

US00RE40576E

(19) **United States**
(12) **Reissued Patent**
Goodwin, III

(10) **Patent Number:** **US RE40,576 E**
(45) **Date of Reissued Patent:** **Nov. 18, 2008**

(54) **POINT-OF-SALE SYSTEM INCLUDING ISOLATION LAYER BETWEEN CLIENT AND SERVER SOFTWARE**

(75) Inventor: **John C. Goodwin, III**, Suwanee, GA (US)

(73) Assignee: **NCR Corporation**, Dayton, OH (US)

(21) Appl. No.: **10/171,366**

(22) Filed: **Jun. 12, 2002**

DE	3114139	A1	11/1982
DE	3330301	A1	3/1985
EP	0060701	B1	9/1982
EP	0124355	B1	11/1984
EP	0178223	A2	4/1986
EP	0199232		10/1986
EP	0491348	A2	6/1992
EP	0767437		4/1997
EP	0811958	A2	12/1997
GB	2068132	A	8/1981
GB	2285687	A	7/1995
JP	5314367		11/1993
WO	9107725		5/1991
WO	9415294		7/1994

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **5,987,426**
Issued: **Nov. 16, 1999**
Appl. No.: **08/949,410**
Filed: **Oct. 14, 1997**

(51) **Int. Cl.**
G06F 15/16 (2006.01)
G06F 15/00 (2006.01)

(52) **U.S. Cl.** **709/232; 709/203; 705/21; 705/16; 705/17**

(58) **Field of Classification Search** **709/203, 709/232; 705/16, 17, 20, 21, 64**
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,182,357	A	5/1965	Tuttle
3,263,774	A	8/1966	Reese
3,436,968	A	4/1969	Unger
3,681,570	A	8/1972	Abt
3,681,571	A	8/1972	Strohschneider
3,688,873	A	9/1972	Potrafke

(Continued)

FOREIGN PATENT DOCUMENTS

BE	899018	6/1984
DE	1927819	2/1971
DE	2125869	12/1972

OTHER PUBLICATIONS

POS System for Massmerchant Store: System 87 Plus, Victor Landa et al., pp. 319-327, Fujitsu Scientific & Technical Journal, No. 4, Winter 1987.

U-Scan Express.TM., The U-Scan Express.TM. Self-Checkout System, Spectra-Physics Scanning Systems, Inc., Feb. 1996.

(Continued)

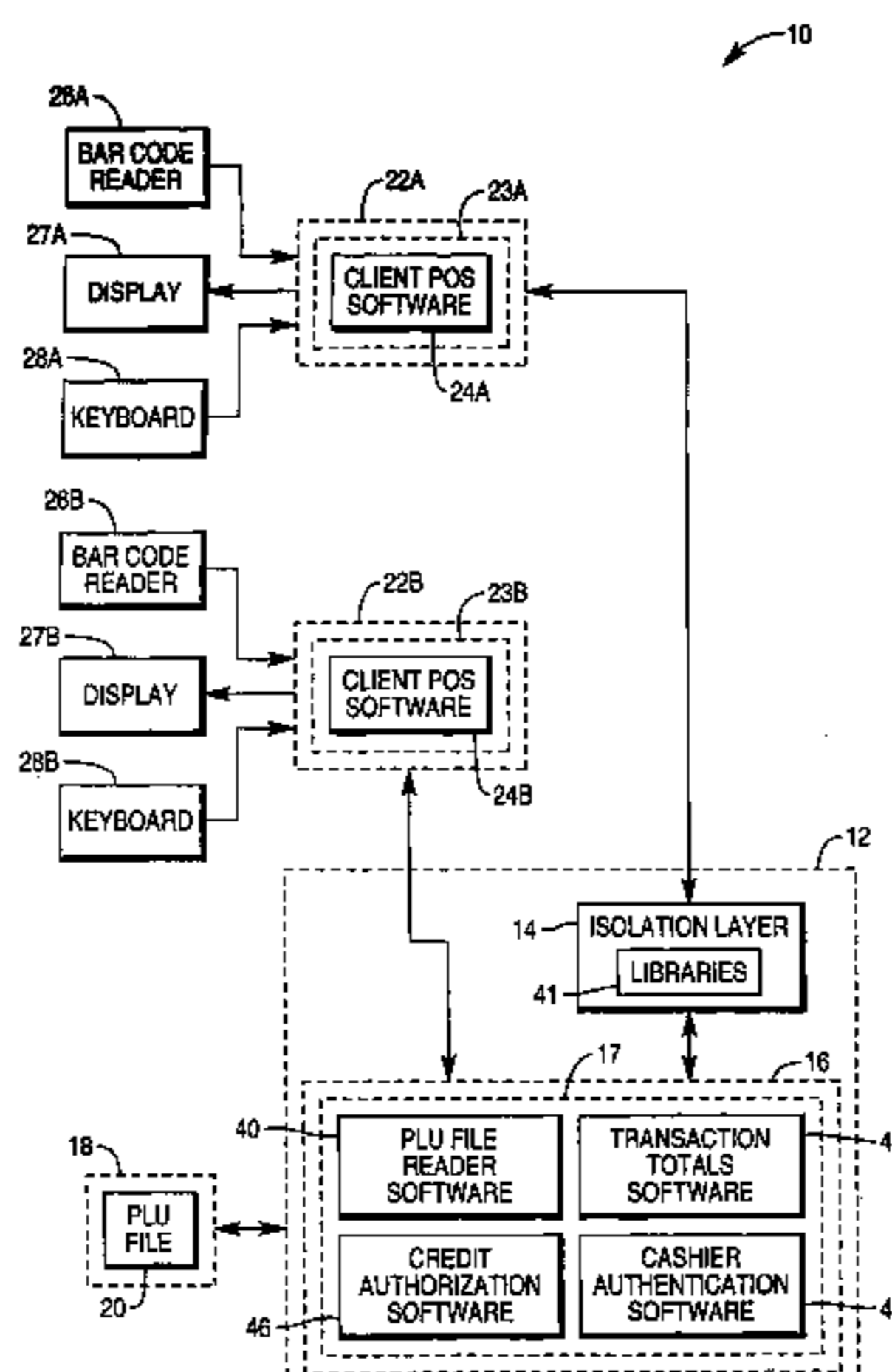
Primary Examiner—Krisna Lim

(74) *Attorney, Agent, or Firm*—Paul W. Martin; Priest & Goldstein, PLLC

(57) **ABSTRACT**

A system and method of transferring information between a first software application and a second software application which employ an isolation layer. The system includes a client computer system provided by a first seller of computer systems, including a client software application, and a server computer system provided by a second seller of computer systems different from the first seller of computer systems, including a server software application which provides information from the server computer system to the client computer system. The system additionally includes isolation layer software, either at the client computer system or at the server computer system, which facilitates communication between the client software application and the server software application to transfer the information.

34 Claims, 5 Drawing Sheets



US RE40,576 E

Page 2

U.S. PATENT DOCUMENTS					
			4,940,116 A	7/1990	O'Connor et al.
			4,947,028 A	8/1990	Gorog
3,696,586 A	10/1972	Sparks	4,959,530 A	9/1990	O'Connor
3,725,895 A	4/1973	Haynes	4,964,053 A	10/1990	Humble
3,735,350 A	5/1973	Lemelson	4,964,167 A	10/1990	Kunizawa et al.
3,819,012 A	6/1974	Beck, Jr., et al.	4,971,177 A	11/1990	Nojiri et al.
3,836,755 A	9/1974	Ehrat	4,975,948 A	12/1990	Andresen et al.
3,869,575 A	3/1975	Spitz, et al.	5,004,916 A	4/1991	Collins, Jr.
3,878,365 A	4/1975	Schwartz	5,040,636 A	8/1991	Forsythe et al.
3,947,817 A	3/1976	Requa, deceased et al.	5,053,636 A	10/1991	Zelina
3,949,194 A	4/1976	Catto et al.	5,058,691 A	10/1991	Sela
3,962,569 A	6/1976	Loshbough et al.	5,079,412 A	1/1992	Sugiyama
3,962,570 A	6/1976	Loshbough et al.	5,083,638 A	1/1992	Schneider
3,984,667 A	10/1976	Loshbough	5,115,888 A	5/1992	Schneider
3,986,012 A	10/1976	Loshbough et al.	5,121,103 A	6/1992	Minasy et al.
4,004,139 A	1/1977	Hall	5,123,494 A	6/1992	Schneider
4,028,537 A	6/1977	Snow	5,125,465 A	6/1992	Schneider
4,069,970 A	1/1978	Buzzard et al.	5,139,100 A	8/1992	Brauneis
4,071,740 A	1/1978	Gogulski	5,140,145 A	8/1992	Bianco
4,084,742 A	4/1978	Silverman	5,142,161 A	8/1992	Brackmann
4,114,721 A	9/1978	Glenn, Jr.	5,149,211 A	9/1992	Pettigrew et al.
4,138,000 A	2/1979	Hartup	5,168,961 A	12/1992	Schneider
4,177,880 A	12/1979	Joseloff	5,174,399 A	12/1992	Brauneis
4,180,204 A	12/1979	Koenig et al.	5,174,413 A	12/1992	Cappi et al.
4,236,604 A	12/1980	Warner	5,177,345 A	1/1993	Baitz
4,237,483 A	12/1980	Clever	5,177,800 A	1/1993	Coats
4,327,819 A	5/1982	Coutta	5,178,234 A	1/1993	Sakurai et al.
4,347,903 A	8/1982	Yano et al.	5,189,291 A	2/1993	Siemiatkowski
4,353,564 A	10/1982	Joseloff	5,191,749 A	3/1993	Cappi et al.
4,369,334 A	1/1983	Nakatani et al.	5,195,613 A	3/1993	Kohno
4,369,361 A	1/1983	Swartz et al.	5,230,391 A	7/1993	Murata et al.
4,373,133 A	2/1983	Clyne et al.	5,231,293 A	7/1993	Longacre, Jr.
4,377,741 A	3/1983	Brekka et al.	5,239,167 A	8/1993	Kipp
4,392,535 A	7/1983	Fujii et al.	5,250,789 A	10/1993	Johnsen
4,412,298 A	10/1983	Feinland et al.	5,256,864 A	10/1993	Rando et al.
4,500,880 A	2/1985	Gomersall et al.	5,265,075 A	11/1993	Bergeron et al.
4,509,129 A	4/1985	Yatsunami et al.	5,293,319 A	3/1994	DeSha et al.
4,529,871 A	7/1985	Davidson	5,297,030 A	3/1994	Vassigh et al.
4,549,867 A	10/1985	Dittakavi	5,314,631 A	5/1994	Katoh et al.
4,574,317 A	3/1986	Scheible	5,315,097 A	5/1994	Collins, Jr. et al.
4,583,083 A	4/1986	Bogasky	5,334,826 A	8/1994	Sato et al.
4,583,619 A	4/1986	Fry	5,335,171 A	8/1994	Fuyama
4,602,152 A	7/1986	Dittakavi	5,340,970 A	8/1994	Wolfe, Jr. et al.
4,605,080 A	8/1986	Lemelson	5,343,025 A	8/1994	Usui
4,660,160 A	4/1987	Tajima et al.	5,345,071 A	9/1994	Dumont
4,661,908 A	4/1987	Hamano et al.	5,361,871 A	11/1994	Gupta et al.
4,676,343 A	6/1987	Humble et al.	5,365,050 A	11/1994	Worthington et al.
4,679,154 A	7/1987	Blanford	5,374,815 A	12/1994	Waterhouse et al.
4,688,105 A	8/1987	Bloch et al.	5,375,680 A	12/1994	Ikeda et al.
4,709,770 A	12/1987	Kohashi et al.	5,378,860 A	1/1995	Dingfelder et al.
4,715,457 A	12/1987	Amacher et al.	5,382,779 A	1/1995	Gupta
4,723,118 A	2/1988	Hooley et al.	5,393,965 A	2/1995	Bravman et al.
4,725,822 A	2/1988	Hooley	5,397,882 A	3/1995	Van Solt
4,729,097 A	3/1988	Takaoka et al.	5,398,191 A	3/1995	Komai
4,734,713 A	3/1988	Sato et al.	5,406,063 A	4/1995	Jelen
4,736,098 A	4/1988	Rehrig	5,410,108 A	4/1995	Williams et al.
D296,788 S	7/1988	Schneider et al.	5,412,191 A	4/1995	Baitz et al.
4,766,296 A	8/1988	Barth	5,424,521 A	6/1995	Wolfe, Jr. et al.
4,766,299 A	8/1988	Tierney et al.	5,424,524 A	6/1995	Ruppert et al.
4,775,782 A	10/1988	Mergenthaler et al.	5,426,282 A	6/1995	Humble
4,779,706 A	10/1988	Mergenthaler	5,426,423 A	6/1995	Raimbault et al.
4,787,467 A	11/1988	Johnson	5,434,394 A	7/1995	Roach et al.
4,789,048 A	12/1988	Cramer et al.	5,437,346 A	8/1995	Dumont
4,792,018 A	12/1988	Humble et al.	5,448,046 A	9/1995	Swartz
4,800,493 A	1/1989	Takagi	5,448,226 A	9/1995	Failing, Jr. et al.
4,838,383 A	6/1989	Saito et al.	5,454,063 A	9/1995	Rossides
4,855,908 A	* 8/1989	Shimoda et al. 705/20	5,457,307 A	10/1995	Dumont
4,866,661 A	9/1989	de Prins	5,461,561 A	10/1995	Ackerman et al.
4,881,606 A	11/1989	Halfon et al.	5,478,989 A	12/1995	Shepley
4,909,356 A	3/1990	Rimondi et al.	5,482,139 A	1/1996	Rivalto
4,914,280 A	4/1990	Peterson	5,484,991 A	1/1996	Sherman et al.
4,929,819 A	5/1990	Collins, Jr.			

5,485,006 A	1/1996	Allen et al.	5,877,485 A	3/1999	Swartz
5,488,202 A	1/1996	Baitz et al.	5,878,064 A	3/1999	Goodwin, III
5,489,773 A	2/1996	Kumar	5,883,968 A	3/1999	Welch et al.
5,493,107 A	2/1996	Gupta et al.	5,884,281 A	3/1999	Smith et al.
5,493,613 A	2/1996	Denno et al.	5,884,728 A	3/1999	d'Estaintot et al.
5,494,136 A	2/1996	Humble	5,886,336 A	3/1999	Tang et al.
5,497,314 A	3/1996	Novak	5,890,135 A	3/1999	Powell
5,497,853 A	3/1996	Collins, Jr. et al.	5,898,158 A	4/1999	Shimizu et al.
5,500,890 A	3/1996	Rogge et al.	5,898,383 A	4/1999	Forsythe
5,525,786 A	6/1996	Dumont	5,900,614 A	5/1999	Nakakawaji et al.
5,540,301 A	7/1996	Dumont	5,901,204 A	5/1999	Gallacher et al.
5,541,925 A *	7/1996	Pittenger et al. 370/389	5,914,473 A	6/1999	Gresky
5,543,607 A	8/1996	Watanabe et al.	5,917,421 A	6/1999	Saunders
5,544,040 A	8/1996	Gerbaulet	5,923,735 A	7/1999	Swartz et al.
5,546,475 A	8/1996	Bolle et al.	5,925,872 A	7/1999	Wyatt et al.
5,560,450 A	10/1996	Kouno	5,950,173 A	9/1999	Perkowski
5,577,165 A	11/1996	Takebayashi et al.	5,965,861 A	10/1999	Addy et al.
5,579,000 A	11/1996	Mulqueen	5,970,470 A	10/1999	Walker et al.
5,594,469 A	1/1997	Freeman et al.	5,978,772 A	11/1999	Mold
5,595,264 A	1/1997	Trotta, Jr.	5,979,757 A	11/1999	Tracy et al.
5,609,223 A	3/1997	Iizaka et al.	5,984,177 A	11/1999	Do et al.
5,635,906 A	6/1997	Joseph	5,987,428 A	11/1999	Walter
5,637,847 A	6/1997	Watanabe	5,992,570 A	11/1999	Walter et al.
5,644,630 A	7/1997	Durco, Jr.	5,993,047 A	11/1999	Novogrod et al.
5,662,190 A	9/1997	Abe	6,000,522 A	12/1999	Johnson
5,663,963 A	9/1997	Goodwin, III et al.	6,002,808 A	12/1999	Freeman
5,684,289 A	11/1997	Detwiler et al.	6,009,538 A	12/1999	Goodwin, III et al.
5,708,782 A	1/1998	Larson et al.	6,056,087 A	5/2000	Addy et al.
5,710,887 A *	1/1998	Chelliah et al. 705/26	6,056,194 A	5/2000	Kolls
5,721,420 A	2/1998	May	6,080,938 A	6/2000	Lutz
5,721,536 A	2/1998	Mulqueen	6,108,717 A	8/2000	Kimura et al.
RE35,758 E	3/1998	Winter et al.	6,163,797 A *	12/2000	Eckley et al. 709/203
5,734,823 A *	3/1998	Saigh et al. 709/229	6,269,342 B1 *	7/2001	Brick et al. 705/20
5,736,721 A	4/1998	Swartz			
5,740,744 A	4/1998	Nashirozawa et al.			
5,745,705 A *	4/1998	Iguchi 705/21			
5,747,784 A	5/1998	Walter et al.			
5,752,582 A	5/1998	Hayward			
5,754,106 A	5/1998	Goodwin, III			
5,756,978 A	5/1998	Soltész et al.			
5,756,984 A	5/1998	Kobayashi			
5,769,269 A	6/1998	Peters			
5,780,825 A	7/1998	Sato et al.			
5,784,446 A	7/1998	Stuart			
5,801,371 A	9/1998	Kahn et al.			
5,801,766 A	9/1998	Alden			
5,814,799 A	9/1998	Swartz et al.			
5,821,512 A	10/1998	O'Hagan et al.			
5,832,417 A	11/1998	Petrucelli et al.			
5,832,457 A	11/1998	O'Brien et al.			
5,837,991 A	11/1998	LaManna et al.			
5,839,104 A	11/1998	Miller et al.			
5,839,116 A *	11/1998	Goodwin, III 705/20			
5,845,259 A	12/1998	West et al.			
5,845,263 A	12/1998	Camaisa et al.			

OTHER PUBLICATIONS

Optimal Robotics 8870 System. TM., Optimal Robotics Corporation, 1995.
 Zimmerman, Denise, "Kroger Unit to Expand Customer Self-Scanning", Supermarket News, Aug. 14, 1995, p. 15.
 Zimmerman, Denise, "Kroger Self-Scan Lane to Test Debit Credit" Supermarket News, vol. 45, No. 14, p. 13+, Apr. 3, 1995.
 Lang Jones, Jeanne, "Robots Will Do Checkout at Grocery", Houston Post, (Final, Business, p. C1, Feb. 22, 1995.
 "Burger King Tests Credit-Card Payments", The Miami Herald Knight-Rider/Tribune Business News, Jul. 9, 1993.
 Wisenberg-Brin, Dinah, Associated Press, "You-buy, you-scan bag", Dayton Daily News, Aug. 11, 1996.
 Gamburer, A.T., et al., "Analogue-code Converter Used with Strain Gauge Hopper Weighters", Instrumental and Control, Mar. 3, 1969.

* cited by examiner

FIG. 1

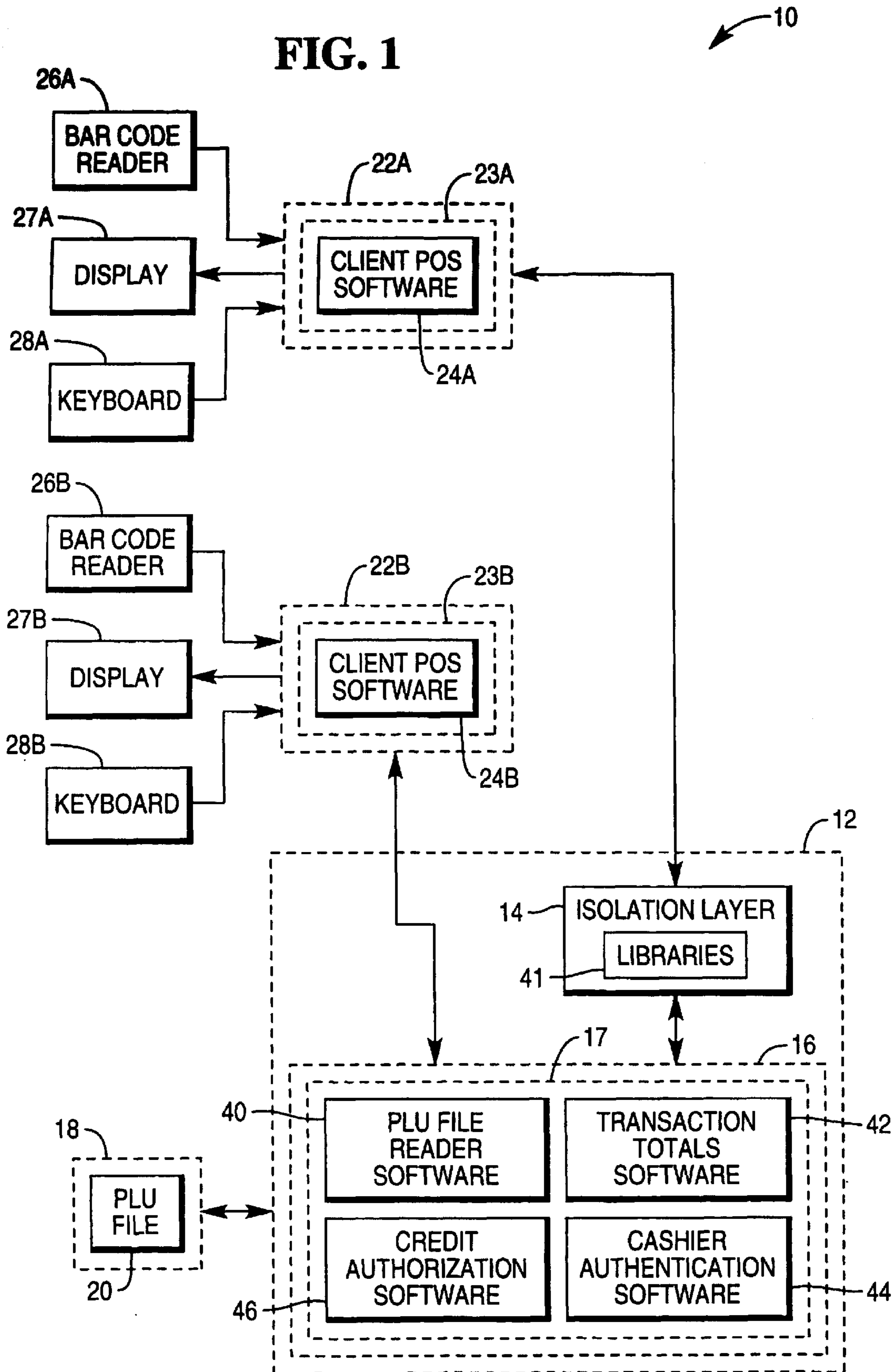


FIG. 2

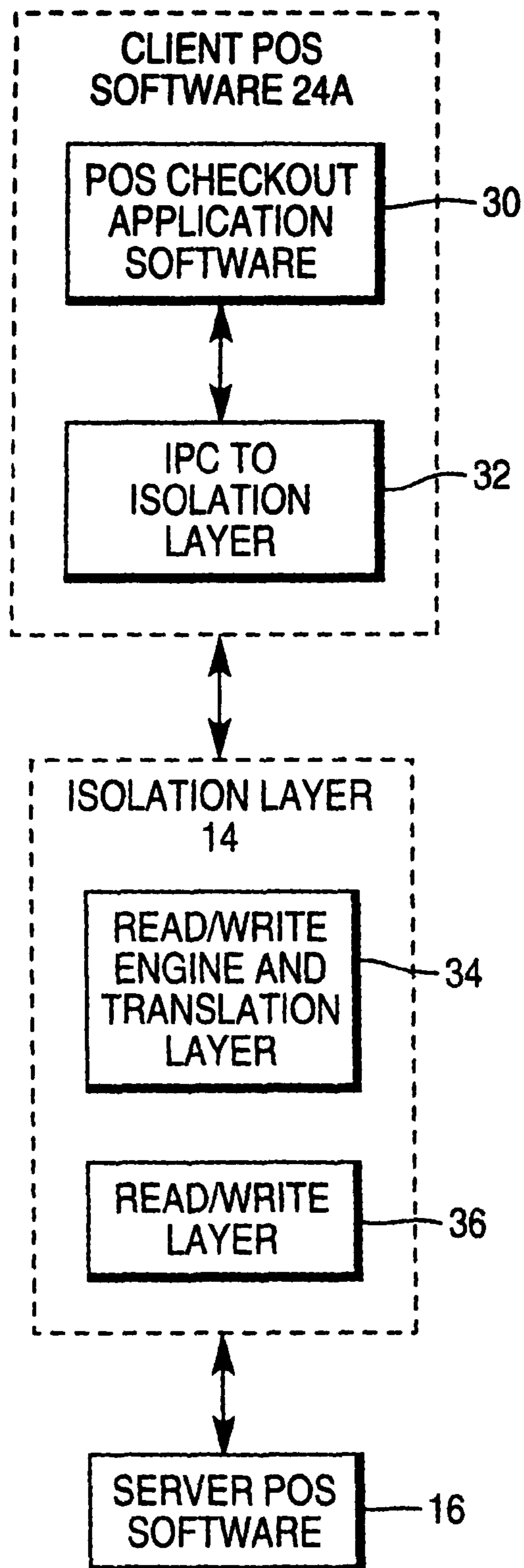


FIG. 3

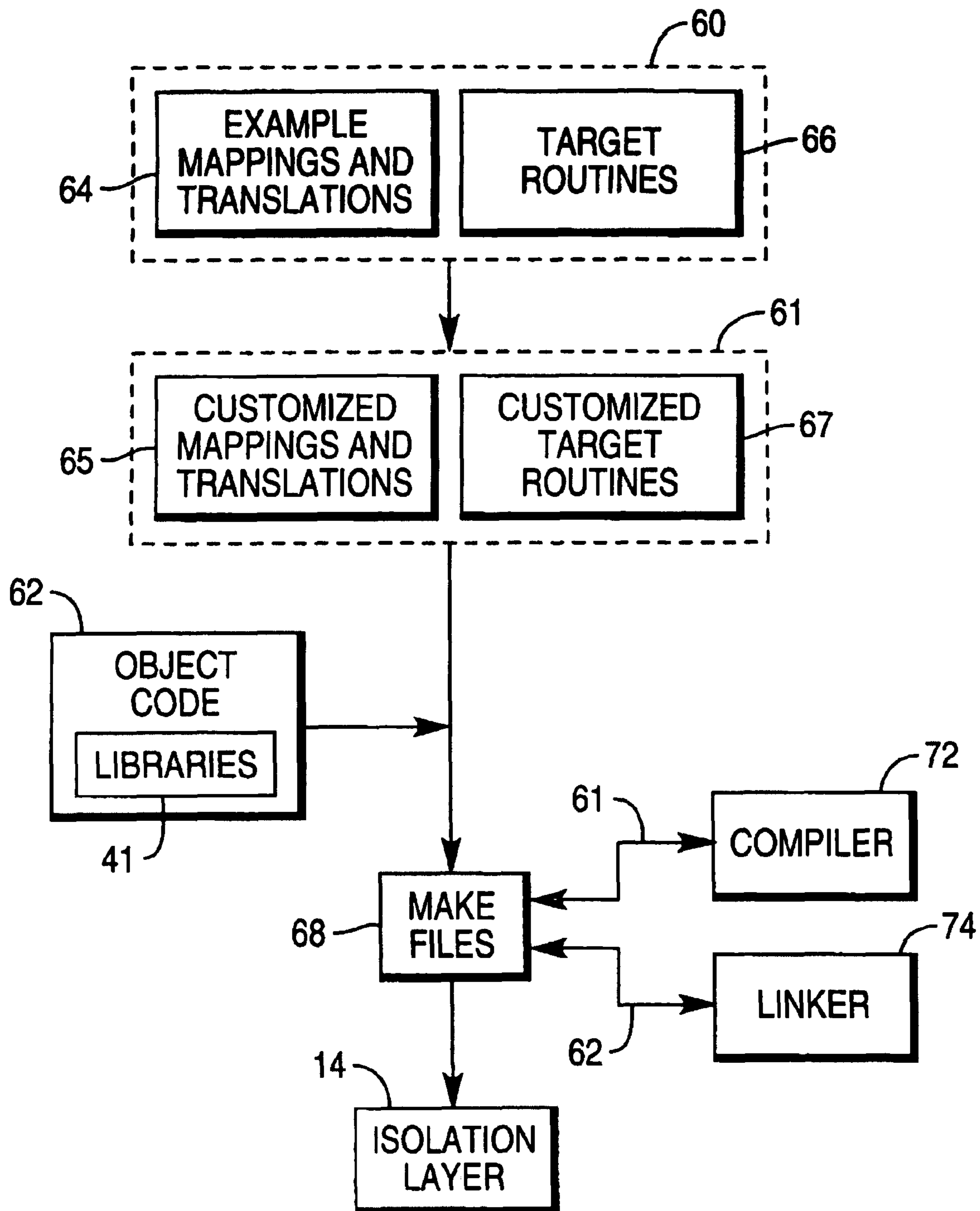


FIG. 4

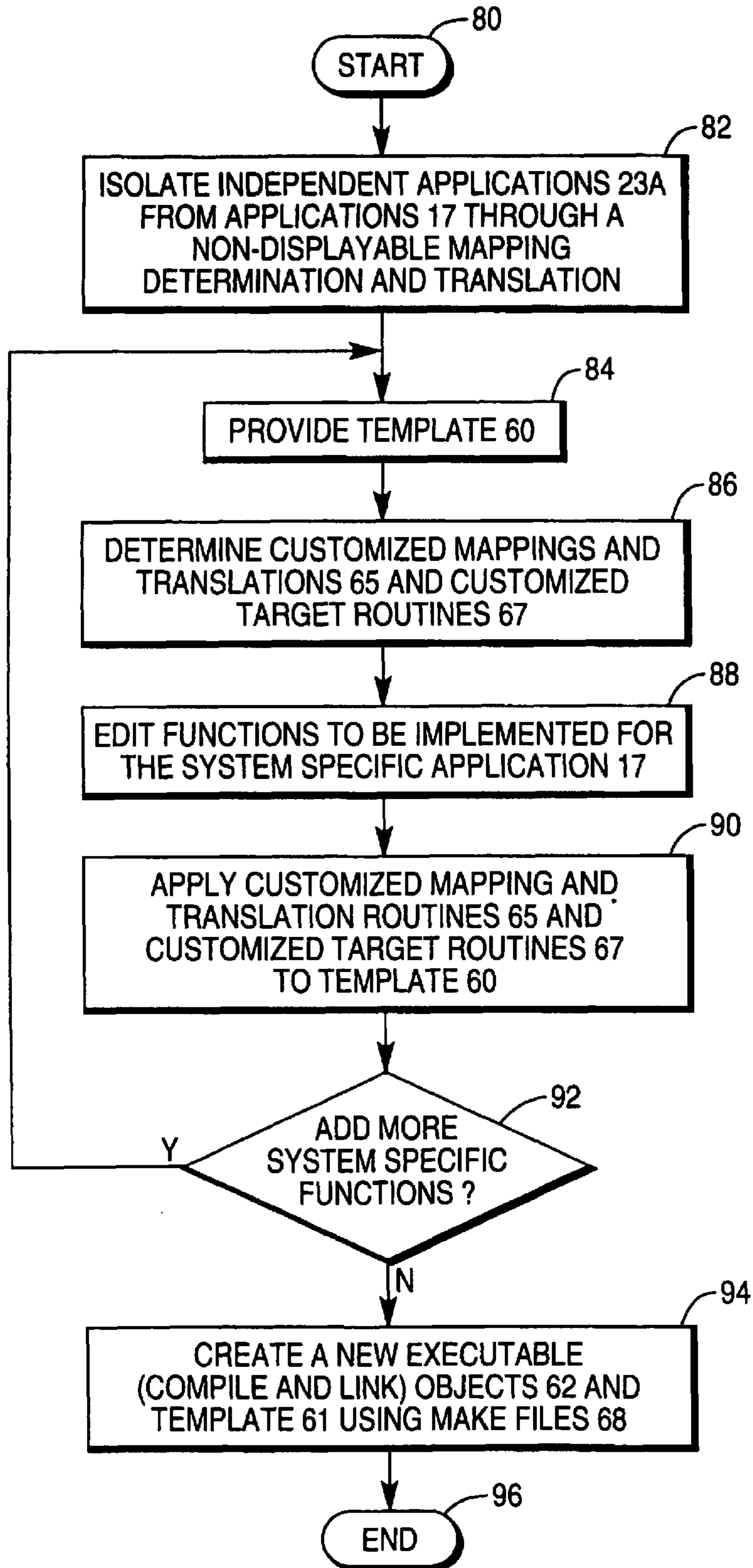
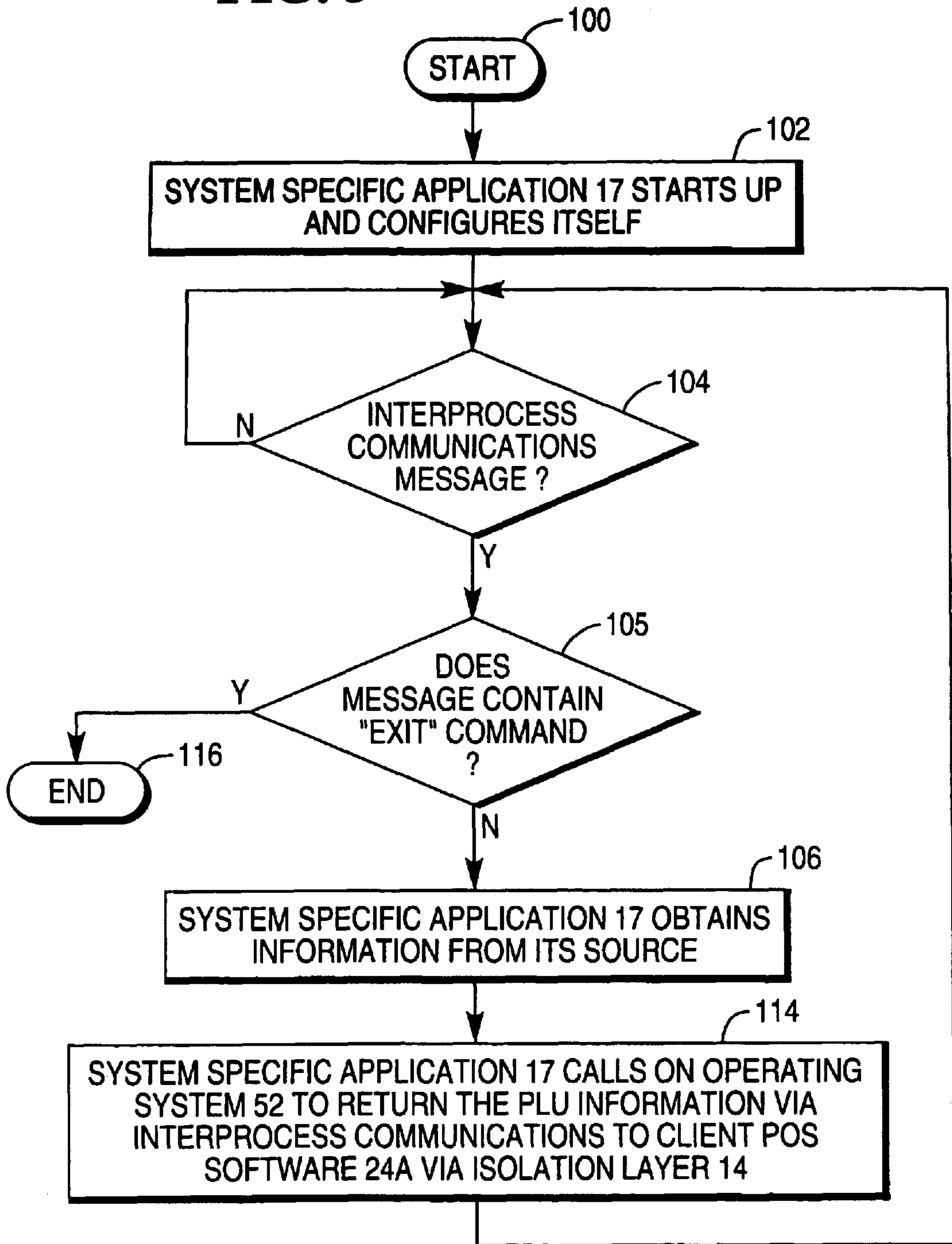


FIG. 5



**POINT-OF-SALE SYSTEM INCLUDING
ISOLATION LAYER BETWEEN CLIENT AND
SERVER SOFTWARE**

Matter enclosed in heavy brackets [] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

CROSS-REFERENCE TO RELATED
APPLICATIONS

The present application is related to the U.S. Pat. No. 5,839,116 issued on Nov. 17, 1998, entitled "SYSTEM AND METHOD OF OBTAINING INFORMATION FROM A PRICE LOOK-UP FILE", and having as inventor, John Goodwin.

BACKGROUND OF THE INVENTION

The present invention relates to point-of-sale (POS) systems, and more specifically to a POS system including an isolation layer between POS server software and a client POS application program.

POS systems typically include a central server and a plurality of checkout terminals connected through a client-server network.

The checkout terminals include bar code readers and keyboards for entry of the item numbers during a transaction.

The central server stores a price look-up (PLU) file which associates item numbers with item prices. The central server processes requests from the checkout terminals for price information for the items. The central server may perform additional tasks, such as storing transaction history (audit log and transaction totals history), storing accountability totals, storing cashier and terminal accountability totals, holding future price change information (batches), performing credit authorization, performing check validation, and performing frequent shopper validation and specials.

POS checkout software has traditionally included client and server software that have been developed together and sold as a single proprietary product. However, this solution may not be an optimal one for a retailer. Retailers may find that combining server software from one developer with client software from another developer gives their POS systems the features that they want. This may be especially true for retailers that want the best solutions for both the server and client software in a market where no one developer of proprietary POS software exceeds at both.

Therefore, it would be desirable to provide a POS system including an isolation layer between the POS server software and the POS client application software to enable server and client software from different developers to be combined.

SUMMARY OF THE INVENTION

In accordance with the teachings of the present invention, a POS system including an isolation layer between POS server software and a client POS application program is provided.

A system and method of transferring information between a first software application and a second software application which employ an isolation layer. The system includes a client computer system provided by a first seller of computer systems, including a client software application, and a server computer system provided by a second seller of computer systems different from the first seller of computer systems,

including a server software application which provides information from the server computer system to the client computer system. The system additionally includes isolation layer software, either at the client computer system or at the server computer system, which facilitates communication between the client software application and the server software application to transfer the information.

The method includes the steps of providing the first software application by a first computer, providing the second software application by a second company different from the first company, executing the first software application by a first computer, executing the second software application by a second computer different from the first computer, providing an isolation layer having hooks into the second software application, executing the isolation layer, and establishing communication between the first software application and the second software application by the isolation layer to transfer the information.

It is accordingly an object of the present invention to provide a POS system including an isolation layer between POS server software and a client POS application program.

It is another object of the present invention to allow server and client software from different developers to be combined using an isolation layer.

It is another object of the present invention to provide an isolation layer for server software that provides read and write services for client POS software in a computer network that includes client computers running different client POS software.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional benefits and advantages of the present invention will become apparent to those skilled in the art to which this invention relates from the subsequent description of the preferred embodiments and the appended claims, taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram of a transaction management system;

FIG. 2 is a block diagram illustrating the client POS software, server POS software, and the isolation layer between them;

FIG. 3 is a block diagram of a system for producing the isolation layer;

FIG. 4 is a flow diagram illustrating how the isolation layer is created; and

FIG. 5 is a flow diagram illustrating operation of a system specific application and, in particular, a method of transferring information from the system specific application to the client POS software through the isolation layer.

DETAILED DESCRIPTION OF THE PREFERRED
EMBODIMENT

Referring now to FIG. 1, transaction system 10 preferably includes host computer system 12 and point-of-service (POS) terminal 22A and 22B. Host computer system 12 and point-of-service (POS) terminals 22A and 22B are coupled together to form a network.

POS terminals 22A and 22B execute independent applications 23A and 23B, including client POS software 24A and 24B for completing transactions. POS software 24A and 24B may be identical (e.g. the same application from the same developer) or may be different transaction processing applications.

As illustrated, independent applications 23A were developed by a different developer than the developer of server

POS software **16**. Thus, independent applications **23A** require isolation layer **14** to communication with POS server software **16**. Independent applications **23B** were developed by the same developer as the developer of POS server software **16**. Thus, independent applications **23B** do not require isolation layer **14** to communication with server POS software **16**. Advantageously, the present invention is capable of connecting a plurality of independent applications **23A** running on a plurality of different client terminals from different developers to server POS software **16**.

Client POS software **24A** and **24B** display transaction information on displays **27A**, **27B**. Client POS software **24A** and **24B** receive article identification information from bar code readers **26A**, **26B** and keyboards **28A**, **28B**. Client POS software **24A** and **24B** send the article identification information to host computer system **12**. Host computer system **12** reads price look-up (PLU) file **20** to obtain price information and sends the price information to POS terminals **26A** and **26B**.

Other examples of independent applications **17** include price checking software for consumer price checking computers, weighing software for electronic scales, price checking software and item description label printing software for hand-held terminals, and EPL auditing software.

Host computer system **12** executes POS server software **16** and isolation layer **14**. POS server software **16** processes requests from independent applications **23A** and **23B**. POS server software **16** includes system specific applications **17**, such as price look-up (PLU) file software **40**, transaction totals software **42**, cashier authentication software **44**, and credit checking software **46**. These are but examples of system specific software in use today. The present invention anticipates other types of system specific software **17** as well.

Isolation layer **14** provides translation services between independent applications **23A** and server POS software **16**. Thus, isolation layer **14** allows client POS software **24A** from one developer to function with system specific applications **17** from a different developer.

Isolation layer **14** may reside in host computer system **12** or client terminal **22A**. Typically, a provider of client terminals **22A** also provides independent applications **23A**, and a provider of host computer system **12** also provides server POS software **16**.

If the provider of client terminals **22A** is adding client terminals **22A** to an existing network including a server from a different provider, the provider of client terminals **22A** may wish to write and store isolation layer **14** in client terminals **22A**.

On the other hand, if a provider of servers and server POS software **16** is adding host computer system **12** to an existing network including client terminals **22A** from a different provider, the provider of servers may wish to write and store isolation layer **14** in host computer system **12**.

Isolation layer **14** preferably includes one library file for all system specific applications **17** or individual library files for system specific applications **17**. Use of multiple libraries adds network design flexibility but penalizes operation by loading each library into memory, even if only part of isolation layer **14** is used. Isolation layer **14** may take other forms including a single application, a single driver, or multiple drivers that perform read operations into server POS software **16** in order to obtain and map information to be used by client POS software **24A**. These drivers also perform writes into server POS software **16** may information created by client POS software **24A**.

Storage medium **18** stores PLU file **20** and other information files and is preferably a fixed disk drive.

Turning now to FIG. **2**, the software architecture within system **10** is described in more detail. Discussion references client POS software **24A**, but other independent applications **23A** operate in a similar fashion.

Client POS software **24A** includes POS checkout application software **30** and interprocess communications (IPC) software **32**. POS checkout software **30** records items by scanning them, prints them on a receipt, and adds their prices to produce a total transaction amount. POS checkout software **30** also performs specific transaction-related functions, such as processing food stamps and other forms of payment under government entitlement programs, calculating service charges, performing price change functions, performing out of transaction functions (loans, pickups), etc. An example of POS checkout software **30** is the UNITY® checkout application developed and sold by the Assignee of the present invention.

IPC software **32** is software that is added to POS checkout software **30** to facilitate interprocess communications (IPC) between POS checkout software **30** and isolation layer **14**. IPC software **32** will vary with operating system, from threads, queues, named pipes, shared files, sockets, etc.

Isolation layer **14** includes read/write engine and translation layer **34** and read/write layer **36**. Read/write engine and translation layer **34** translates one request into one or more read/write requests as needed for host computer system **12** based on standard server inputs and custom outputs. Read/write layer **36** performs the actual reads and writes to host computer system **12** based on the requests issued in read/write engine and translation layer **34**.

An example of read/write layer **36** is the UNITY® file service layer.

Server POS software **16** may include both the UNITY® file service layer and the UNIX® file system.

In a traditional POS system, client POS software communicates directly with POS server software through a communication methodology, such as named pipes. Under Applicant's invention, client POS software **24A** is modified to include IPC software **32**, and an isolation layer **14** is added. Communication between client POS software **24A** and POS server software **16** occurs between IPC software and isolation layer **14**.

Turning now to FIG. **3**, the components that are used to construct isolation layer **14** include templates **60**, object code **62**, example mappings and translations **64**, target routines **66**, and make files **68** for each of system-specific applications **17**. Templates **60** are a super set of example mappings and translations **64** and target routines **66**. Isolation layer **14** includes templates **60** for each of system specific application **17**. Example mappings and translations **64** and target routines **66** are edited and customized by developers to produce customized mappings and translations **65** and customized target routines **67** (customized templates **61**). Customized mappings and translations **65** and customized target routines **67** along with object code **62** are fed into make files **68** in order to produce executable code for isolation layer **14**.

Object code **62** consists of libraries **41** for each of system specific applications **17**. As mentioned above, isolation layer **14** may include one library for establishing communication for all system specific applications **17** or a plurality of separate libraries.

Example mappings and translations **64** provide non-displayable information (in code and/or in documentation) and may be edited to customize isolation layer **14**.

5

Target routines 66 are system specific application routines that may be edited and customized to suit the needs of the target transaction establishment.

Make files 68 are files that pass source code (templates 60) through compiler 72 and then pass object code 62 through linker 74 to produce the target executable.

Turning now to FIG. 4, the method of creating isolation layer 14 begins with START 80.

In step 82, independent applications 23A are isolated from system specific applications 17 through a non-displayable mapping determination and translation process. Example mappings and translations 64 result.

In step 84, a template 60 for one of system specific applications 17 is provided. Template 60 is packaged in an installable format that can then be distributed and installed on the target system.

In step 86, customized mappings and translations 65 and customized target routines 67 are determined.

In steps 88-92, customized template 61 is produced by changing template 60 to reflect customized mappings and translations 65.

In step 88, functions to be implemented are edited. For example, for PLU file reader software 40, the "read first", "read next", and "read specific" sections are edited. The "read first" section is a routine that must be completed/customized to read the first PLU in PLU file 20. The "read next" section is a routine that must be completed/customized to read the next PLU in PLU file 20 sequentially. The "read specific" section is a routine that must be completed/customized to read a specific PLU file record.

In step 90, customized mapping and translations 65 and customized target routines 67 derived from step 86 are applied to template 60.

In step 92, operation returns to step 84 if library functionality for an additional specific application 17 must be added. Otherwise, operation continues to step 94.

In step 94, a new executable (isolation layer 14) is compiled and linked from object code 62 and templates 61 for each of system specific applications 17 using make files 68, compiler 72, and linker 74.

In step 96, the method ends.

Referring now to FIG. 5, operation of a system specific application 17 is illustrated in detail, beginning with START 100. In particular, operation illustrates a method of transferring information between a terminal 22A and host computer system 12 through isolation layer 14.

In step 102, a system specific application 17 starts up and configures itself.

In step 104, system specific application 17 waits for an interprocess communications message from isolation layer 14. Interprocess communication services (e.g., "queues" for UNIX, threads or pipes for OS/2) are provided by the operating system executed by host computer system 12.

Isolation layer 14 calls on the operating system to send a request from client POS software 24A to system specific application 17. Client POS software 24A issues a call to isolation layer 14, instructing isolation layer 14 to perform a routine provided by isolation layer 14 and related to the purpose of system specific application 17.

For example, when system specific application 17 is PLU file reader software 40, the routine would be a "read direct" routine within isolation layer 14. Isolation layer 14 calls on the operating system to send the interprocess communications message to system specific application 17.

6

If such a message is received, system specific application 17 determines whether the interprocess communications message contains an "exit" command in step 105. If it does, system specific application 17 terminates in step 116.

If the interprocess communications message does not contain an "exit" command, system specific application 17 obtains the information from its source in step 106. For example, if system specific application 17 is PLU file reader software 40, system specific application 17 uses operating system 52 to obtain the information from PLU file 20.

In step 114, system specific application 17 calls on the operating system to return the information via interprocess communications to the client POS software 24A via isolation layer 14. System specific application 17 returns to a waiting state in step 104.

After system specific application 17 sends the information to isolation layer 14, isolation layer 14 passes the information to client POS software 24A. Client POS software application 24A can then display the information, compare the information, or otherwise examine the information in accordance with the functions of client POS software 24A.

Although the present invention has been described with particular reference to certain preferred embodiments thereof, variations and modifications of the present invention can be effected within the spirit and scope of the following claims.

What is claimed is:

1. A transaction system comprising:

a client computer system provided by a first seller of computer systems, including a client software application; *and*

a server computer system provided by a second seller of computer systems different from the first seller of computer systems, including a server software application which provides information from the server computer system to the client computer system and isolation layer software which facilitates communication between the client software application and the server software application to transfer the information *between the client software application and the server software application, the isolation layer providing translation for commands and data communicated between the client software application and the server software application, the isolation layer including a translation layer producing custom outputs for communicating between the client software application and the server software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the client software application and the server software application.*

2. A transaction system comprising:

a server computer system provided by a first seller of computer systems, including a server software application; *and*

a client computer system provided by a second seller of computer systems different from the first seller of computer systems, including a client software application which obtains information from the server computer system and isolation layer software which facilitates communication between the client software application and the server software application to transfer the information *between the client software application and the server software application, the isolation layer providing translation for commands and data communicated between the client software application and the server*

7

software application, the isolation layer including a translation layer producing custom outputs for communicating between the client software application and the server software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the client software application and the server software application.

3. A method of transferring information between a first software application and a second software application comprising the steps of:

- (a) providing the first software application by a first company;
- (b) providing the second software application by a second company different from the first company;
- (c) executing the first software application by a first computer;
- (d) executing the second software application by a second computer different from the first computer;
- (e) providing an isolation layer having hooks into the second software application, *the isolation layer providing translation for commands and data communicated between the first software application and the second software application, the isolation layer including a translation layer producing custom outputs for communicating between the first software application and the second software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the first software application and the second software application;*
- (f) executing the isolation layer; and
- (g) establishing communication between the first software application and the second software application by the isolation layer to transfer the information.

4. The method as recited in claim 3, wherein step f comprises the substep of:

- (f) executing the isolation layer by the first computer.

5. The method as recited in claim 3, wherein step f comprises the substep of:

- (f) executing the isolation layer by the second computer.

6. A method of transferring information between a client software application and a server software application comprising the steps of:

- (a) installing a client computer by a first company;
- (b) providing and installing the client software application by the first company;
- (c) installing a server computer by a second company different from the first company after installation of the client computer and the client software by the first company;
- (d) providing and installing the server software application by the second company;
- (e) providing and installing an isolation layer having hooks into the server software application by the second company, *the isolation layer providing translation for commands and data communicated between the client software application and the server software application, the isolation layer including a translation layer producing custom outputs for communicating between the client software application and the server software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the client software application and the server software applications;*

8

- (f) executing the client software application by the client computer;
- (g) executing the server software application and the isolation layer by the server computer; and
- (h) establishing communication between the client software application and the server software application by the isolation layer to transfer the information.

7. A method of transferring information between a client software application and a server software application comprising the steps of:

- (a) installing a server computer by a first company;
- (b) providing and installing the server software application by the first company;
- (c) installing a client computer by a second company different from the first company after installation of the server computer and the server software by the first company;
- (d) providing and installing the client software application by the second company;
- (e) providing and installing an isolation layer having hooks into the server software application by the second company, *the isolation layer providing translation for commands and data communicated between the client software application and the server software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the client software application and the server software application;*
- (f) executing the client software application and the isolation layer by the client computer;
- (g) executing the server software application by the server computer; and
- (h) establishing communication between the client software application and the server software application by the isolation layer to transfer the information.

8. A method of obtaining information for a client software application from a server software application comprising the steps of:

- (a) providing the client software application from a first company;
- (b) providing a server software application for accessing the information by a second company different from the first company;
- (c) providing an isolation layer between the client software application and the server software application, *the isolation layer providing translation for commands and data communicated between the client software application and the server software application, the isolation layer including a translation layer producing custom outputs for communicating between the client software application and the server software application, the custom outputs being based on customized mappings and translations directed to facilitating communication between the client software application and the server software application;*
- (d) sending a call for the information to the isolation layer by the client software application;
- (e) sending a request message for the information to the server software application by the isolation layer;
- (f) instructing control software to obtain the information by the server software application;

- (g) sending the information to the isolation layer by the server software application; and
 (h) sending the information to the client software application by the isolation layer.

9. The transaction system of claim 1 in which the client computer system comprises a first POS client terminal communicating with the server computer system through the isolation layer software, and in which the transaction system further comprises a second client computer system including a second POS client terminal that communicates with the server software application on the server computer system without requiring isolation layer software as required by the first POS client terminal.

10. The transaction system of claim 9 in which the first and second POS client terminals respectively run different POS client software applications from different developers and with different requirements for communicating with the server computer system.

11. The transaction system of claim 1 in which the client computer system comprises a first POS client terminal communicating with the server computer system through the isolation layer software, and in which the transaction system further comprises a second client computer system including a second POS client terminal communicating with the server computer system through different, custom isolation layer software.

12. The transaction system of claim 11 in which the first and second POS client terminals respectively run different POS client software applications from different developers and with different requirements for communicating with the server computer system.

13. The transaction system of claim 1 in which the client computer system comprises a first POS client terminal communicating with the server computer system through the isolation layer software, and in which the transaction system further comprises a second client computer system including a second POS client terminal communicating with the server computer system through different, custom isolation layer software.

14. The transaction system of claim 13 in which the first and second POS client terminals respectively run different POS client software applications from different developers and with different requirements for communicating with the server computer system.

15. A method of providing the transaction system of claim 1 in which the client software application and the server software application are POS software, and enabling client terminals with POS client application software from different developers to be connected to and communicate with the server software application on the server computer system, which comprises the following steps:

providing the client computer system including a client terminal adding to an existing POS system network of which the server computer system is a part, where the existing network includes a second client computer system with a client terminal running a POS client software application not requiring the isolation layer software to communicate with the server software application on the server computer system; and

providing the isolation layer software to provide translation services for the client terminal adding to the existing POS system network, to facilitate communications with the server software application on the server computer system.

16. A method of providing the transaction system of claim 1 in which the client software application and the server software application are POS software, comprising the step

of providing the client computer system including a client terminal adding to an existing POS system network in which the server computer system is included, and in which the network includes a second client computer system, including a client terminal running a POS client software application not requiring the isolation layer software to communicate with the server software application running on the server computer system.

17. A method of providing the transaction system of claim 1, in which the client software application and the server software application are POS software, and enabling client terminals with POS client application software from different developers to be connected to and communicate with the server software application on the server computer system, comprising the step of providing the client computer system including a first POS client terminal adding to an existing POS system network of which the server computer system is a part, where the existing network includes a second client computer system with a second POS client terminal running a POS client software application requiring isolation layer software different than required by the first client terminal to communicate with the server software application on the server computer system.

18. A method of providing the transaction system of claim 1 in which the client software application and the server software application are POS software, comprising the step of providing the server computer system as part of a host computer system added to an existing POS system network in which the client computer system is included.

19. The method of claim 18 that comprises the further steps of writing the isolation layer software to provide translation services between the client software application and the server software application, and storing the isolation layer software on the host computer system added to the existing POS system network.

20. The method of claim 18 in which the isolation layer software provides read and write services for the client software application.

21. The method of claim 18 in which the client computer system includes IPC software facilitating inter-process communications between the client software application and the isolation layer software.

22. The transaction system of claim 2 in which the client computer system comprises a first POS client terminal communicating with the server computer system through the isolation layer software, and in which the transaction system further comprises a second client computer system including a second POS client terminal that communicates with the server software application on the server computer system without requiring isolation layer software as required by the first POS client terminal.

23. The transaction system of claim 22 in which the first and second POS client terminals respectively run different POS client software applications from different developers and with different requirements for communicating with the server computer system.

24. A method of providing the transaction system of claim 2 in which the client software application and the server software application are POS software, and enabling client terminals with POS client application software from different developers to be connected to and communicate with the server software application on the server computer system, which comprises the following steps:

providing the client computer system including a client terminal adding to an existing POS system network of which the server computer system is a part, where the existing network includes a second client computer sys-

11

tem with a client terminal running a POS client software application not requiring the isolation layer software to communicate with the server software application on the server computer system; and

providing the isolation layer software to provide translation services for the client terminal adding to the existing POS system network, to facilitate communications with the server software application on the server computer system.

25. A method of providing the transaction system of claim 2 in which the client software application and the server software application are POS software, comprising the step of providing the client computer system including a client terminal adding to an existing POS system network in which the server computer system is included, and in which the network includes a second client computer system, including a client terminal running a POS client software application not requiring the isolation layer software to communicate with the server software application running on the server computer system.

26. A method of providing the transaction system of claim 2, in which the client software application and the server software application are POS software, and enabling client terminals with POS client application software from different developers to be connected to and communicate with the server software application on the server computer system, comprising the step of providing the client computer system including a first POS client terminal adding to an existing POS system network of which the server computer system is a part, where the existing network includes a second client computer system with a second POS client terminal running a POS client software application requiring isolation layer software different than required by the first client terminal to communicate with the server software application on the server computer system.

27. A method of providing the transaction system of claim 2 in which the client software application and the server software application are POS software, comprising the step of providing the client computer system including a client

12

terminal added to an existing POS system network in which the server computer system is included.

28. The method of claim 27 in which the isolation layer software provides read and write services for the client software application.

29. The method of claim 27 in which the client computer system includes IPC software facilitating inter-process communications between the client software application and the isolation layer software.

30. The method of claim 27 that comprises the further steps of writing the isolation layer software to provide translation services between the client software application and the server software application, and storing the isolation layer software on the client terminals added to the existing POS system network.

31. The method of claim 30 in which the isolation layer software is written by compiling sample target routines that have been edited and customized to meet the needs of a target transaction establishment.

32. The method of claim 30 in which the client computer system includes a plurality of different independent software applications running on a plurality of different client terminals from different developers.

33. The method of claim 32 in which the client computer system includes client terminals running independent software applications developed by the same developer as the server software application and not requiring isolation layer software to communicate with the server software application, and client terminals running independent software applications developed by a developer different from the server software application and using isolation layer software to communicate with the server software application.

34. The method of claim 29 for which communications between the server computer system and the added client POS computer terminal occurs between interprocess communications software and the isolation layer software.

* * * * *