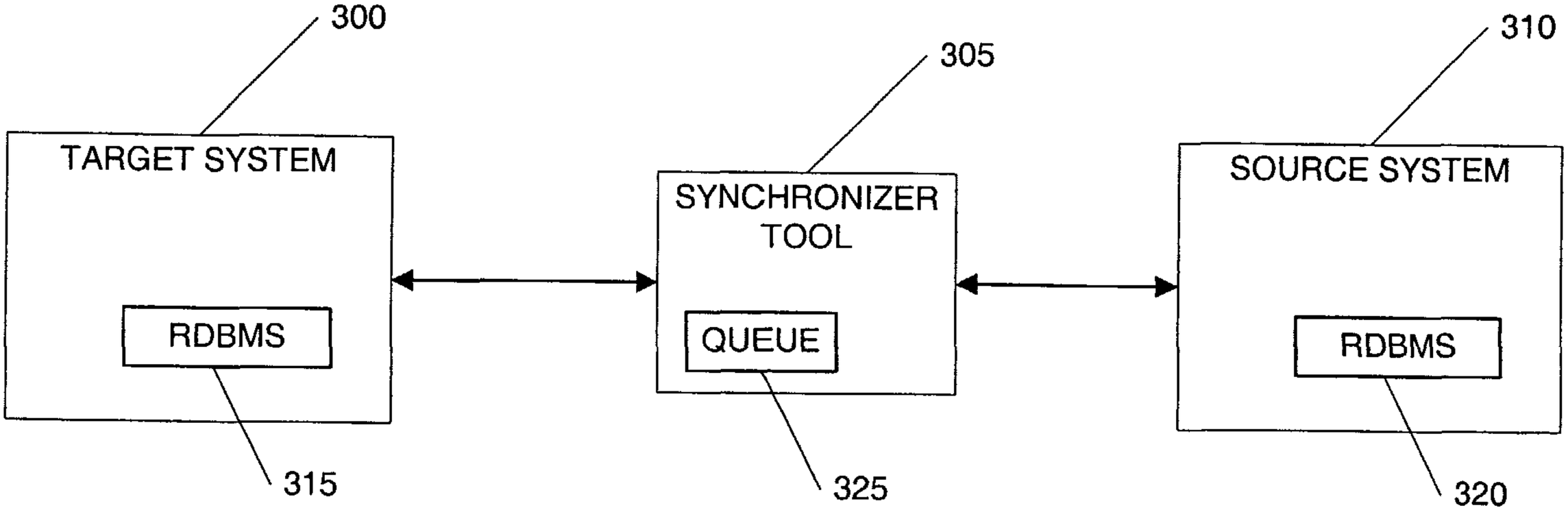

400
SOURCE SYSTEM TRANSLATES DATA INTO XML FORMAT
405
RECEIVE DATA IN XML FORMAT FROM SOURCE SYSTEM
410
STORE DATA IN PERSISTENT INTERMEDIATE STORAGE
415
INITIATE TRANSMISSION OF DATA TO TARGET SYSTEM(S)
420
DATA RECEIVED AT TARGET?
NO
KEEP DATA IN STORAGE AND RETRY
430
YES
425
SEND ACKNOWLEDGEMENT TO SOURCE
440
STOP

Figure 4


300
TARGET SYSTEM
RDBMS
315
305
SYNCHRONIZER TOOL
QUEUE
325
310
SOURCE SYSTEM
RDBMS
320