US005915019A

United States Patent [19]

Ginter et al.

[11] Patent Number: 5,915,019

[45] Date of Patent: Jun. 22, 1999

[54] **SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION**

[75] Inventors: **Karl L. Ginter**, Beltsville; **Victor H. Shear**, Bethesda, both of Md.; **Francis J. Spahn**, El Cerrito; **David M. Van Wie**, Sunnyvale, both of Calif.

[73] Assignee: **InterTrust Technologies Corp.**, Sunnyvale, Calif.

[21] Appl. No.: **08/780,393**

[22] Filed: **Jan. 8, 1997**

Related U.S. Application Data

[62] Division of application No. 08/388,107, Feb. 13, 1995, abandoned.

[51] **Int. Cl.**[6] **H04L 9/00**

[52] **U.S. Cl.** **380/4**; 380/21; 380/49; 395/680; 705/26; 705/400

[58] **Field of Search** 380/3, 4, 5, 21, 380/49; 395/680, 683; 705/26, 400

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,573,747	4/1971	Adams et al. .
3,609,697	9/1971	Blevins .

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

9 004 79	12/1984	Belgium .
0 84 441	7/1983	European Pat. Off. .
0128672	12/1984	European Pat. Off. .
A0135422	3/1985	European Pat. Off. .
0180460	5/1986	European Pat. Off. .
0 370 146	11/1988	European Pat. Off. .
0399822A2	11/1990	European Pat. Off. .
0421409A2	4/1991	European Pat. Off. .
0 456 386 A2	11/1991	European Pat. Off. .

(List continued on next page.)

OTHER PUBLICATIONS

Applications Requirements for Innovative Video Programming; How to Foster (or Cripple) Program Development Opportunities for Interactive Video Programs Delivered on Optical Media; A Challenge for the Introduction of DVD (Digital Video Disc) (Oct. 19–20, 1995, Sheraton Universal Hotel, Universal City CA).

Arneke, David, et al., News Release, AT&T, Jan. 9, 1995, AT&T encryption system protects information services, 1 page.

AT&T Technology, vol. 9, No. 4, New Products, Systems and Services, pp. 16–19, Undated.

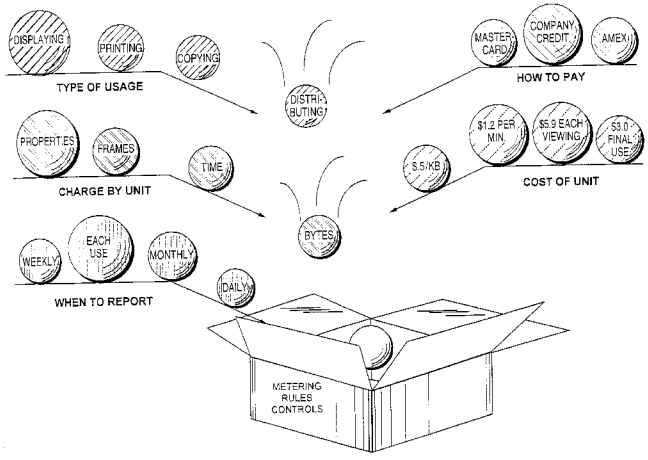
Barassi, Theodore Sedgwick, Esq., The Cybernotary: Public Key Registration and Certificaiton and Authentication of International Legal Transactions, 4 pages, Undated.

Primary Examiner—Gilberto Barrón, Jr.
Attorney, Agent, or Firm—Nixon & Vanderhye P.C.

[57] **ABSTRACT**

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

101 Claims, 146 Drawing Sheets



U.S. PATENT DOCUMENTS

3,796,830	3/1974	Smith .	
3,798,359	3/1974	Feistel .	
3,798,360	3/1974	Feistel .	
3,798,605	3/1974	Feistel .	
3,806,882	4/1974	Clarke .	
3,829,833	8/1974	Freeny, Jr. .	
3,906,448	9/1975	Henriques .	
3,911,397	10/1975	Freeny, Jr. .	
3,924,065	12/1975	Freeny, Jr. .	
3,931,504	1/1976	Jacoby .	
3,946,220	3/1976	Brobeck et al. .	
3,956,615	5/1976	Anderson et al. .	
3,958,081	5/1976	Ehrsam et al. .	
3,970,992	7/1976	Boothroyd et al. .	
4,048,619	9/1977	Forman, Jr. et al. .	
4,071,911	1/1978	Mazur .	
4,112,421	9/1978	Freeny, Jr. .	
4,120,030	10/1978	Johnstone .	
4,163,280	7/1979	Mori et al. .	
4,168,396	9/1979	Best .	
4,196,310	4/1980	Forman et al. .	
4,200,913	4/1980	Kuhar et al. .	
4,209,787	6/1980	Freeny, Jr. .	
4,217,588	8/1980	Freeny, Jr. .	
4,220,991	9/1980	Hamano et al. .	
4,232,193	11/1980	Gerard .	
4,232,317	11/1980	Freeny, Jr. .	
4,236,217	11/1980	Kennedy .	
4,253,157	2/1981	Kirschner et al. .	
4,262,329	4/1981	Bright et al. .	
4,265,371	5/1981	Desai et al. .	
4,270,182	5/1981	Asija .	
4,278,837	7/1981	Best .	
4,305,131	12/1981	Best .	
4,306,289	12/1981	Lumley .	
4,309,569	1/1982	Merkle .	
4,319,079	3/1982	Best .	
4,323,921	4/1982	Guillou .	
4,328,544	5/1982	Baldwin et al. .	
4,337,483	6/1982	Guillou .	
4,361,877	11/1982	Dyer et al. .	
4,375,579	3/1983	Davida et al. .	
4,433,207	2/1984	Best .	
4,434,464	2/1984	Suzuki et al. .	
4,442,486	4/1984	Mayer .	
4,446,519	5/1984	Thomas .	
4,454,594	6/1984	Heffron et al. .	
4,458,315	7/1984	Uchenick .	
4,462,076	7/1984	Smith, III .	
4,462,078	7/1984	Ross .	
4,465,901	8/1984	Best .	
4,471,163	9/1984	Donald et al. .	
4,484,217	11/1984	Block et al. .	
4,494,156	1/1985	Kadison et al. .	
4,513,174	4/1985	Herman .	
4,528,588	7/1985	Lofberg .	
4,528,643	7/1985	Freeny, Jr. .	
4,553,252	11/1985	Egendorf .	
4,558,176	12/1985	Arnold et al. .	
4,558,413	12/1985	Schmidt et al. .	
4,562,306	12/1985	Chou et al. .	
4,562,495	12/1985	Bond et al. .	
4,577,289	3/1986	Comerford et al. .	
4,584,641	4/1986	Guglielmino .	
4,588,991	5/1986	Atalla .	
4,589,064	5/1986	Chiba et al. .	
4,593,353	6/1986	Pickholtz .	
4,593,376	6/1986	Volk .	
4,595,950	6/1986	Lofberg .	
4,597,058	6/1986	Izumi et al. .	
4,634,807	1/1987	Chorley et al. .	
4,644,493	2/1987	Chandra et al. .	
4,646,234	2/1987	Tolman et al. .	
4,652,990	3/1987	Pailen et al. .	
4,658,093	4/1987	Hellman .	
4,670,857	6/1987	Rackman .	
4,672,572	6/1987	Alsberg .	
4,677,434	6/1987	Fascenda .	
4,680,731	7/1987	Izumi et al. .	
4,683,553	7/1987	Mollier .	
4,685,056	8/1987	Barnsdale et al. .	
4,688,169	8/1987	Joshi .	
4,691,350	9/1987	Kleijne et al. .	
4,696,034	9/1987	Wiedemer .	
4,701,846	10/1987	Ikeda et al. .	
4,712,238	12/1987	Gilhousen et al. .	
4,713,753	12/1987	Boebert et al.	380/4 X
4,740,890	4/1988	William .	
4,747,139	5/1988	Taaffe .	
4,757,533	7/1988	Allen et al. .	
4,757,534	7/1988	Matyas et al. .	
4,768,087	8/1988	Taub et al. .	
4,791,565	12/1988	Dunham et al. .	
4,796,181	1/1989	Wiedemer .	
4,799,156	1/1989	Shavit et al. .	
4,807,288	2/1989	Ugon et al. .	
4,817,140	3/1989	Chandra et al. .	
4,823,264	4/1989	Deming .	
4,827,508	5/1989	Shear .	
4,858,121	8/1989	Barber et al. .	
4,864,494	9/1989	Kobus .	
4,868,877	9/1989	Fischer .	
4,903,296	2/1990	Chandra et al. .	
4,924,378	5/1990	Hershey et al. .	
4,930,073	5/1990	Cina, Jr. .	
4,949,187	8/1990	Cohen .	
4,977,594	12/1990	Shear .	
4,999,806	3/1991	Chernow et al. .	
5,001,752	3/1991	Fischer .	
5,005,122	4/1991	Griffin et al. .	
5,005,200	4/1991	Fischer .	
5,010,571	4/1991	Katznelson	380/4
5,023,907	6/1991	Johnson et al.	380/4
5,047,928	9/1991	Wiedemer .	
5,048,085	9/1991	Abraham et al. .	
5,050,213	9/1991	Shear .	
5,091,966	2/1992	Bloomberg et al. .	
5,103,392	4/1992	Mori .	
5,103,476	4/1992	Waite et al. .	
5,111,390	5/1992	Ketcham .	
5,119,493	6/1992	Janis et al. .	
5,128,525	7/1992	Stearns et al. .	
5,136,643	8/1992	Fischer .	
5,136,646	8/1992	Haber et al. .	
5,136,647	8/1992	Haber et al. .	
5,136,716	8/1992	Harvey et al. .	
5,146,575	9/1992	Nolan, Jr. .	
5,148,481	9/1992	Abraham et al. .	
5,155,680	10/1992	Wiedemer .	
5,168,147	12/1992	Bloomberg .	
5,185,717	2/1993	Mori .	
5,201,046	4/1993	Goldberg et al. .	
5,201,047	4/1993	Maki et al. .	
5,208,748	5/1993	Flores et al. .	
5,214,702	5/1993	Fischer .	
5,216,603	6/1993	Flores et al. .	
5,221,833	6/1993	Hecht .	
5,222,134	6/1993	Waite et al. .	
5,224,160	6/1993	Paulini et al. .	
5,224,163	6/1993	Gasser et al. .	
5,235,642	8/1993	Wobber et al. .	
5,245,165	9/1993	Zhang .	

5,247,575	9/1993	Sprague et al. .	
5,260,999	11/1993	Wyman .	
5,263,158	11/1993	Janis	395/600
5,265,164	11/1993	Matyas et al. .	
5,276,735	1/1994	Boebert et al. .	
5,280,479	1/1994	Mary .	
5,285,494	2/1994	Sprecher et al. .	
5,301,231	4/1994	Abraham .	
5,311,591	5/1994	Fischer	380/4
5,319,705	6/1994	Halter et al. .	
5,337,360	8/1994	Fischer .	
5,341,429	8/1994	Stringer et al. .	
5,343,527	8/1994	Moore .	
5,347,579	9/1994	Blandford .	
5,351,293	9/1994	Michener et al. .	
5,355,474	10/1994	Thuraisngham et al. .	
5,373,561	12/1994	Haber et al. .	
5,390,247	2/1995	Fischer	380/25
5,390,330	2/1995	Talati .	
5,392,220	2/1995	van den Hamer et al. .	
5,392,390	2/1995	Crozier .	
5,394,469	2/1995	Nagel et al. .	
5,410,598	4/1995	Shear .	
5,412,717	5/1995	Fischer .	
5,421,006	5/1995	Jablon .	
5,422,953	6/1995	Fischer .	
5,428,606	6/1995	Moskowitz .	
5,438,508	8/1995	Wyman	380/4 X
5,442,645	8/1995	Ugon .	
5,444,779	8/1995	Daniele .	
5,449,895	9/1995	Hecht et al. .	
5,449,896	9/1995	Hecht et al. .	
5,450,493	9/1995	Maher .	
5,453,601	9/1995	Rosen .	
5,453,605	9/1995	Hecht et al. .	
5,455,407	10/1995	Rosen .	
5,455,861	10/1995	Faucher et al. .	
5,455,953	10/1995	Russell .	
5,457,746	10/1995	Dolphin .	
5,463,565	10/1995	Cookson et al. .	
5,473,687	12/1995	Lipscomb et al. .	
5,473,692	12/1995	Davis .	
5,479,509	12/1995	Ugon .	
5,485,622	1/1996	Yamaki .	
5,491,800	2/1996	Goldsmith et al. .	
5,497,479	3/1996	Hornbuckle .	
5,497,491	3/1996	Mitchell et al. .	
5,499,298	3/1996	Narasimhalu et al. .	
5,504,757	4/1996	Cook et al. .	
5,504,818	4/1996	Okano	380/49
5,504,837	4/1996	Griffeth et al. .	
5,508,913	4/1996	Yamamoto et al. .	
5,509,070	4/1996	Schull	380/4
5,513,261	4/1996	Maher .	
5,530,235	6/1996	Stefik et al. .	
5,530,752	6/1996	Rubin .	
5,533,123	7/1996	Force et al. .	
5,534,975	7/1996	Stefik et al. .	
5,537,526	7/1996	Anderson et al. .	
5,539,735	7/1996	Moskowitz .	
5,539,828	7/1996	Davis .	
5,550,971	8/1996	Brunner et al. .	
5,553,282	9/1996	Parrish et al. .	
5,557,518	9/1996	Rosen	364/408
5,563,946	10/1996	Cooper et al.	380/4
5,568,552	10/1996	Davis .	
5,572,673	11/1996	Shurts .	
5,592,549	1/1997	Nagel et al. .	
5,606,609	2/1997	Houser et al.	380/4
5,613,004	3/1997	Cooperman et al. .	
5,621,797	4/1997	Rosen .	
5,629,980	5/1997	Stefik et al. .	
5,633,932	5/1997	Davis et al. .	
5,634,012	5/1997	Stefik et al. .	
5,636,292	6/1997	Rhoads .	
5,638,443	6/1997	Stefik et al.	380/4
5,638,504	6/1997	Scott et al. .	
5,640,546	6/1997	Gopinath et al. .	
5,655,077	8/1997	Jones et al. .	
5,687,236	11/1997	Moskowitz et al. .	
5,689,587	11/1997	Bender et al. .	
5,692,180	11/1997	Lee .	
5,710,834	1/1998	Rhoads .	
5,740,549	4/1998	Reilly et al. .	
5,745,604	4/1998	Rhoads .	
5,748,763	5/1998	Rhoads .	
5,748,783	5/1998	Rhoads .	
5,748,960	5/1998	Fischer	395/683
5,754,849	5/1998	Dyer et al. .	
5,757,914	5/1998	McManis .	
5,758,152	5/1998	LeTourneau .	
5,765,152	1/1998	Erickson .	
5,768,426	6/1998	Rhoads .	

FOREIGN PATENT DOCUMENTS

0 469 864 A2	2/1992	European Pat. Off. .
0 565 314 A2	10/1993	European Pat. Off. .
0 593 305 A2	4/1994	European Pat. Off. .
0 651 554 A1	5/1995	European Pat. Off. .
0 668 695 A2	8/1995	European Pat. Off. .
0 725 376	1/1996	European Pat. Off. .
0 695 985 A1	2/1996	European Pat. Off. .
0 696 798 A1	2/1996	European Pat. Off. .
0715243A1	6/1996	European Pat. Off. .
0715244A1	6/1996	European Pat. Off. .
0715245A1	6/1996	European Pat. Off. .
0715246A1	6/1996	European Pat. Off. .
0715247A1	6/1996	European Pat. Off. .
0749081A1	12/1996	European Pat. Off. .
0 778 513 A2	6/1997	European Pat. Off. .
0 795 873 A2	9/1997	European Pat. Off. .
3803982A1	1/1990	Germany .
57-726	5/1982	Japan .
62-241061	10/1987	Japan .
01-068835	3/1989	Japan .
64-68835	3/1989	Japan .
02-242352	9/1990	Japan .
02-247763	10/1990	Japan .
02-294855	12/1990	Japan .
04-369068	12/1992	Japan .
05-181734	7/1993	Japan .
05-257783	10/1993	Japan .
05-268415	10/1993	Japan .
06-175794	6/1994	Japan .
06-215010	8/1994	Japan .
6225059	8/1994	Japan .
07-056794	3/1995	Japan .
07-084852	3/1995	Japan .
07-141138	6/1995	Japan .
07-200317	8/1995	Japan .
07-200492	8/1995	Japan .
07-244639	9/1995	Japan .
08-137795	5/1996	Japan .
08-152990	6/1996	Japan .
08-185298	7/1996	Japan .
A2136175	9/1984	United Kingdom .
2264796	9/1993	United Kingdom .
2294348	4/1996	United Kingdom .
2295947	6/1996	United Kingdom .
WOA8502310	5/1985	WIPO .
WO 85/03584	8/1985	WIPO .
WO 90/02382	3/1990	WIPO .
WO92/06438	4/1992	WIPO .
WO92/22870	12/1992	WIPO .

WO93/01550	1/1993	WIPO .
WO94/01821	1/1994	WIPO .
WO94/03859	2/1994	WIPO .
WO9406103	3/1994	WIPO .
WO 94/16395	7/1994	WIPO .
WO 94/18620	8/1994	WIPO .
WO 94/22266	9/1994	WIPO .
WO 94/27406	11/1994	WIPO .
WO95/14289	6/1995	WIPO .
WO 96/00963	1/1996	WIPO .
WO 96/03835	2/1996	WIPO .
WO 96/05698	2/1996	WIPO .
WO 96/06503	2/1996	WIPO .
WO96/13013	5/1996	WIPO .
WO96/21192	7/1996	WIPO .
WO97/03423	1/1997	WIPO .
WO97/07656	3/1997	WIPO .
WO97/32251	9/1997	WIPO .
WO 97/48203	12/1997	WIPO .

OTHER PUBLICATIONS

Bruner, Rick E., PowerAgent, NetBot help advertisers reach Internet shoppers, Aug. 1997 (Document from Internet).

CD ROM, Introducing . . . The Workflow CD–ROM Sampler, Creative Networks, MCIMail: Creative Networks, Inc., Palo Alto, California, Undated.

Clark, Tim, Ad service gives cash back, www.news.com, Aug. 4, 1997, 2 pages (Document from Internet).

Communications of the ACM, Jun. 1996, vol. 39, No. 6.

Cunningham, Donna, et al., News Release, AT&T, Jan. 31, 1995, AT&T, VLSI Technology join to improve info highway security, 3 pages.

Data Sheet, About the Digital Notary Service, Surety Technologies, Inc., 1994–95, 6 pages.

Dempsey, et al., *D–Lib Magazine,* Jul./Aug. 1996 The Warwick Metadata Workshop: A Framework for the Deployent of Resource Description, Jul. 15, 1966.

Document from Internet, cgi@ncsa.uiuc.edu, CGI Common Gateway Interface, 1 page, 1996.

Firefly Network, Inc., www.ffly.com, What is Firefly? Firefly revision: 41.4 Copyright 1995, 1996.

Gleick, James, "Dead as a Dollar" *The New York Times Magazine,* Jun. 16, 1996, Section 6, pp. 26–30, 35, 42, 50, 54.

Greguras, Fred, Softic Symposium '95, Copyright Clearances and Moral Rights, Nov. 30, 1995 (as updated Dec. 11, 1995), 3 pages.

Harman, Harry H., *Modern Factor Analysis,* Third Edition Revised, University of Chicago Press Chicago and London, Third revision published 1976.

Herzberg, Amir et al., Public Protection of Software, *ACM Transactions on Computer Systems,* vol. 5, No. 4, Nov. 1987, pp. 371–393.

Holt, Stannie, Start–up promises user confidentiality in Web marketing service, *Info World Electric,* Aug. 13, 1997 (Document from Internet).

Hotjava™: The Security Story, 4 pages, Undated.

Invoice? What is an Invoice? *Business Week,* Jun. 10, 1996.

Javasoft, Frequently Asked Questions—Applet Security, What's Java™? Products and Services, Java/Soft News, Developer's Cornier,Jun. 7, 1996, 8 pages.

Jiang, et al, A concept–Based Approach to Retrieval from an Electronic Industrialn Directory, *International Journal of Electronic Commerce,* vol. 1, No. 1, Fall 1996, pp. 51–72.

Jones, Debra, Top Tech Stories, PowerAgent Introducts First Internet 'Infomediary' to Empower and Protect Consumers, Aug. 13, 1997 3 pages (Document from Internet).

Kohntopp, M., Sag's durch die Blume, Apr. 1996, marit@schulung.netuse.de.

Lagoze, Carl, *D–Lib Magazine,* Jul./Aug. 1996, The Warwick Framework, A Container Architecture for Diverse Sets of Metadata.

Maclachlan, Malcolm, PowerAgent Debuts Spam–Free Marketing, *TechWire,* Aug. 13, 1997, 3 pages (Document from Internet), Undated.

Milbrandt, E., Stenanography Info and Archive, 1996.

Mossberg, Walter S., Personal Technology, Threats to Privacy On–Line Become More Worrisome, *Wall Street Journal,* Oct. 24, 1996.

Negroponte, Electronic Word of Mouth, *Wired* Oct. 1996, p. 218.

News Release, Premenos Announces Templar 2.0—Next Generation Software for Secure Internet EDI, webmaster@templar.net, 1 page, Jan. 17, 1996.

News Release, *The Document Company Xerox,* Xerox Announces Software Kit for Creating Working Documents with Dataglyphs, Nov. 6, 1995, Minneapolis, MN, 13 pages.

PowerAgent Inc., Proper Use of Consumer Information on the Internet White Paper, Jun. 1997, Document from Internet, 9 pages (Document from Internet).

PowerAgent Press Releases, What the Experts are Reporting on PowerAgent, Aug. 13, 1997, 6 pages (Document from Internet).

PowerAgent Press Releases, What the Experts are Reporting on PowerAgent, Aug. 4, 1997, 5 pages (Document from Internet).

PowerAgent Press Releases, What the Experts are Reporting on PowerAgent, Aug. 13, 1997, 3 pages (Document from Internet).

Premenos Corp. White Paper: The Future of Electronic Commerce, A Supplement to Midrange Systems, Internet webmaster@premenos.com, 4 pages, Undated.

Resnick, et al., Recommender Systems, *Communications of the ACM,* vol. 40, No. 3, Mar. 1997,pp. 56–89.

Rothstein, Edward, *The New York Times,* Technology, Connections, Making th eInternet come to you, through 'push' technology . . . p. D5, Jan. 20, 1997.

Rutkowski, Ken, PowerAgent Introduces First Internet 'Infomediary' to Empower and Protect Consumers, *Tech Talk News Story,* Aug. 4, 1997 (Document from Internet).

Sager, Ira (Edited by), Bits & Bytes, *Business Week,* Sep. 23, 1996, p. 142E.

Schurmann, Jurgen, *Pattern Classification, A Unified View of Statistical and Neural Approaches,* John Wiley & Sons, Inc., 1996.

Special Report, The Internet:Fulfilling the Promise The Internet: Bring Order From Chaos; Lynch, Clifford, Search the Internet; Resnick, Paul, Filtering Information on the Internet; Hearst, Marti A., Interfaces for Searching the Web; Stefik, Mark, Trusted Systems; *Scientific American,* Mar. 1997, pp. 49–56, 62–64, 68–72, 78–81.

Stefik, Mark, *Introduction to Knowledge Systems,* Chapter 7, Classification, pp. 543–607, 1995 by Morgan Kaufmann Publishers, Inc.

Templar Overview,: Premenos, Internet info@templar.net, 4 pages, Undated.

Templar Software and Services: Secure, Reliable, Standards–Based EDI Over the Internet, Prementos, Internet info@templar.net, 1 page, Undated.

Voight, Joan, Beyond the Banner, *Wired,* Dec. 1996, pp. 196, 200, 204.

Vonder Haar, Steven, PowerAgent Launches Commercial Service, *Inter@ctive Week,* Aug. 4, 1997 (Document from Internet).

Weber, Dr. Robert, Digital Rights Management Technologies, A Report to the International Federation of Reproduction Rights Organisations, Oct. 1995,pp. 1–49.

Weber, Dr. Robert, Digital Rights Management Technologies, Oct. 1995, 21 pages.

Wepin Store, Stenography (Hidden Writing) (Common Law 1995).

World Wide Web FAQ, How can I put an access counter on my home page?, 1 page, 1996.

Yellin, F. Low Level Security in Java, 8 pages, Undated.

IBM Technical Disclosure Bulletin, "Multimedia Mixed Object Envelopes Supporting a Graduated Fee Scheme via Encryption," vol. 37, No. 03, Mar. 1994, Armonk, NY.

IBM Technical Disclosure Bulletin, "Transformer Rules for Software Distribution Mechanism–Support Products," vol. 37, No. 04B, Apr. 1994, Armonk, NY.

Suida, Karl, *Mapping New Applications onto New Technologies,* "Security Services in Telecommunications Networks," Mar. 8–10, 1988, Zurich.

Portland Software's ZipLock, Internet information, Copyright Portland Software 1996–1997, 12 pages.

Dyson, Esther, "Intellectual Value," Wired Magazine, Jul. 1995, pp. 136–141 and 182–184.

Argent Information Q&A Sheet, http://www.digital–watermark.com/, Copyright 1995, The Dice Company, 7 pages.

Guillou, L.: "Smart Cards and Conditional Access", pp. 480–490 Advances in Cryptography, Proceedings of EuroCrypt 84 (Beth et al, Ed., Springer–Verlag 1985).

Rankine, G., "Thomas—A Complete Single–Chip RSA Device," Advances in Cryptography, Proceedings of Crypto 86, pp. 480–487 (A.M. Odlyzko Ed., Springer–Verlag 1987).

DSP56000/DSP56001 Digital Signal Processor User's Manual, Motorola, 1990, p. 2–2.

Dusse, Stephen R. and Burton S. Kaliski "A Cryptographic Library for the Motorola 56000" in Damgard, I. M., Advances in Cryptology—Proceedings Eurocrypt 90, Springer–Verlag, 1991, pp. 230–244.

Struif, Bruno "The Use of Chipcards for Electronic Signatures and Encryption" in : Proceedings for the 1989 Conference on VSLI and Computer Peripherals, IEEE Computer Society Press, 1989, pp. 4/155–4/158.

Ryoichi Mori and Masaji Kawahara, *The Transactions of the EIEICE, V.* "Superdistribution: The Concept and the Architecture," E73 (Jul. 1990), No. 7, Tokyo, Japan.

Stefik, "Internet Dreams: Archetypes, Myths, and Metaphors, Letting Loose the Light: Igniting Commerce in Electronic Publication," pp. 219–253, (1996) Massachusetts Institute of Technology.

Stefik, Mark, "Letting Loose the Light, Igniting Commerce in Electronic Publication," (1994, 1995) Palo Alto, California.

Shear, "Solutions for CD–ROM Pricing and Data Security Problems", pp. 530–533, *CD ROM Yearbook 1988–1989* (Microsoft Press 1988 or 1989).

Press Release, "National Semiconductor and EPR Partner For Information Metering/Data Security Cards" (Mar. 4, 1994).

"Electronic Publishing Resources Inc. Protecting Electronically Published Properties Increasing Publishing Profits" (Electronic Publishing Resources, 1991).

"The Benefits of ROI For Database Protection and Usage Based Billing" (Personal Library Software, 1987 or 1988).

ROI–Solving Critical Electronic Publishing Problems (Personal Library Software, 1987 or 1988).

Weber, "Metering Technologies for Digital Intellectual Property, A Report to the International Federation of Reproduction Rights Organisations," pp. 1–29; Oct. 1994, Boston, MA, USA.

ROI (Personal Library Software, 1987 or 1988).

DiscStore (Electronic Publishing Resources 1991).

Yee, "Using Secure Coprocessors," CMU–CS–94–149, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, Undated.

Tygar et al., "Dyad: A System for Using Physically Secure Coprocessors," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (undated).

Tygar et al., "Dyad: A System for Using Physically Secure Coprocessors," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (May 1991).

Maxemchuk, "Electronic Document Distribution," AT&T Bell Laboratories, Murry Hill, New Jersey 07974, Undated.

Choudhury, et al., "Copyright Protection for Electronic Publishing over Computer Networks," AT&T Bell Laboratores, Murray Hill, New Jersey 07974 (Jun. 1994).

Weingart, "Physical Security for the μABYSS System," IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 (1987).

White, "ABYSS: A Trusted Architecture for Software Protection," IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 (1987).

Neumann, et al., "A Provably Secure Operating System: The System, Its Applications, and Proofs," Computer Science Laboratory Report CSL–116, Second Edition, SRI International (May 1980).

Caruso, "Technology, Digital Commerce 2 plans for watermarks, which can bind proof of authorship to electronic works," *New York Times* (Aug. 1995).

"Electronic Currency Requirements, XIWT (Cross Industry Working Group)," no date.

"NII, Architecture Requirements, XIWT," no date.

Arthur K. Reilly, *Standards committee T1–Telecommunications*, Input to the 'International Telecommunications Hearings,' Panel 1: Component Technologies of the NII/GII, no date.

Dan Bart, Comments in the Matter of Public Hearing and Request for Comments on the International Aspects of the National Information Infrastructure, Aug. 12, 1994.

"Open System Environment Architectural Framework for National Information Infrastructure Services and Standards, in Support of National Class Distributed Systems," Distributed System Engineering Program Sponsor Group, Draft 1.0. Aug. 5, 1994.

"Information Infrastructure Standards Panel: NII 'The Information Superhighway'," NationsBank—HGDeal—ASC X9, 15 pages, Undated.

Jud Hofmann, "Interfacing the NII to User Homes," Electronic Industries Association, Consumer Electronic Bus Committee, 14 slides, no date.

"Framework for National Information Infrastructure Services," NIST, Jul. 1994, 12 slides.

Claude Baggett, "Cable's Emerging Role in the Information Superhighway," Cable Labs, 13 slides, Undated.

"IISP Break Out Session Report for Group No. 3, Standards Development and Tracking System," no date.
"XIWT Cross Industry Working Team," 5 pages, Jul. 1994.
"Computer Systems Policy Project (CSSP), Perspectives on the National Information Infrastructure: Ensuring Interoperability (Feb. 1994)," Feb. 1994.
"Framework for National Information Infrastructure Services," Draft, U.S. Department of Commerce, Jul. 1994.
"EIA and TIA White Paper on National Information Infrastructure,"published by the Electronic Industries Association and the Telecommunications Industry Association, Washington, D.C., no date.
Michael Baum, "Worldwide Electronic Commerce: Law, Policy and Controls Conference," program details, Nov. 11, 1993.
Bruce Sterling, "Literary freeware: Not for Commercial Use," remarks at Computers, Freedom and Privace Conference IV, Chicago, Mar. 26, 1994.
"The 1:1 Future of the Electronic Marketplace: Return to a Hunting and Gathering Society," 2 pages, no date.
D. Linda Garcia, testimony before a hearing on science, space and technology, May 26, 1994.
Wired 1.02, "Is Advertising Really dead?, Part 2," 1994.
Hugh Barnes, memo to Henry LaMuth, subject: George Gilder articles, May 31, 1994.
Daniel J. Weitzner, A Statement on EFF's Open Platform Campaign as of Nov., 1993, 3 pages.
"Serving the Community: A Public–Interest Vision of the National Information Infrastructure," Computer Professionals for Social Responsibility, Executive Summary, no date.
Steven Schlossstein, *International Economy,* "America: The G7's Comeback Kid," Jun./Jul. 1993.
Lance Rose, "Cyberspace and the Legal Matrix: Laws or Confusion?," 1991.
"Cable Television and America's Telecommunications Infrastructure," National Cable Television Association, Apr. 1993.
Adele Weder, "Life on the Infohighway," 4 pages, no date.
T. Valovic, *Telecommunications,* "The Role of Computer Networking in the Emerging Virtual Marketplace," pp. 40–44, Undated.
Dr. Joseph N. Pelton, *Telecommunications,* "Why Nicholas Negroponte is Wrong About the Future of Telecommunication," pp. 35–40, Jan. 1993.
Nicholas Negroponte, *Telecommunications,* "Some Thoughts on Likely and expected Communications scenarios: A Rebuttal," pp. 41–42, Jan. 1993.
Tom Stephenson, *Advanced Imaging,* "The Info Infrastructure Initiative: Data SuperHighways and You," pp. 73–74, May 1993.
Steve Rosenthal, *New Media,* "Mega Channels," pp. 36–46, Sep. 1993.
News Release, The White House, Office of the President, "Background on the Administration's Telecommunications Policy Reform Initiative," Jan. 11, 1994.
Steve Rosenthal, *New Media,* "Interactive Network: Viewers Get Involved," pp. 30–31, Dec. 1992.
Steve Rosenthal, *New Media,* "Interactive TV: The Gold Rush Is On," pp. 27–29, Dec. 1992.
EFFector Online vol. 6 No. 6, "A Publication of the Electronic Frontier Foundation," 8 pages, Dec. 6, 1993.
Mike Lanza, electronic mail, "George Gilder's Fifth Article—Digital Darkhorse—Newspapers," Feb. 21, 1994.
Steven Levy, *Wired,* "E–Money, That's What I Want," 10 pages, Dec. 1994.
Kevin Kelly, *Whole Earth Review,* "E–Money,"pp. 40–59, Summer 1993.
Green paper, "Intellectual Property and the National Information Infrastructure, a Preliminary Draft of the Report of the Working Group on Intellectual Property Rights," Jul. 1994.
Communications of the ACM, "Intelligent Agents," Jul. 1994, vol. 37, No. 7.
"Encapsulation: An Approach to Operating System Security," Bisbey, II et al., Oct. 1973, pp. 666–675.
"Encryption Methods in Data Networks," Blom et al., Ericsson Technics, No. 2, 1978, Stockholm, Sweden.
First CII Honeywell Bull International Symposium on Computer Security and Confidentiality, Jan. 26–28, 1981, Conference Text, pp. 1–21.
Codercard, Spec Sheet—Basic Coder Subsystem, No date given.
"Micro Card"—Micro Card Technologies, Inc., Dallas, Texas, No date given.
"A Method of Software Protection Based on the Use of Smart Cards and Cryptographic Techniques," Scnaumueller–Bichl et al., No date given.
I "The New Alexandria" No. 1, Alexandria Institute, pp. 1–12, Jul.–Aug. 1986.
Denning et al., "Data Security," 11 *Computing Surveys* No. 3, Sep. 1979.
Kent, "Protecting Externally Supplied Software In Small Computers" (MIT/LCS/TR–255 Sep. 1980).
Proceedings of the IEEE, vol. 67, No. 3, Mar. 1979, "Privacy and Authentication: An Introduction to Cryptography," Whitfield Diffie and Martin E. Hellman, pp. 397–427.
Digest of Papers, VLSI: New Architectural Horizons, Feb. 1980, "Preventing Software Piracy With Crypto–Microprocessors," Robert M. Best, pp. 466–469.
IEEE Transactions on Information Theory, vol. 22, No. 6, Nov. 1976, "New Directions in Cryptography," Whitfield Diffie and Martin E. Hellman, pp. 644–651.
Low, et al., "Anonymous Credit Cards," AT&T Bell Laboratories, Proceedings of the 2nd ACM Conference on Computer and Communication Security, Fairfax, Virginia, Nov. 2–4, 1994.
Tygar et al., "Cryptography: It's Not Just For Electronic Mail Anymore," CMU–CS–93–107, School of Computer Science Carnegie Mellon University, Pittsburgh, Pennsylvania, Mar. 1, 1993.
Smith, et al., "Signed Vector Timestamps: A Secure Protocol for Partial Order Time," CMU–93–116, School of Computer Science Carnegie Mellon University, Pittsburgh, Pennsylvania, Oct. 1991; version of Feb. 1993.
Kristol et al., "Anonymous Internet Mercantile Protocol," AT&T Bell Laboratories, Murray Hill, New Jersey, Draft: Mar. 17, 1994.
Low et al., "Document Marking and Identification using both Line and Word Shifting," AT&T Bell Laboratories, Murray Hill, New Jersey, Jul. 29, 1994.
Low et al., "Anonymous Credit Cards and its Collusion Analysis," AT&T Bell Laboratories, Murray Hill, New Jersey, Oct. 10, 1994.

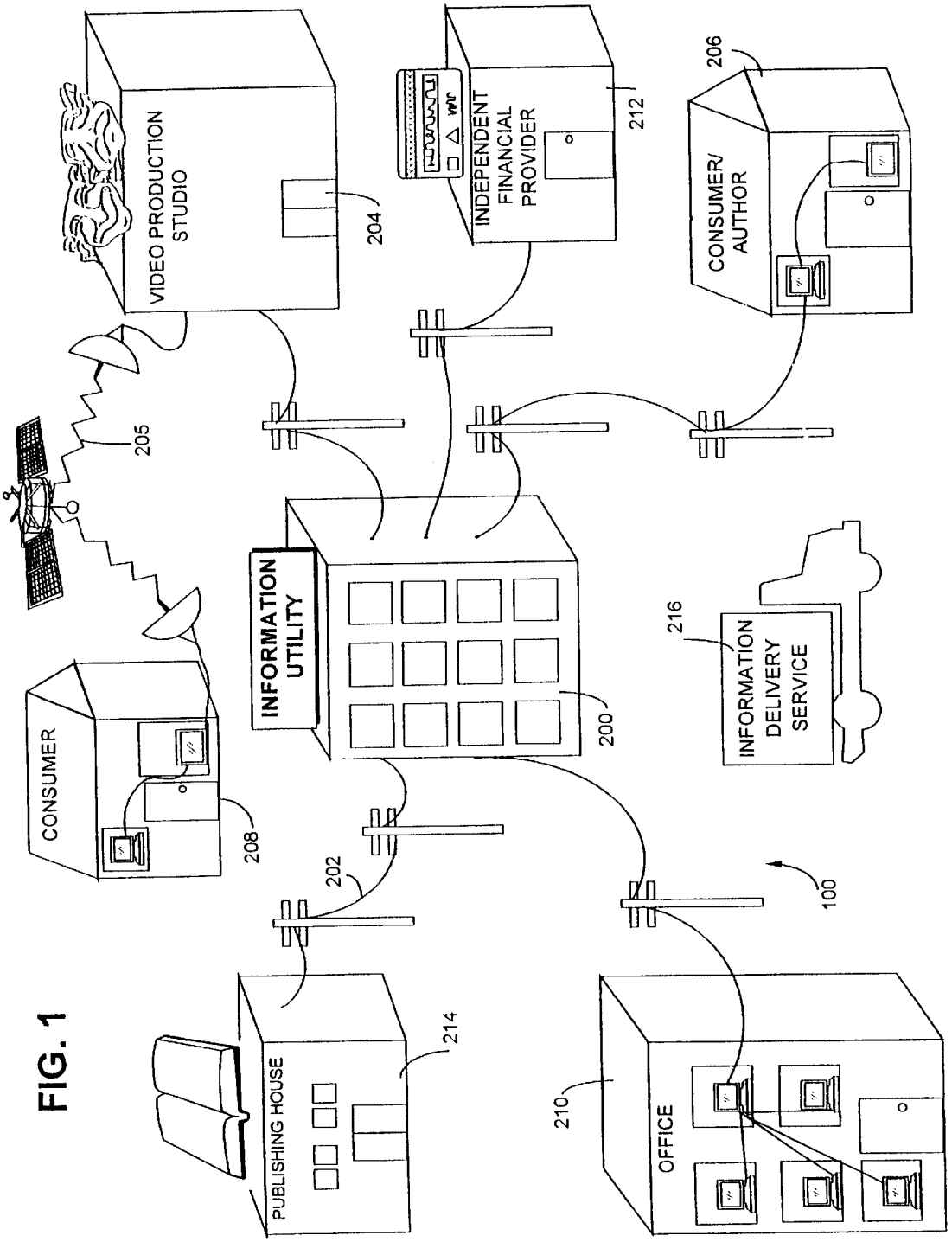

VIDEO PRODUCTION STUDIO
204
INDEPENDENT FINANCIAL PROVIDER
212
206
CONSUMER/ AUTHOR
205
INFORMATION UTILITY
200
216
INFORMATION DELIVERY SERVICE
CONSUMER
208
202
100
PUBLISHING HOUSE
214
210
OFFICE

FIG. 1

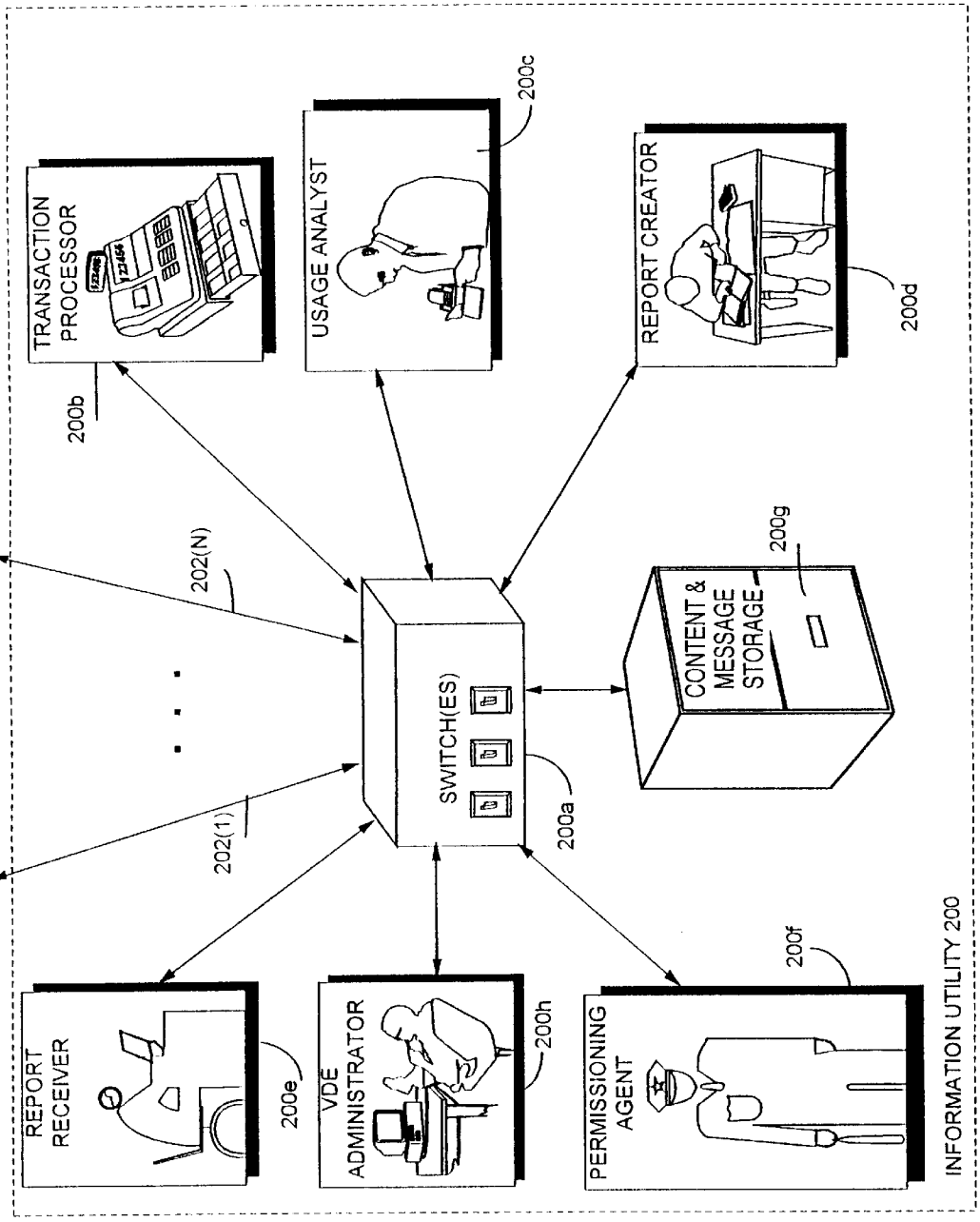

TO VDE PARTICIPANT 1
TO VDE PARTICIPANT N
202(1)
202(N)
SWITCH(ES)
200a
TRANSACTION PROCESSOR
200b
USAGE ANALYST
200c
REPORT CREATOR
200d
CONTENT & MESSAGE STORAGE
200g
REPORT RECEIVER
200e
VDE ADMINISTRATOR
200h
PERMISSIONING AGENT
200f
INFORMATION UTILITY 200

FIG. 1A

FIG. 2

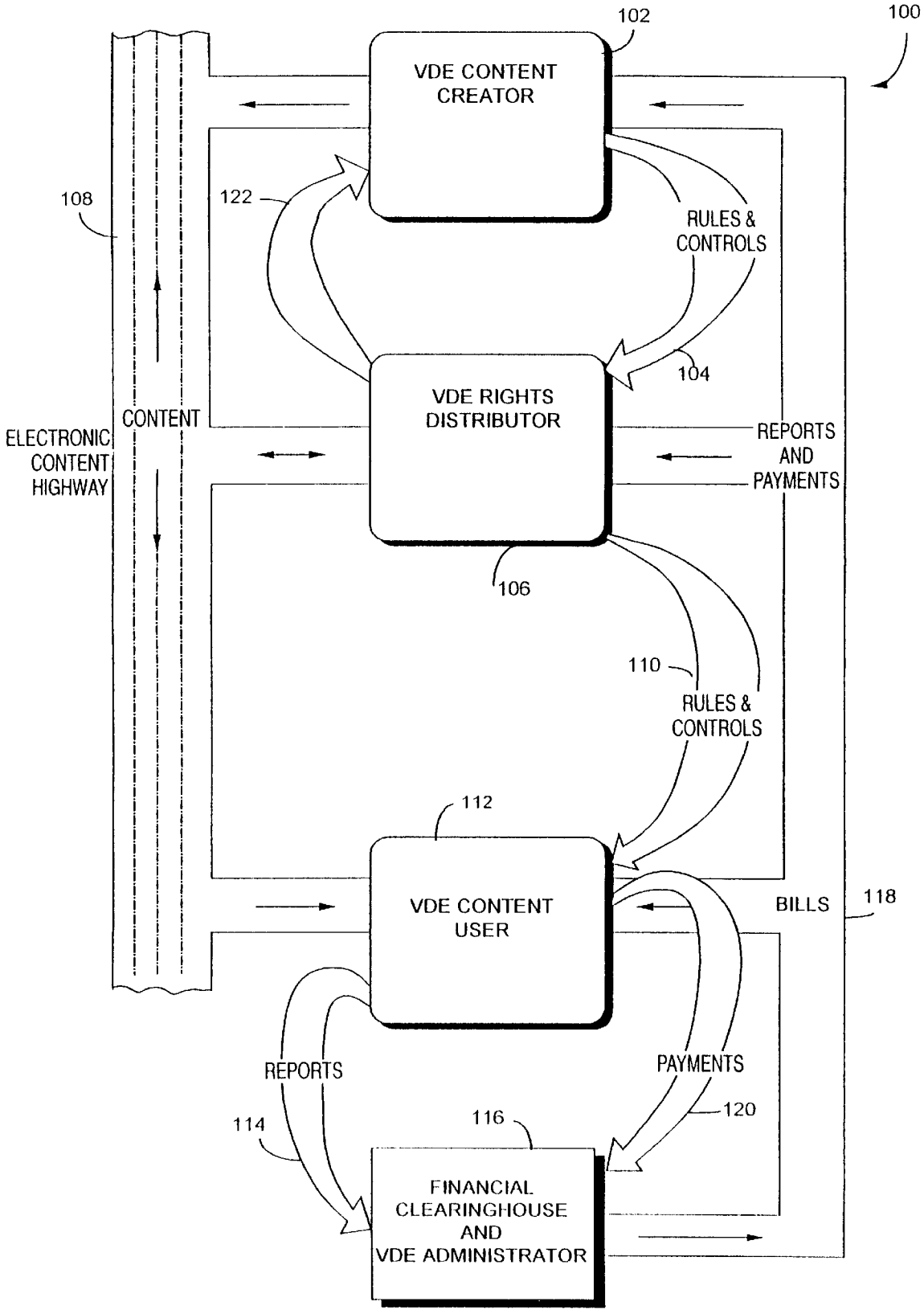

100
102
VDE CONTENT CREATOR
108
122
RULES & CONTROLS
104
VDE RIGHTS DISTRIBUTOR
CONTENT
ELECTRONIC CONTENT HIGHWAY
REPORTS AND PAYMENTS
106
110
RULES & CONTROLS
112
VDE CONTENT USER
BILLS
118
REPORTS
PAYMENTS
114
116
120
FINANCIAL CLEARINGHOUSE AND VDE ADMINISTRATOR

FIG. 2A

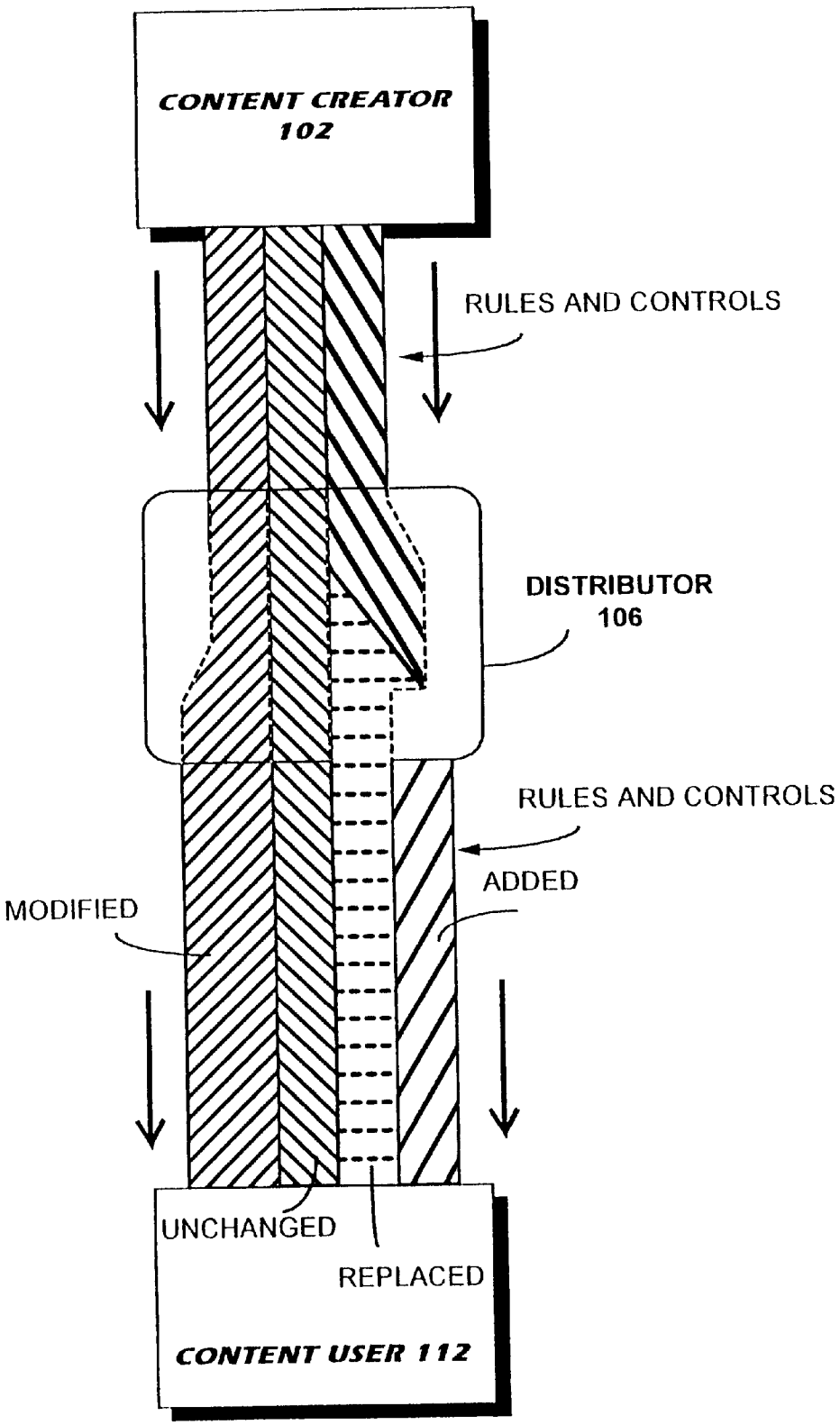
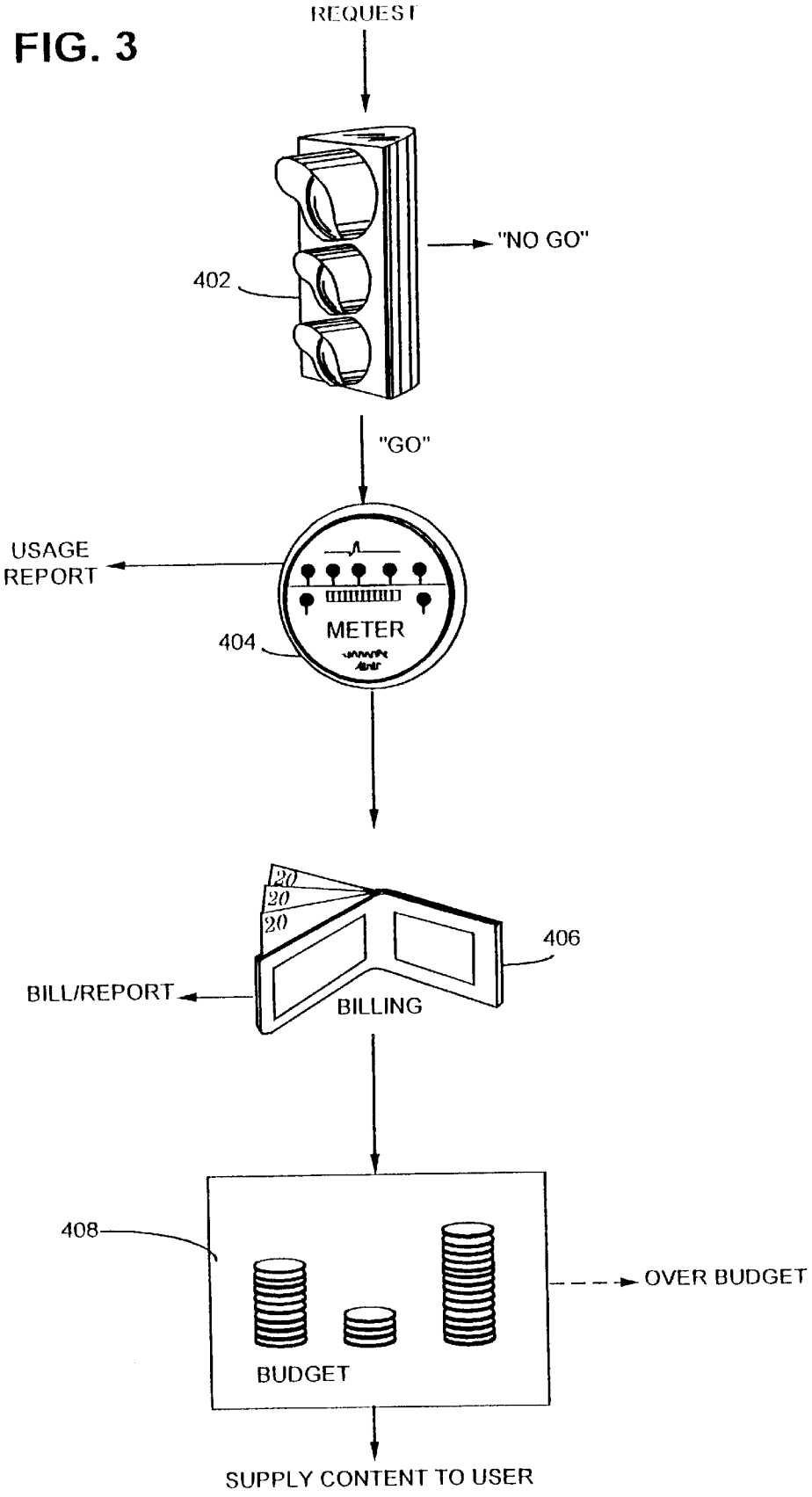

CONTENT CREATOR
102
RULES AND CONTROLS
DISTRIBUTOR
106
RULES AND CONTROLS
ADDED
MODIFIED
UNCHANGED
REPLACED
CONTENT USER 112

FIG. 3


REQUEST
"NO GO"
402
"GO"
USAGE REPORT
METER
404
20
20
20
406
BILL/REPORT
BILLING
408
OVER BUDGET
BUDGET
SUPPLY CONTENT TO USER

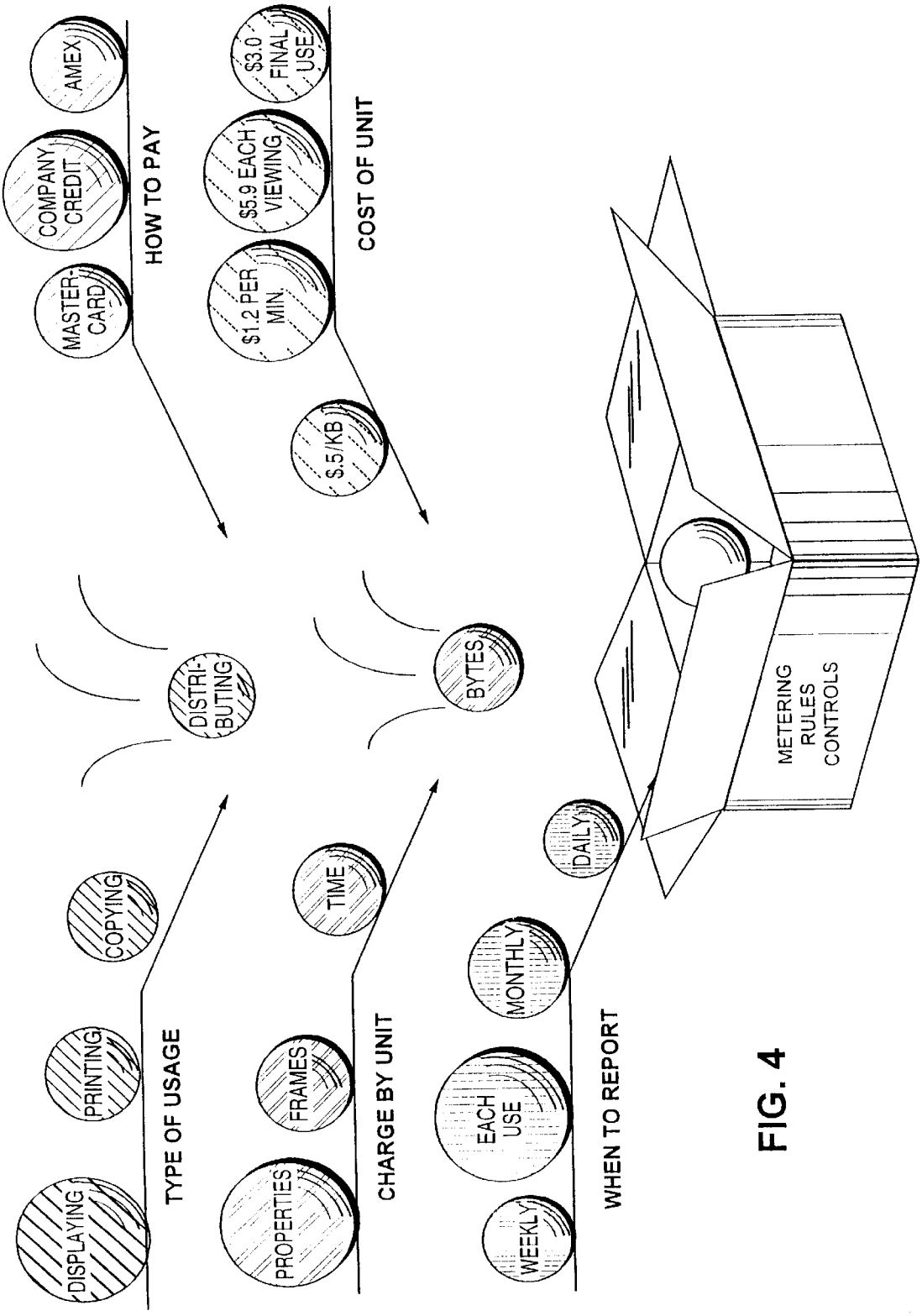
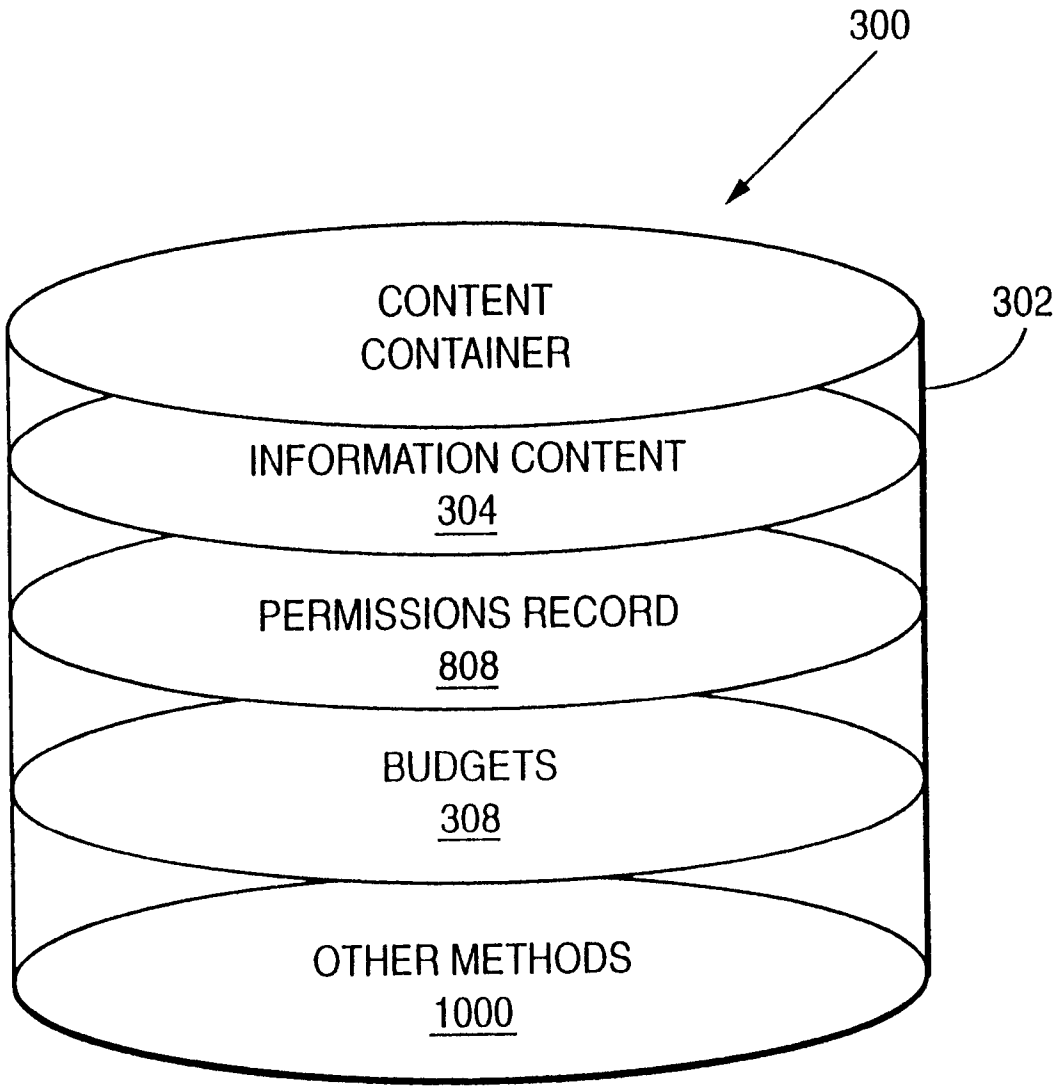

TYPE OF USAGE
DISPLAYING
PRINTING
COPYING
DISTRI-BUTING
CHARGE BY UNIT
PROPERTIES
FRAMES
TIME
BYTES
WHEN TO REPORT
WEEKLY
EACH USE
MONTHLY
DAILY
HOW TO PAY
MASTER-CARD
COMPANY CREDIT
AMEX
COST OF UNIT
$.5/KB
$1.2 PER MIN.
$5.9 EACH VIEWING
$3.0 FINAL USE
METERING RULES CONTROLS

FIG. 4

FIG. 5A


300
CONTENT CONTAINER
302
INFORMATION CONTENT
304
PERMISSIONS RECORD
808
BUDGETS
308
OTHER METHODS
1000

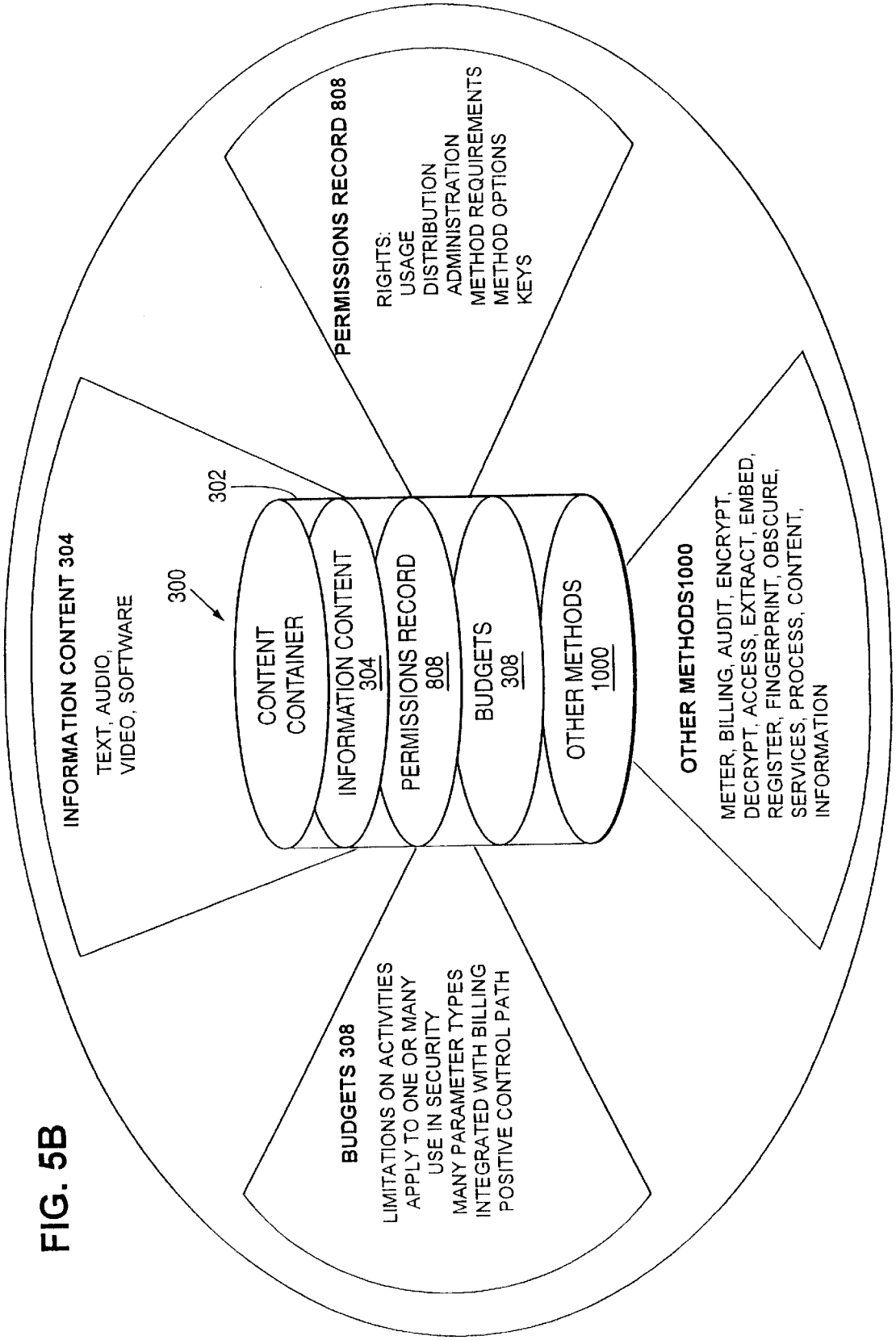

INFORMATION CONTENT 304
TEXT, AUDIO,
VIDEO, SOFTWARE
300
302
CONTENT CONTAINER
INFORMATION CONTENT 304
PERMISSIONS RECORD 808
BUDGETS 308
OTHER METHODS 1000
PERMISSIONS RECORD 808
RIGHTS:
USAGE
DISTRIBUTION
ADMINISTRATION
METHOD REQUIREMENTS
METHOD OPTIONS
KEYS
OTHER METHODS1000
METER, BILLING, AUDIT, ENCRYPT,
DECRYPT, ACCESS, EXTRACT, EMBED,
REGISTER, FINGERPRINT, OBSCURE,
SERVICES, PROCESS, CONTENT,
INFORMATION
BUDGETS 308
LIMITATIONS ON ACTIVITIES
APPLY TO ONE OR MANY
USE IN SECURITY
MANY PARAMETER TYPES
INTEGRATED WITH BILLING
POSITIVE CONTROL PATH

FIG. 5B

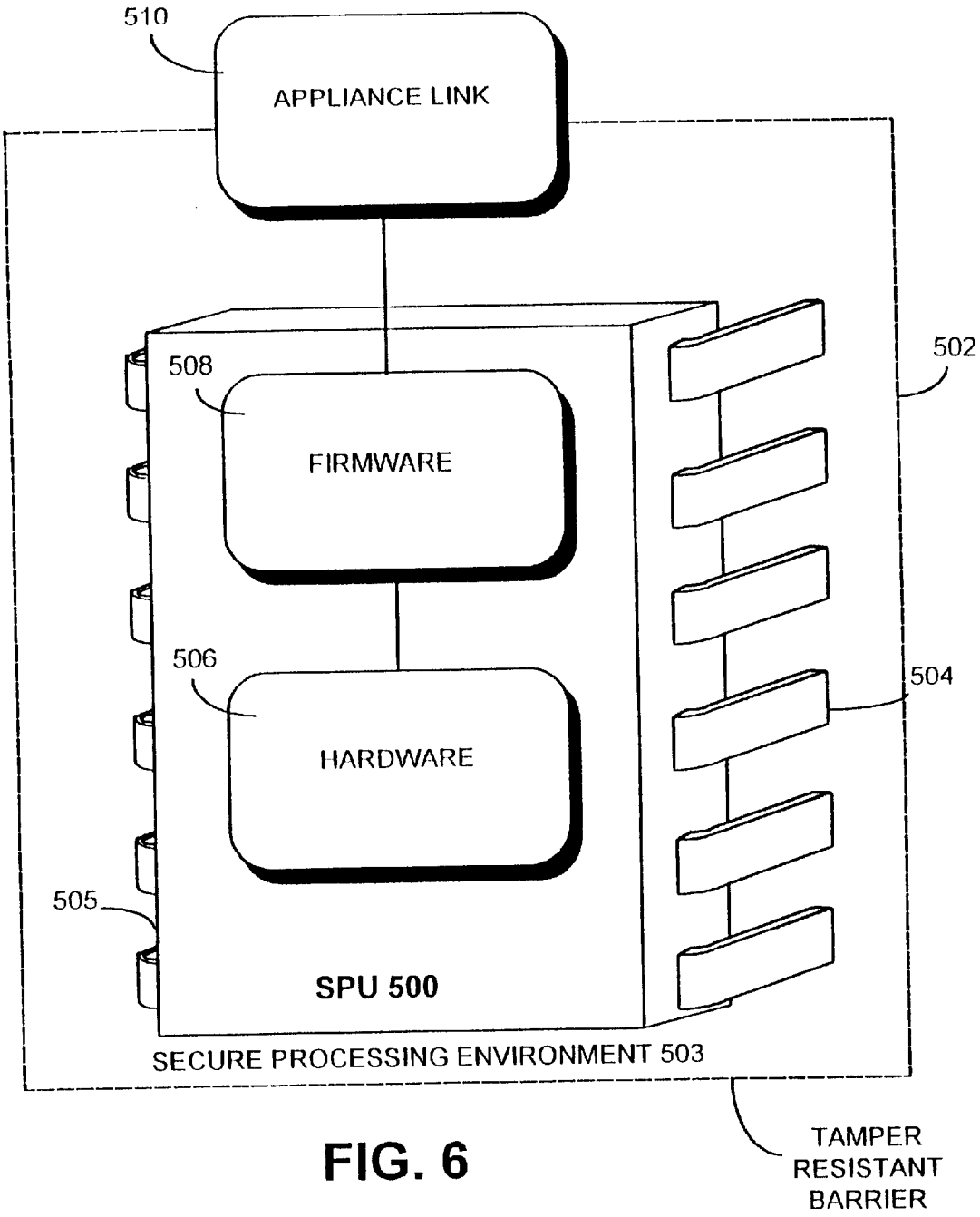

510
APPLIANCE LINK
502
508
FIRMWARE
506
HARDWARE
504
505
SPU 500
SECURE PROCESSING ENVIRONMENT 503
TAMPER RESISTANT BARRIER

FIG. 6

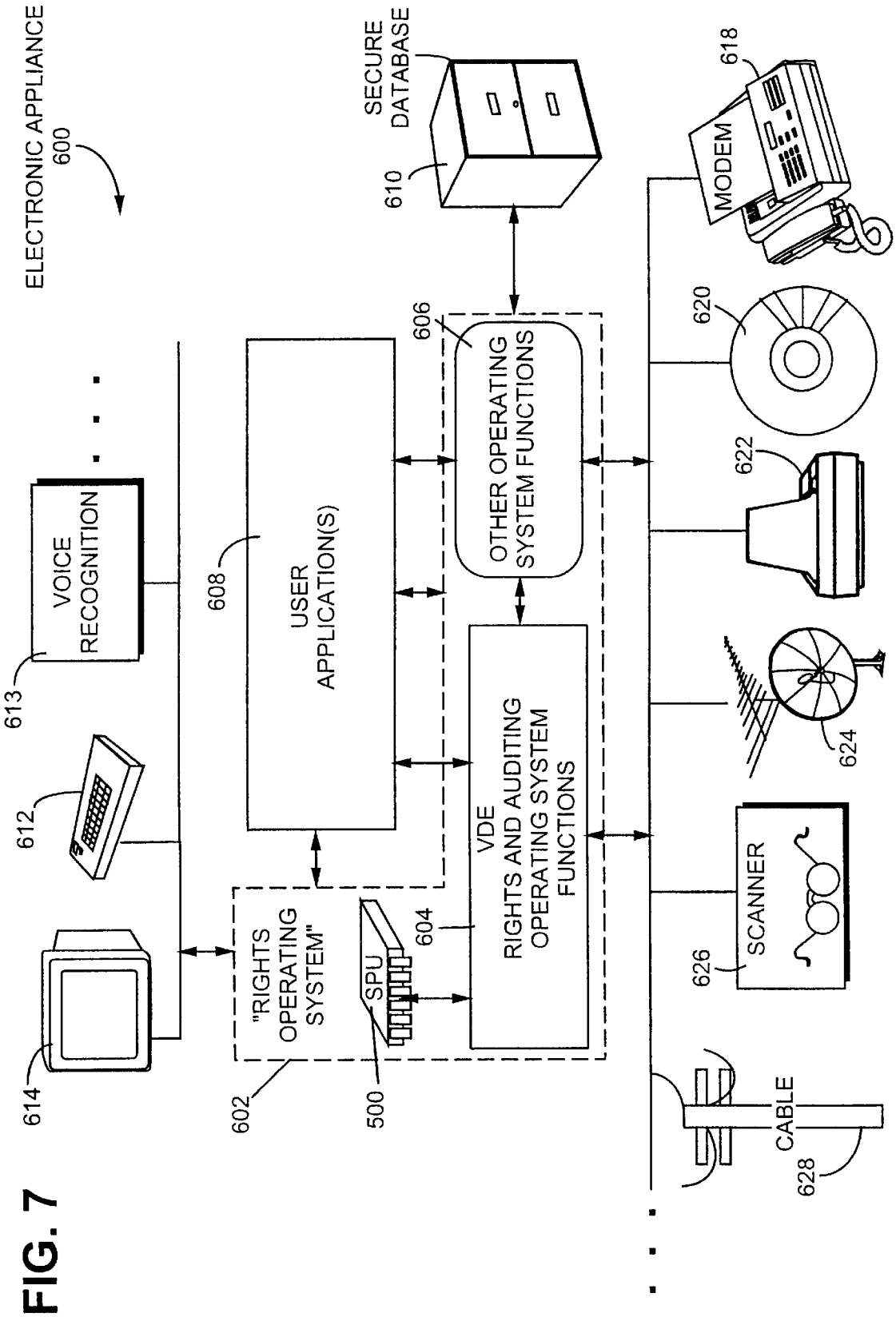
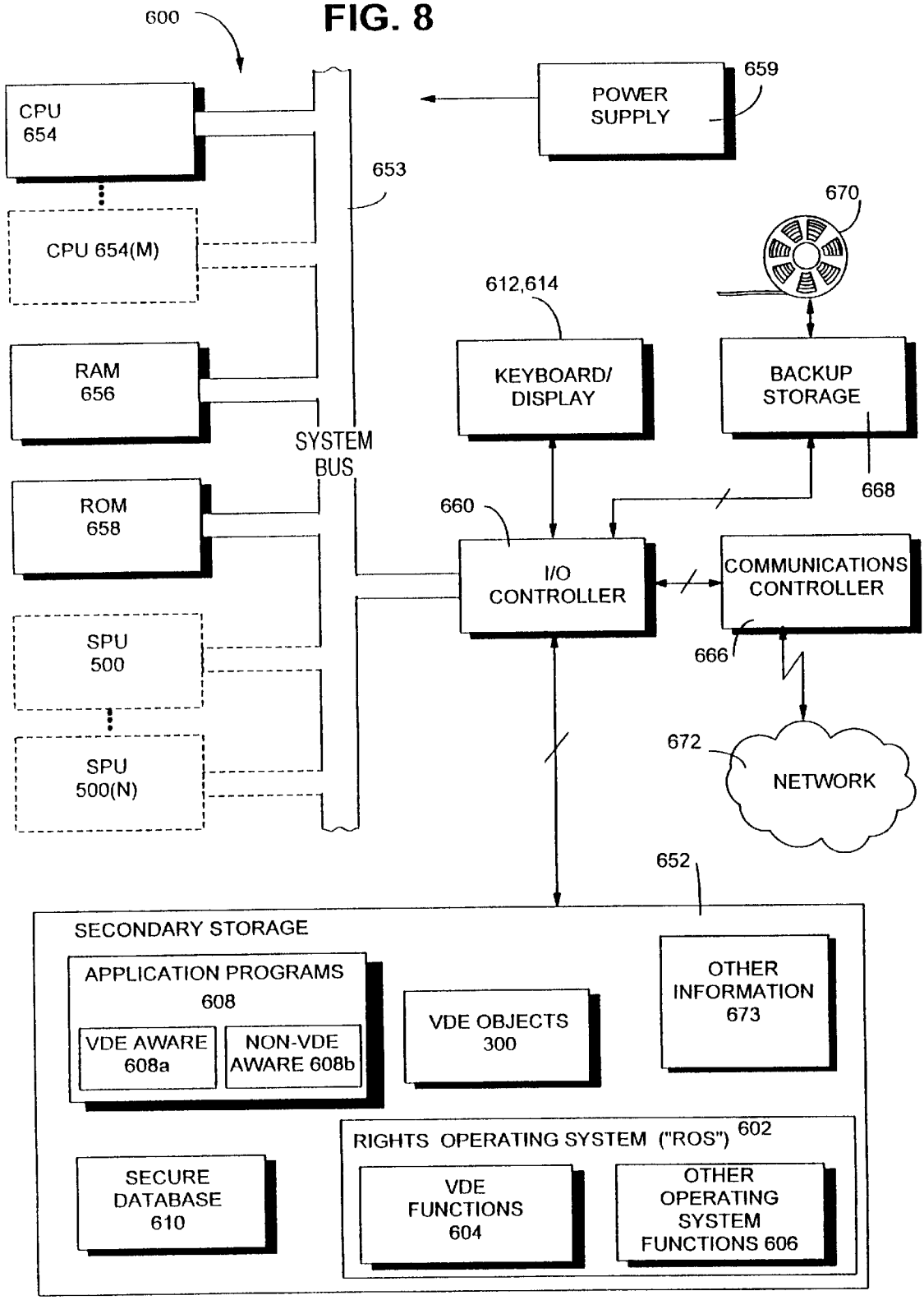

ELECTRONIC APPLIANCE
600
SECURE DATABASE
610
MODEM
618
606
OTHER OPERATING SYSTEM FUNCTIONS
620
608
USER APPLICATION(S)
622
VOICE RECOGNITION
613
612
624
VDE RIGHTS AND AUDITING OPERATING SYSTEM FUNCTIONS
604
"RIGHTS OPERATING SYSTEM"
SPU
SCANNER
626
614
602
500
CABLE
628

FIG. 7

FIG. 8


600
CPU 654
CPU 654(M)
RAM 656
ROM 658
SPU 500
SPU 500(N)
SYSTEM BUS
653
POWER SUPPLY
659
670
612,614
KEYBOARD/ DISPLAY
BACKUP STORAGE
668
660
I/O CONTROLLER
COMMUNICATIONS CONTROLLER
666
672
NETWORK
652
SECONDARY STORAGE
APPLICATION PROGRAMS 608
VDE AWARE 608a
NON-VDE AWARE 608b
VDE OBJECTS 300
OTHER INFORMATION 673
RIGHTS OPERATING SYSTEM ("ROS") 602
SECURE DATABASE 610
VDE FUNCTIONS 604
OTHER OPERATING SYSTEM FUNCTIONS 606

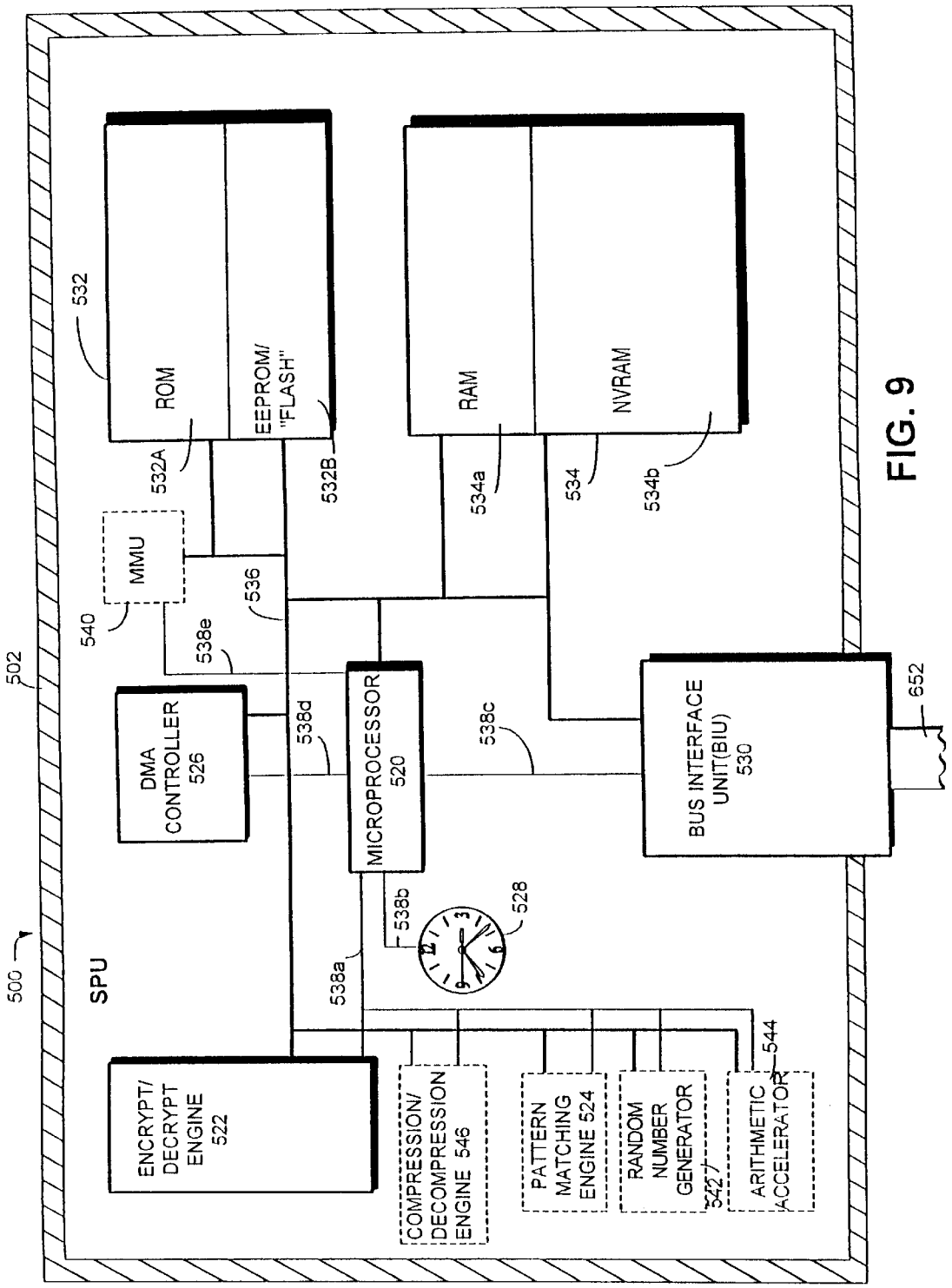

500
502
SPU
532
ROM
EEPROM/ "FLASH"
532A
532B
RAM
NVRAM
534a
534
534b
MMU
540
536
538e
538d
538c
538b
538a
528
DMA CONTROLLER 526
MICROPROCESSOR 520
BUS INTERFACE UNIT(BIU) 530
652
ENCRYPT/ DECRYPT ENGINE 522
COMPRESSION/ DECOMPRESSION ENGINE 546
PATTERN MATCHING ENGINE 524
RANDOM NUMBER GENERATOR
542
ARITHMETIC ACCELERATOR
544

FIG. 9

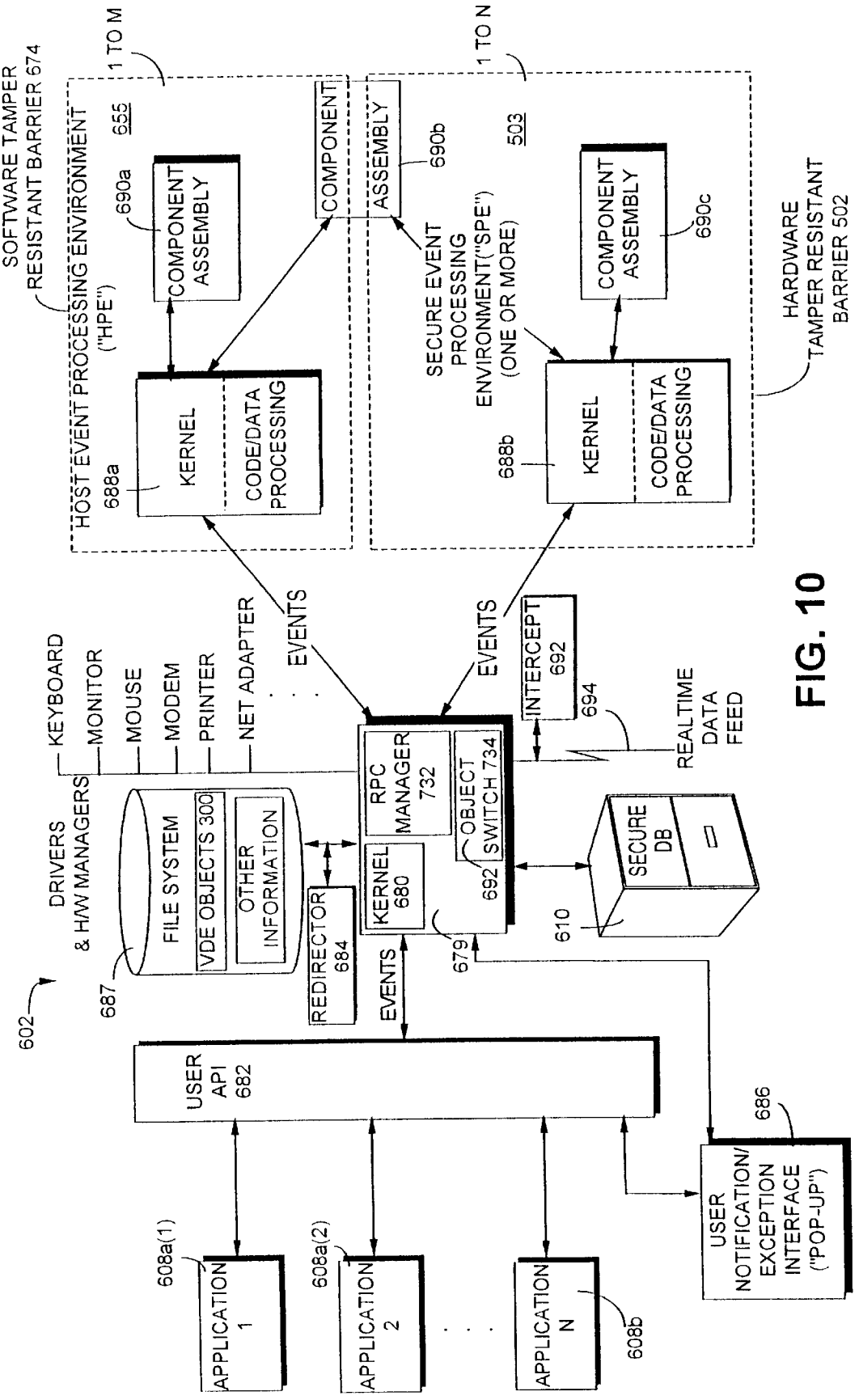

SOFTWARE TAMPER RESISTANT BARRIER 674
HOST EVENT PROCESSING ENVIRONMENT ("HPE")
655
1 TO M
690a
COMPONENT ASSEMBLY
COMPONENT ASSEMBLY
690b
1 TO N
503
SECURE EVENT PROCESSING ENVIRONMENT("SPE") (ONE OR MORE)
COMPONENT ASSEMBLY
690c
HARDWARE TAMPER RESISTANT BARRIER 502
688a
KERNEL
CODE/DATA PROCESSING
688b
KERNEL
CODE/DATA PROCESSING
EVENTS
EVENTS
INTERCEPT 692
694
REALTIME DATA FEED
KEYBOARD
MONITOR
MOUSE
MODEM
PRINTER
NET ADAPTER
DRIVERS & H/W MANAGERS
FILE SYSTEM
VDE OBJECTS 300
OTHER INFORMATION
687
602
REDIRECTOR 684
RPC MANAGER 732
KERNEL 680
OBJECT SWITCH 734
692
679
SECURE DB
610
EVENTS
USER API 682
686
USER NOTIFICATION/ EXCEPTION INTERFACE ("POP-UP")
608a(1)
APPLICATION 1
608a(2)
APPLICATION 2
APPLICATION N
608b

FIG. 10

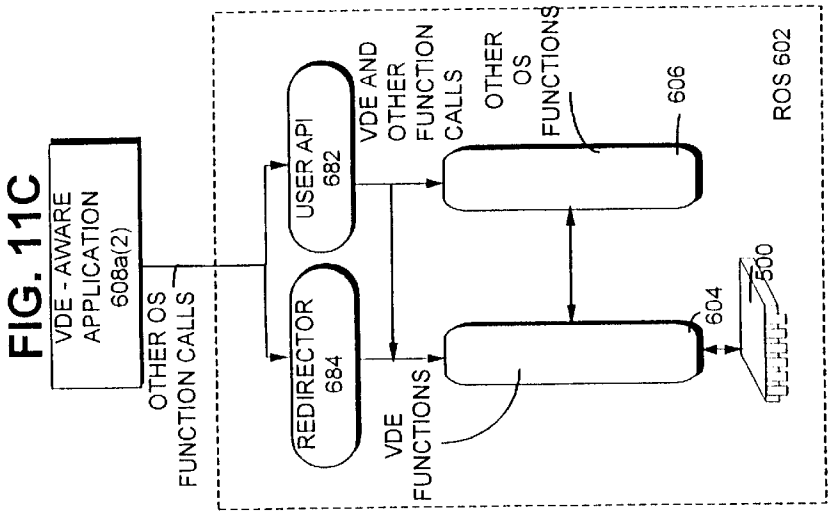


FIG. 11C

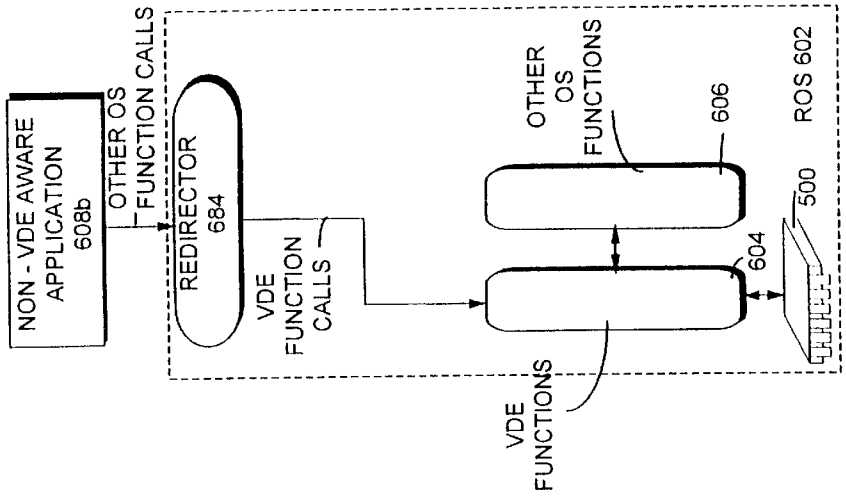


FIG. 11B

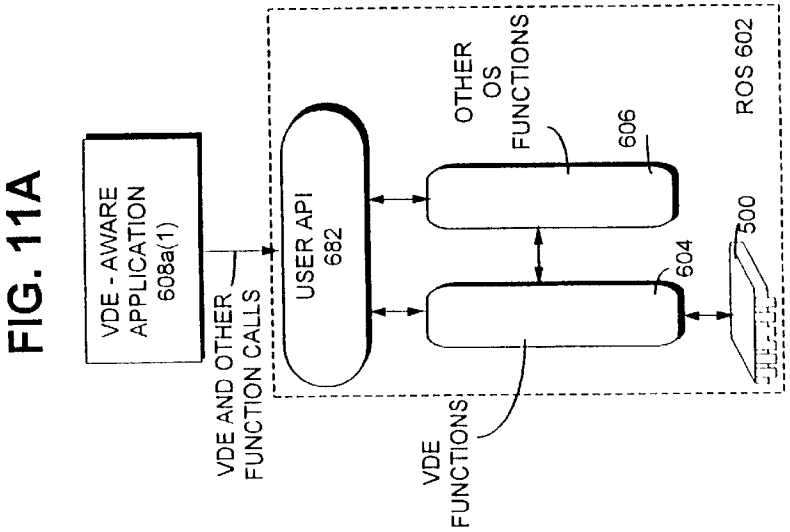


FIG. 11A

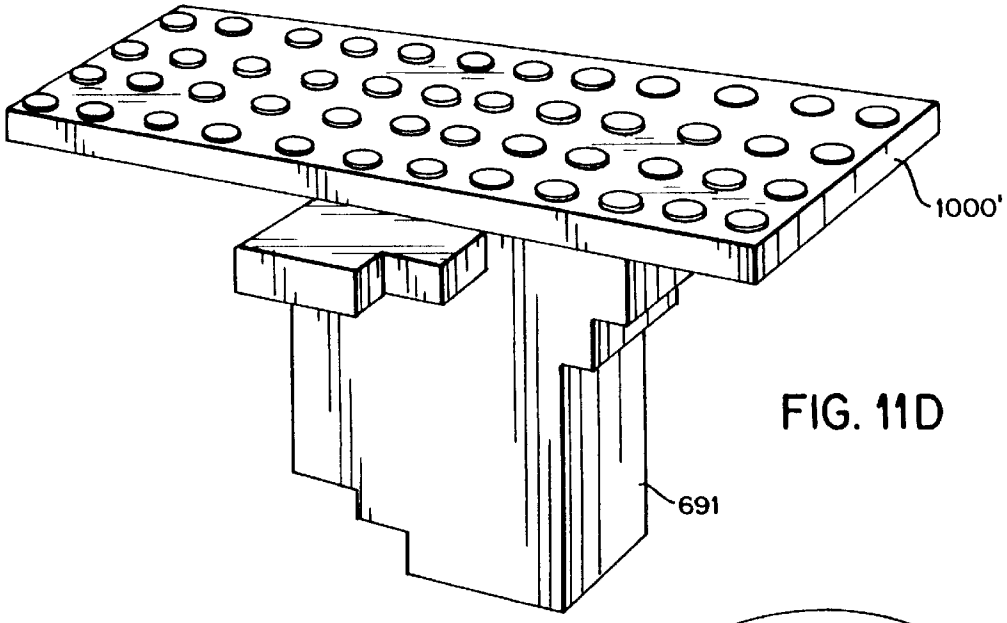


FIG. 11D

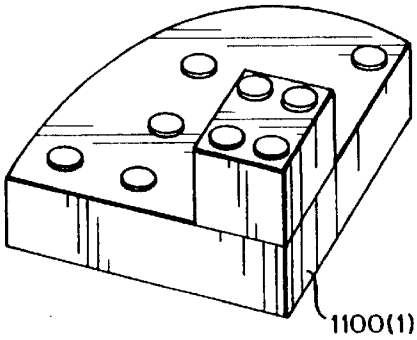


FIG. 11E

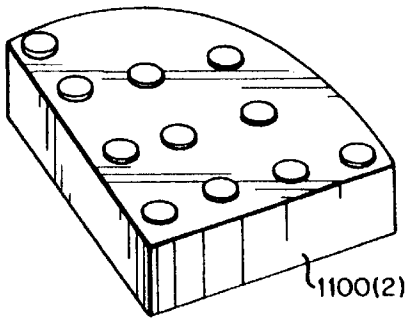


FIG. 11F

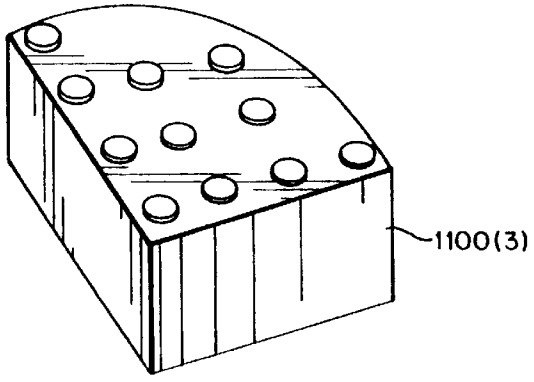


FIG. 11G

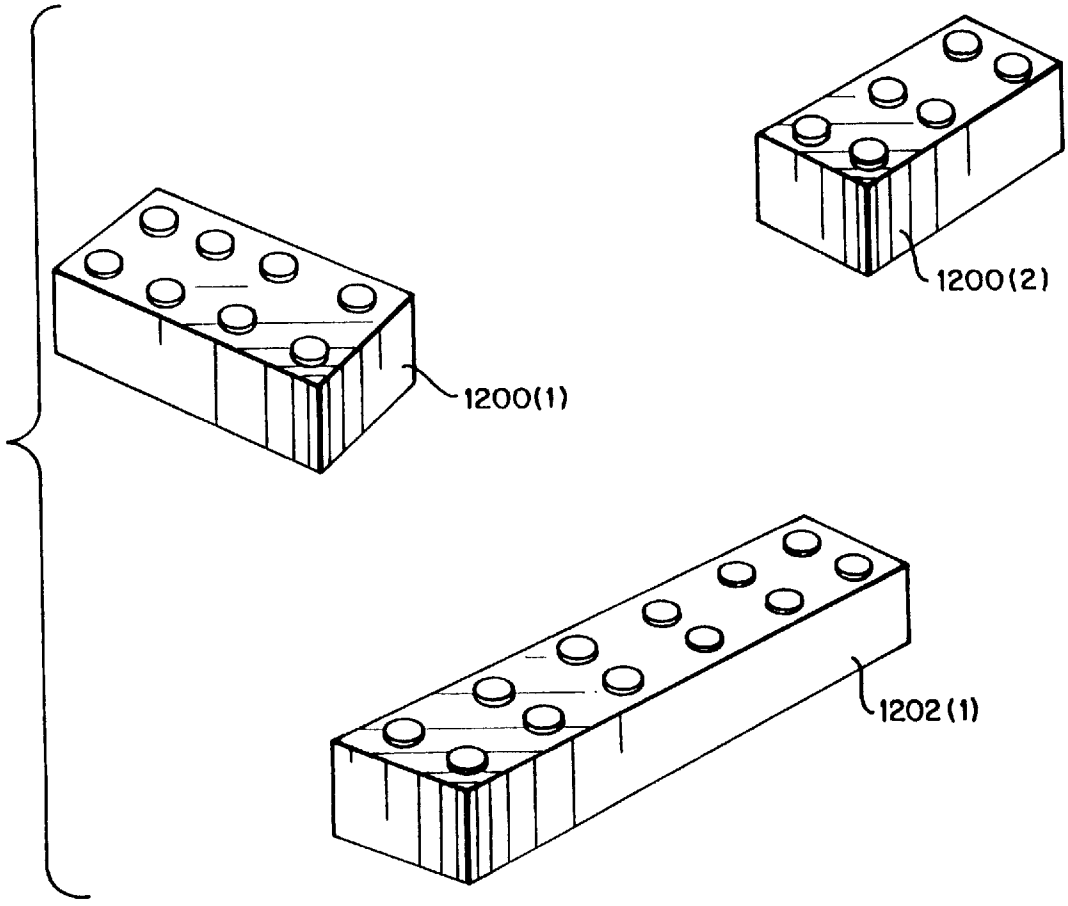
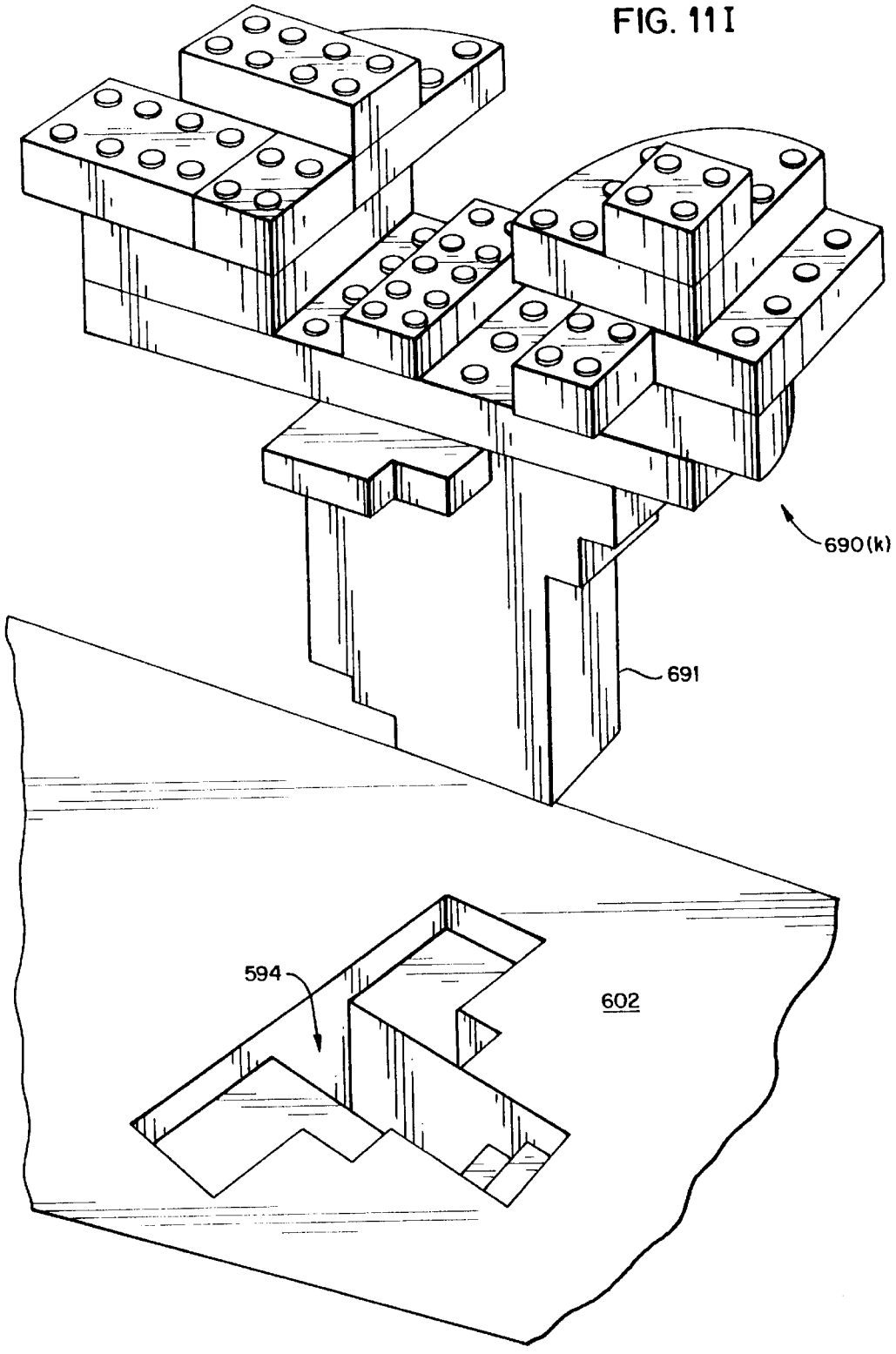


FIG. 11H

FIG. 11I



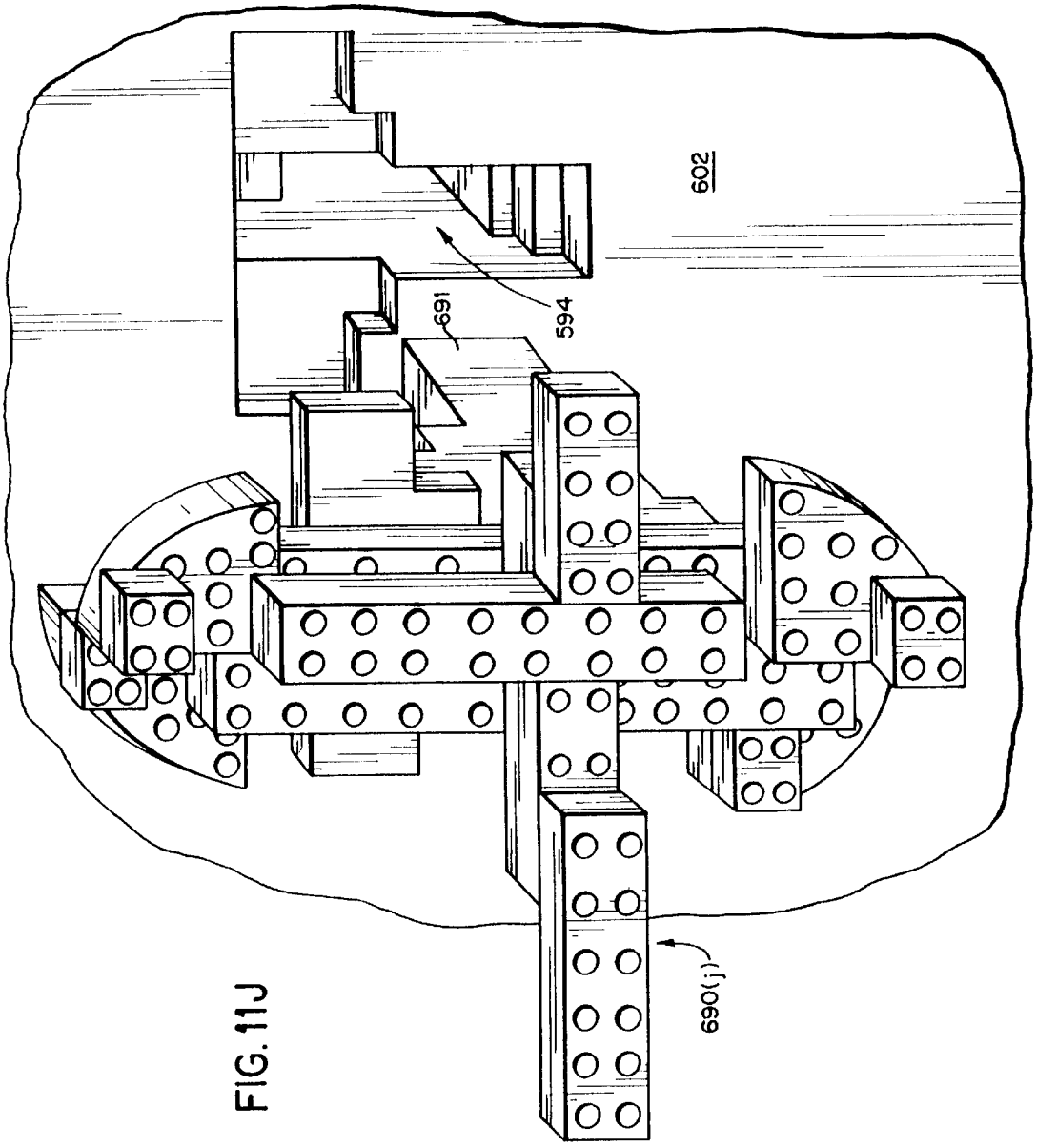


FIG. 11J

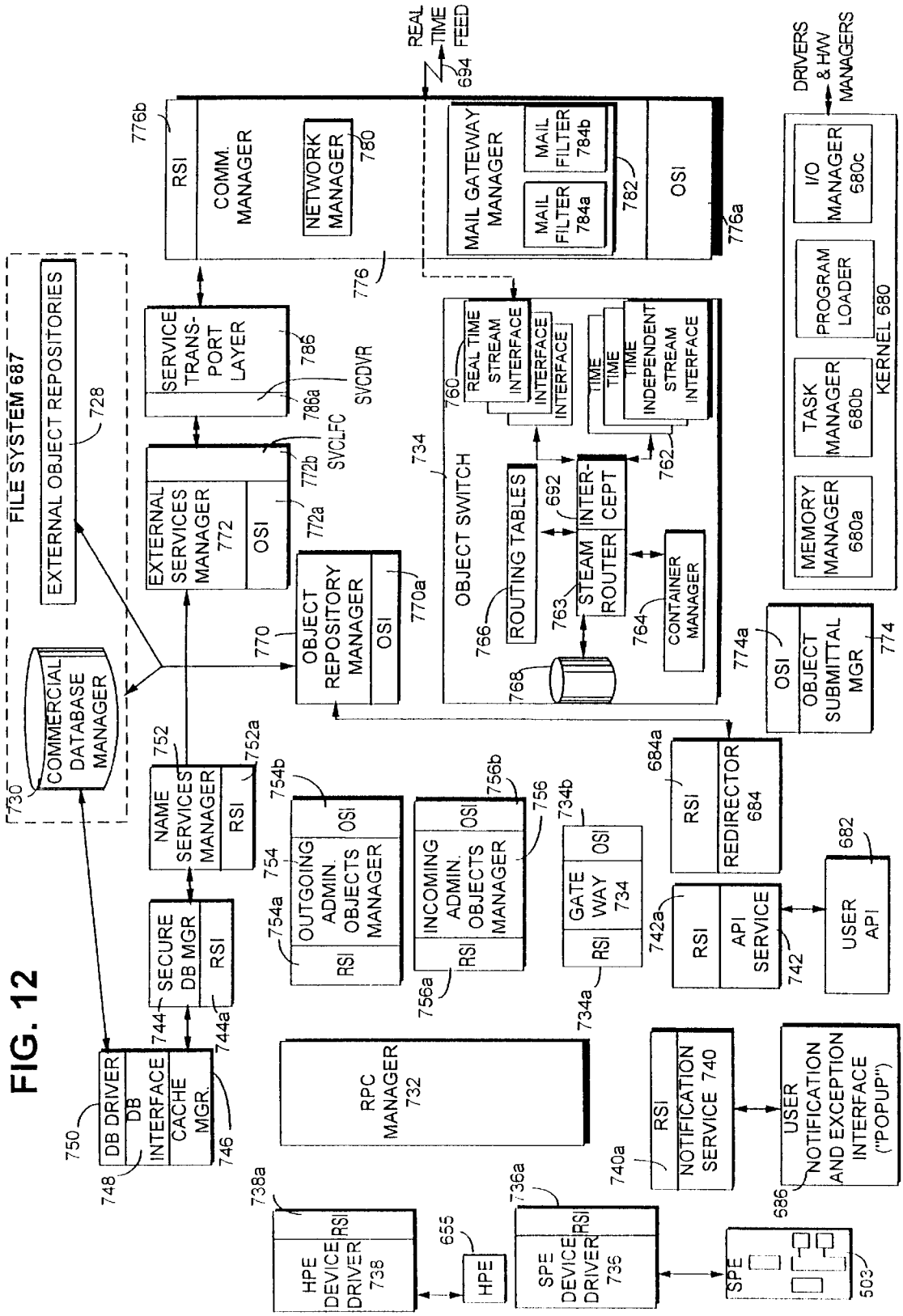

REAL TIME FEED
694
DRIVERS & H/W MANAGERS
776b
RSI
COMM. MANAGER
NETWORK MANAGER
780
MAIL GATEWAY MANAGER
MAIL FILTER 784a
MAIL FILTER 784b
782
OSI
776a
776
I/O MANAGER 680c
PROGRAM LOADER
TASK MANAGER 680b
MEMORY MANAGER 680a
KERNEL 680
FILE SYSTEM 687
EXTERNAL OBJECT REPOSITORIES
728
SERVICE TRANS-PORT LAYER
786
786a
SVCDVR
SVCLFC
772b
EXTERNAL SERVICES MANAGER 772
OSI
772a
734
OBJECT SWITCH
760
REAL TIME STREAM INTERFACE
INTERFACE
INTERFACE
TIME
TIME
TIME INDEPENDENT STREAM INTERFACE
762
692
ROUTING TABLES
INTER-CEPT
STEAM ROUTER
763
766
768
764
CONTAINER MANAGER
770
OBJECT REPOSITORY MANAGER
OSI
770a
774a
OSI
OBJECT SUBMITTAL MGR
774
730
COMMERCIAL DATABASE MANAGER
752
NAME SERVICES MANAGER
RSI
752a
754b
754
OUTGOING ADMIN. OBJECTS MANAGER
OSI
RSI
754a
756b
INCOMING ADMIN. OBJECTS MANAGER
OSI
RSI
756
756a
734b
GATE WAY 734
OSI
RSI
734a
684a
RSI
REDIRECTOR 684
682
USER API
742a
RSI
API SERVICE
742
SECURE DB MGR
RSI
744
744a
DB DRIVER
DB INTERFACE
CACHE MGR.
746
750
748
RPC MANAGER 732
RSI
NOTIFICATION SERVICE 740
740a
USER NOTIFICATION AND EXCEPTION INTERFACE ("POPUP")
686
738a
HPE DEVICE DRIVER 738
RSI
655
HPE
736a
SPE DEVICE DRIVER 736
RSI
SPE
503

FIG. 12

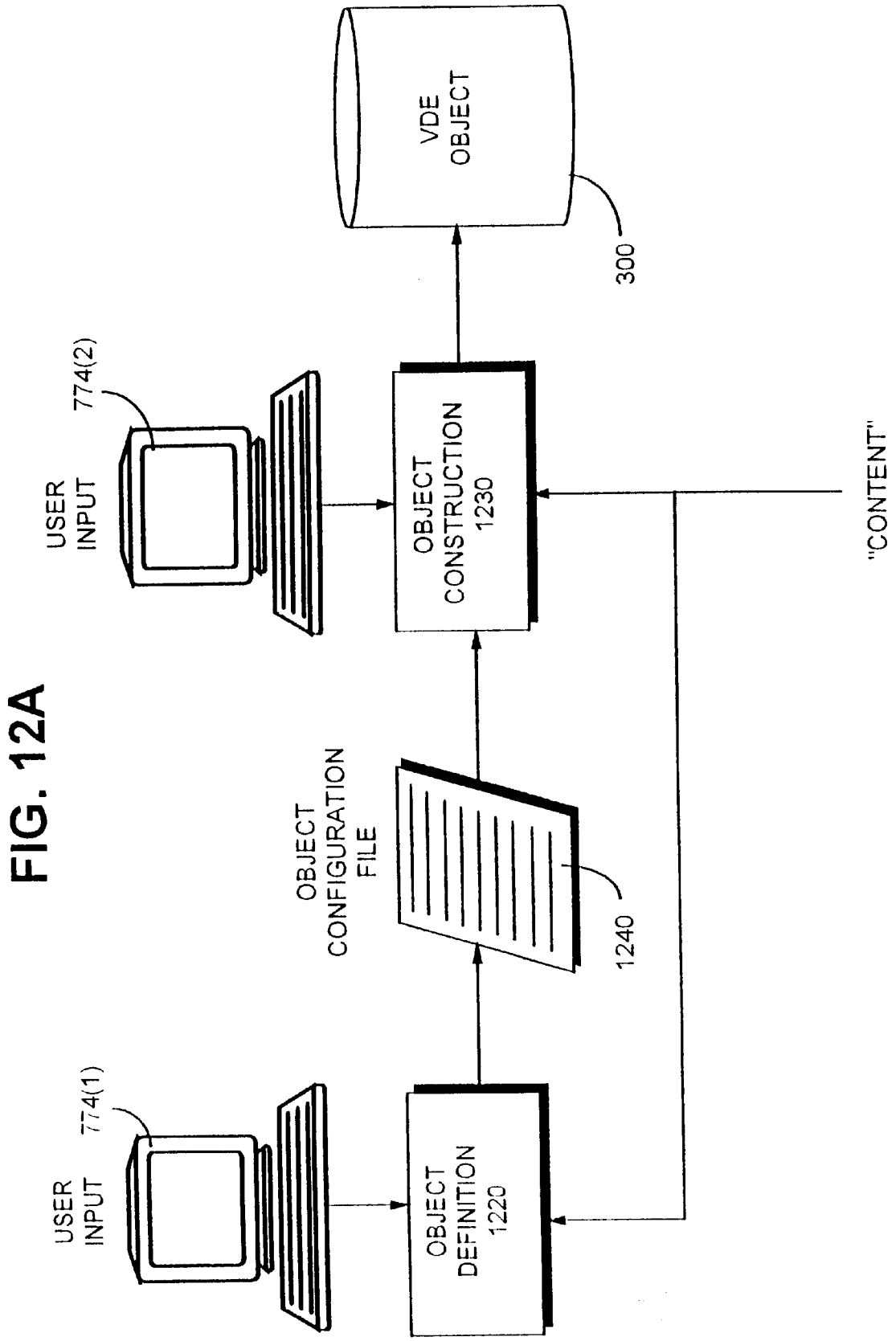

VDE OBJECT
300
USER INPUT
774(2)
OBJECT CONSTRUCTION 1230
"CONTENT"
OBJECT CONFIGURATION FILE
1240
USER INPUT
774(1)
OBJECT DEFINITION 1220

FIG. 12A

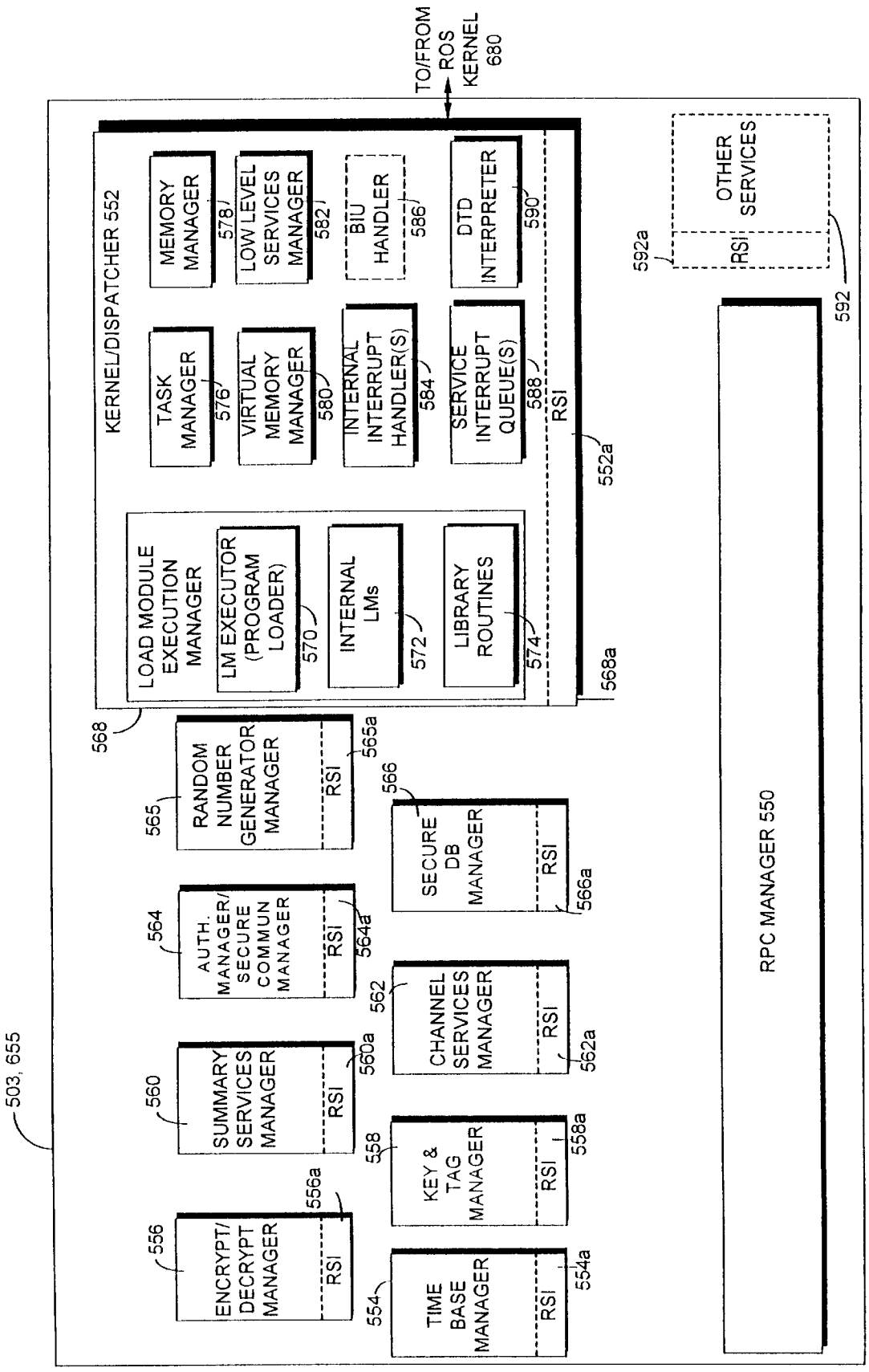

TO/FROM ROS KERNEL 680
KERNEL/DISPATCHER 552
MEMORY MANAGER 578
LOW LEVEL SERVICES MANAGER 582
BIU HANDLER 586
DTD INTERPRETER 590
TASK MANAGER 576
VIRTUAL MEMORY MANAGER 580
INTERNAL INTERRUPT HANDLER(S) 584
SERVICE INTERRUPT QUEUE(S) 588
RSI
552a
LOAD MODULE EXECUTION MANAGER
LM EXECUTOR (PROGRAM LOADER) 570
INTERNAL LMs 572
LIBRARY ROUTINES 574
568a
568
OTHER SERVICES
RSI
592a
592
RPC MANAGER 550
RANDOM NUMBER GENERATOR MANAGER 565
RSI
565a
SECURE DB MANAGER 566
RSI
566a
AUTH. MANAGER/ SECURE COMMUN MANAGER 564
RSI
564a
CHANNEL SERVICES MANAGER 562
RSI
562a
SUMMARY SERVICES MANAGER 560
RSI
560a
KEY & TAG MANAGER 558
RSI
558a
ENCRYPT/ DECRYPT MANAGER 556
RSI
556a
TIME BASE MANAGER 554
RSI
554a
503, 655

FIG. 13

FIG. 14A

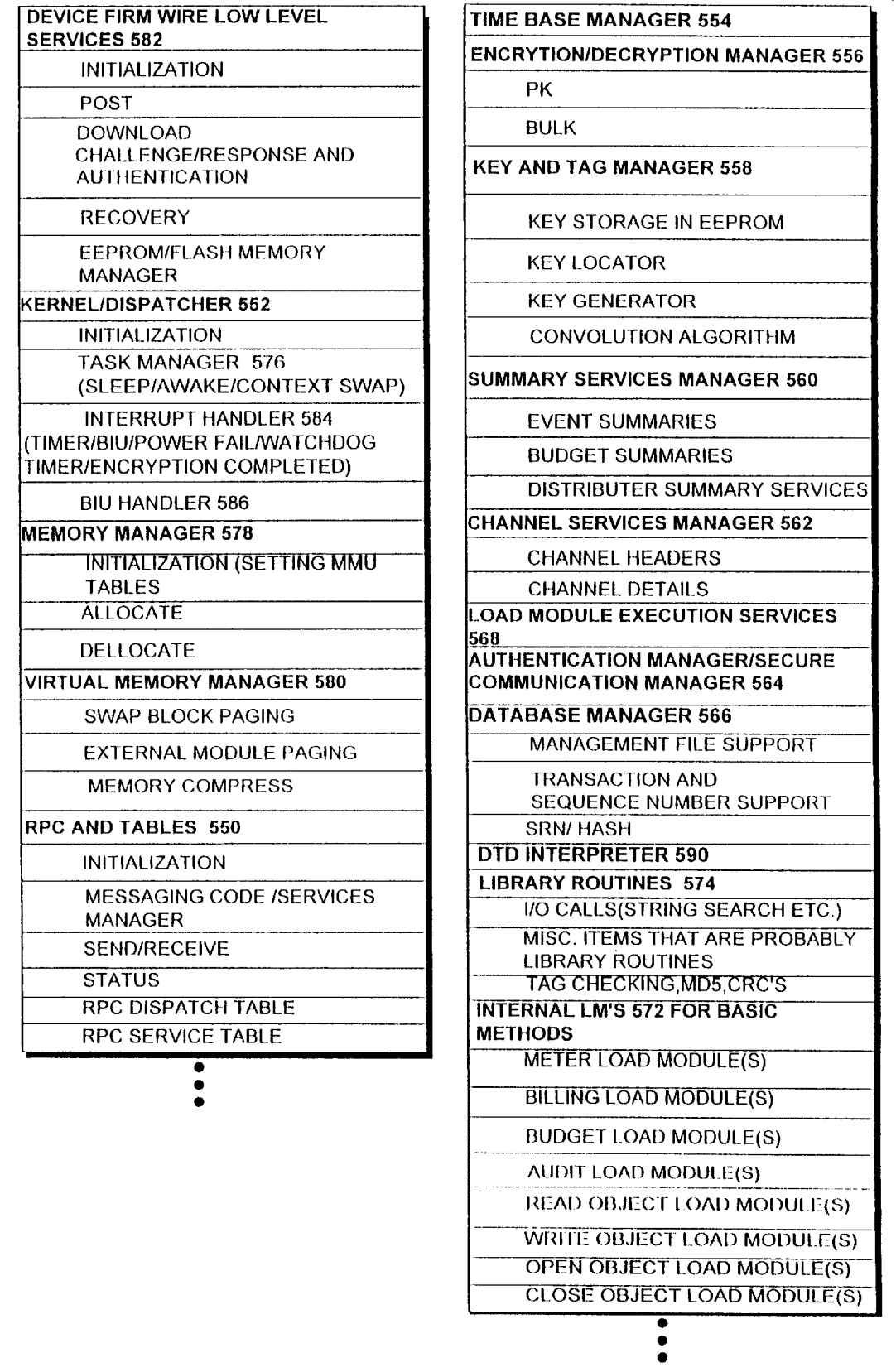

DEVICE FIRM WIRE LOW LEVEL SERVICES 582
INITIALIZATION
POST
DOWNLOAD CHALLENGE/RESPONSE AND AUTHENTICATION
RECOVERY
EEPROM/FLASH MEMORY MANAGER
KERNEL/DISPATCHER 552
INITIALIZATION
TASK MANAGER 576 (SLEEP/AWAKE/CONTEXT SWAP)
INTERRUPT HANDLER 584 (TIMER/BIU/POWER FAIL/WATCHDOG TIMER/ENCRYPTION COMPLETED)
BIU HANDLER 586
MEMORY MANAGER 578
INITIALIZATION (SETTING MMU TABLES
ALLOCATE
DELLOCATE
VIRTUAL MEMORY MANAGER 580
SWAP BLOCK PAGING
EXTERNAL MODULE PAGING
MEMORY COMPRESS
RPC AND TABLES 550
INITIALIZATION
MESSAGING CODE /SERVICES MANAGER
SEND/RECEIVE
STATUS
RPC DISPATCH TABLE
RPC SERVICE TABLE
TIME BASE MANAGER 554
ENCRYTION/DECRYPTION MANAGER 556
PK
BULK
KEY AND TAG MANAGER 558
KEY STORAGE IN EEPROM
KEY LOCATOR
KEY GENERATOR
CONVOLUTION ALGORITHM
SUMMARY SERVICES MANAGER 560
EVENT SUMMARIES
BUDGET SUMMARIES
DISTRIBUTER SUMMARY SERVICES
CHANNEL SERVICES MANAGER 562
CHANNEL HEADERS
CHANNEL DETAILS
LOAD MODULE EXECUTION SERVICES 568
AUTHENTICATION MANAGER/SECURE COMMUNICATION MANAGER 564
DATABASE MANAGER 566
MANAGEMENT FILE SUPPORT
TRANSACTION AND SEQUENCE NUMBER SUPPORT
SRN/ HASH
DTD INTERPRETER 590
LIBRARY ROUTINES 574
I/O CALLS(STRING SEARCH ETC.)
MISC. ITEMS THAT ARE PROBABLY LIBRARY ROUTINES
TAG CHECKING,MD5,CRC'S
INTERNAL LM'S 572 FOR BASIC METHODS
METER LOAD MODULE(S)
BILLING LOAD MODULE(S)
BUDGET LOAD MODULE(S)
AUDIT LOAD MODULE(S)
READ OBJECT LOAD MODULE(S)
WRITE OBJECT LOAD MODULE(S)
OPEN OBJECT LOAD MODULE(S)
CLOSE OBJECT LOAD MODULE(S)

FIG. 14B

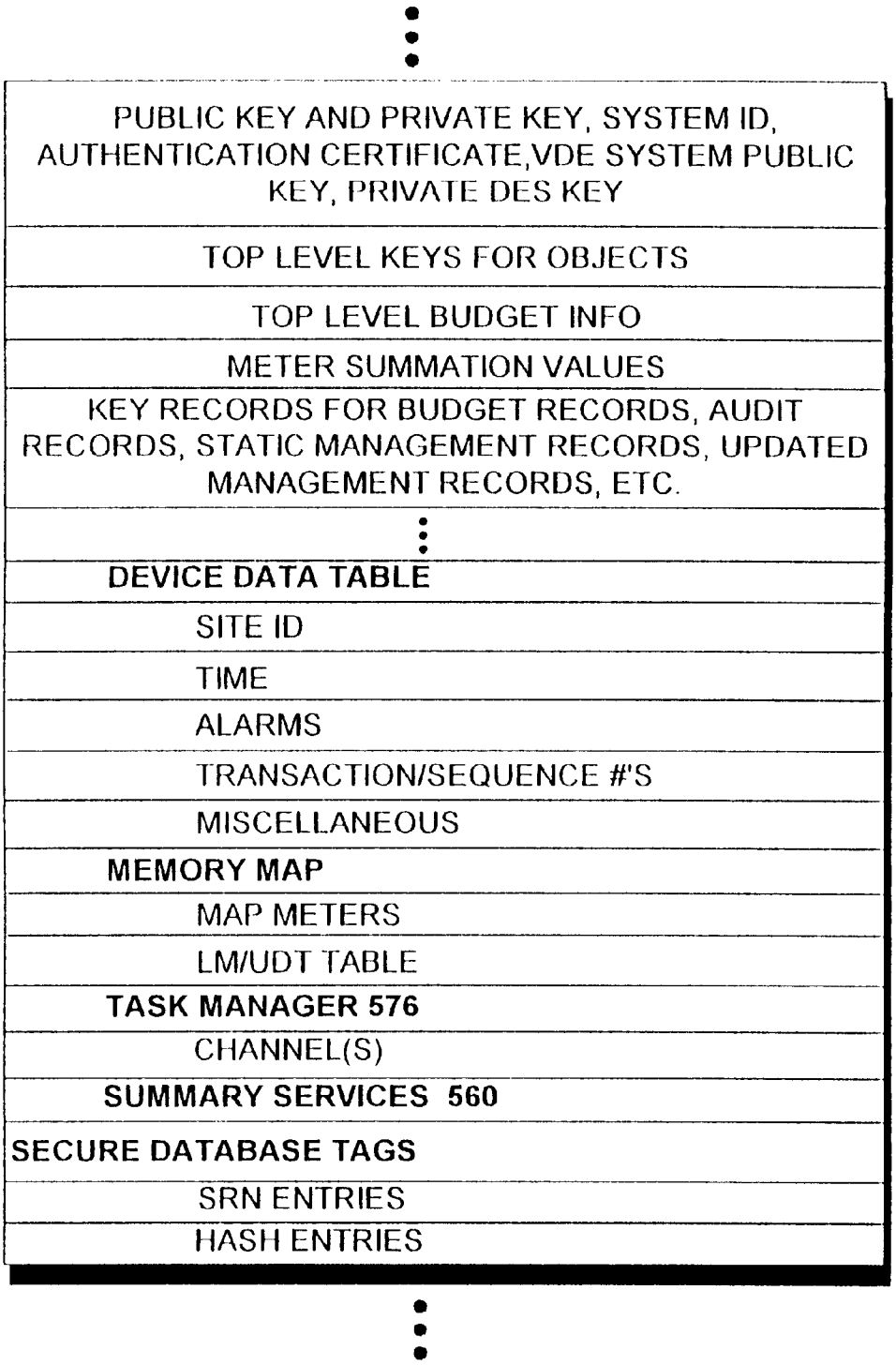

PUBLIC KEY AND PRIVATE KEY, SYSTEM ID, AUTHENTICATION CERTIFICATE,VDE SYSTEM PUBLIC KEY, PRIVATE DES KEY
TOP LEVEL KEYS FOR OBJECTS
TOP LEVEL BUDGET INFO
METER SUMMATION VALUES
KEY RECORDS FOR BUDGET RECORDS, AUDIT RECORDS, STATIC MANAGEMENT RECORDS, UPDATED MANAGEMENT RECORDS, ETC.
DEVICE DATA TABLE
SITE ID
TIME
ALARMS
TRANSACTION/SEQUENCE #'S
MISCELLANEOUS
MEMORY MAP
MAP METERS
LM/UDT TABLE
TASK MANAGER 576
CHANNEL(S)
SUMMARY SERVICES 560
SECURE DATABASE TAGS
SRN ENTRIES
HASH ENTRIES

FIG. 14C

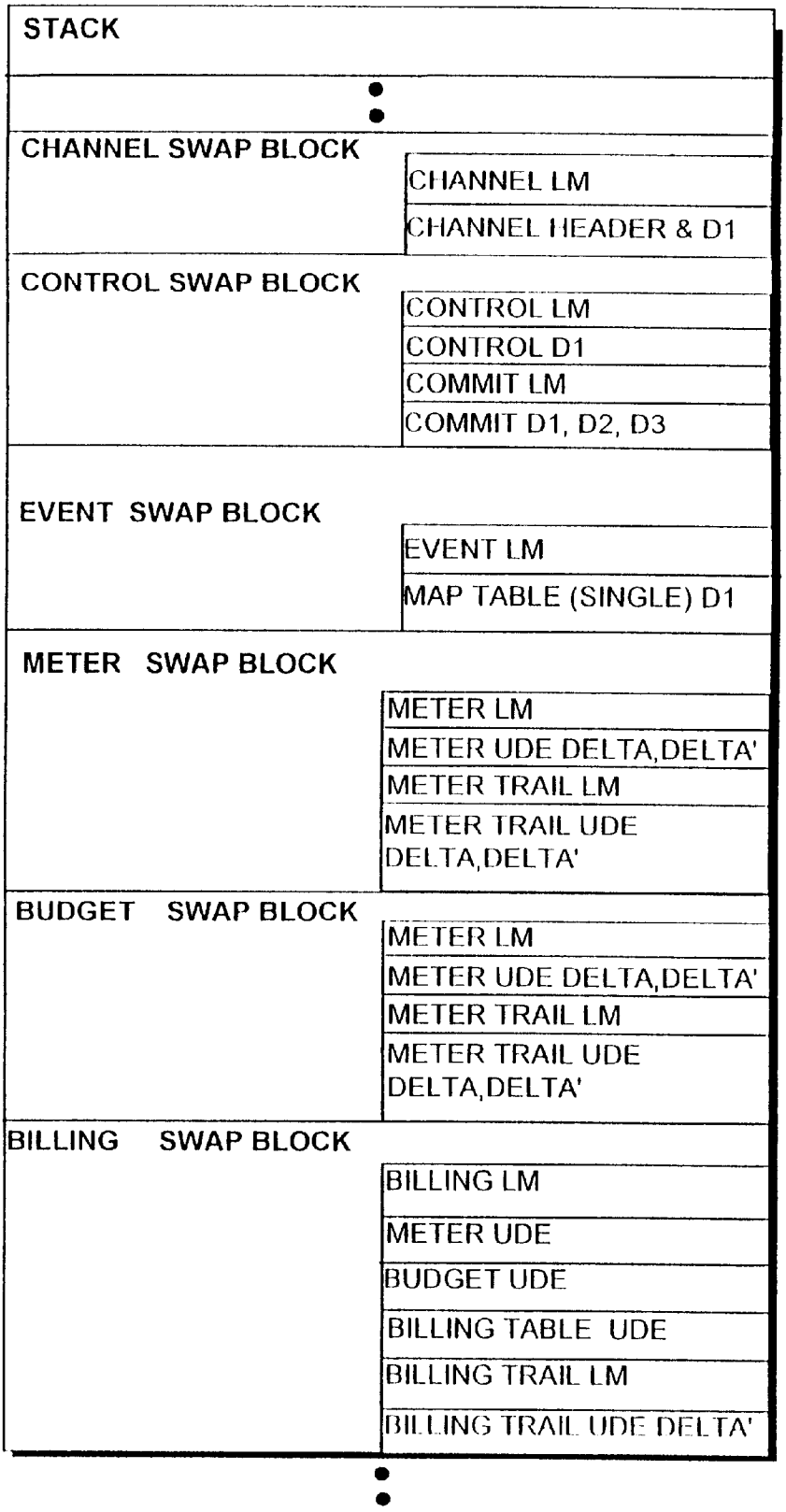

STACK
CHANNEL SWAP BLOCK
CHANNEL LM
CHANNEL HEADER & D1
CONTROL SWAP BLOCK
CONTROL LM
CONTROL D1
COMMIT LM
COMMIT D1, D2, D3
EVENT SWAP BLOCK
EVENT LM
MAP TABLE (SINGLE) D1
METER SWAP BLOCK
METER LM
METER UDE DELTA,DELTA'
METER TRAIL LM
METER TRAIL UDE DELTA,DELTA'
BUDGET SWAP BLOCK
METER LM
METER UDE DELTA,DELTA'
METER TRAIL LM
METER TRAIL UDE DELTA,DELTA'
BILLING SWAP BLOCK
BILLING LM
METER UDE
BUDGET UDE
BILLING TABLE UDE
BILLING TRAIL LM
BILLING TRAIL UDE DELTA'

FIG. 15

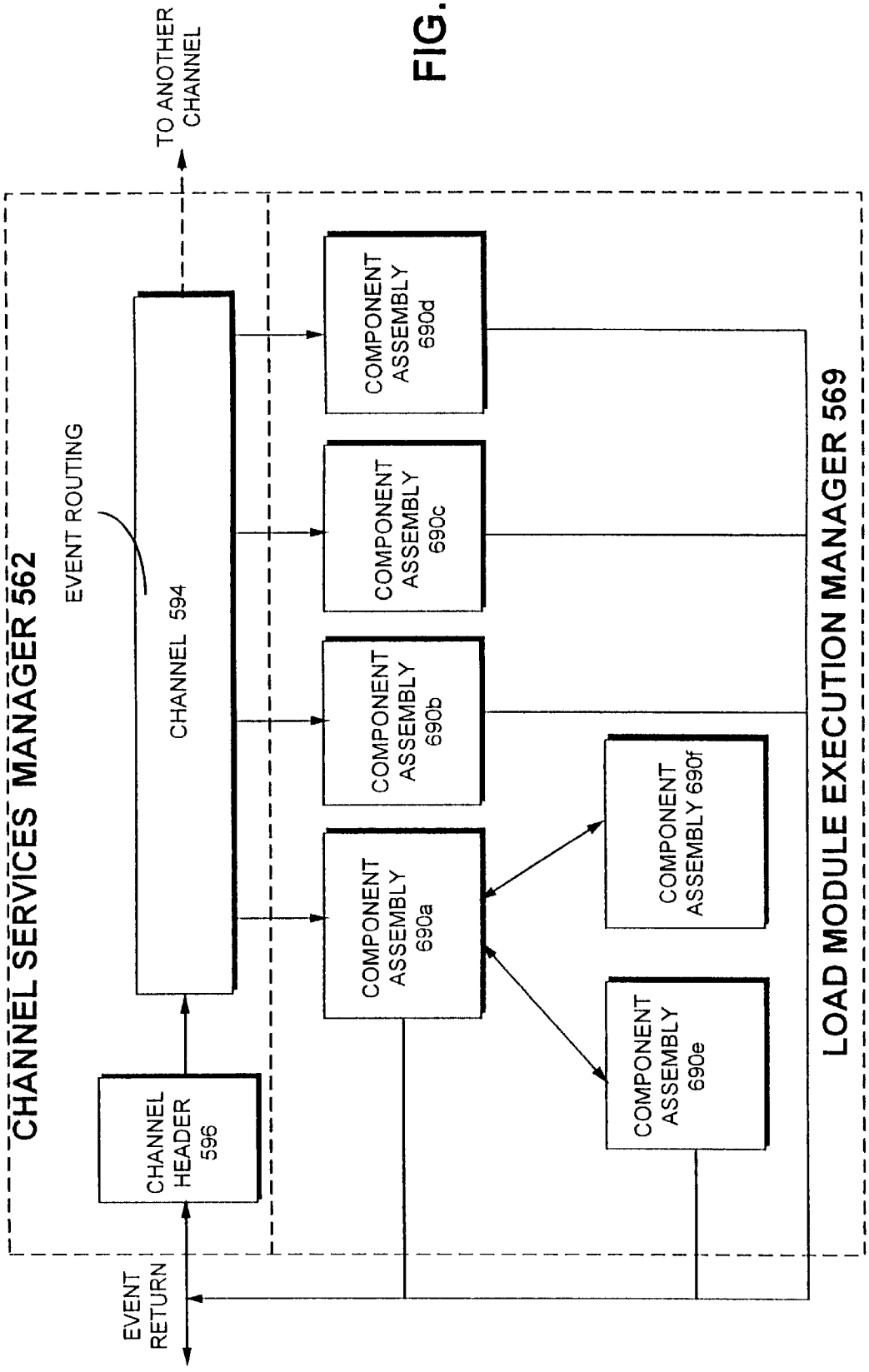

TO ANOTHER CHANNEL
CHANNEL SERVICES MANAGER 562
EVENT ROUTING
CHANNEL 594
CHANNEL HEADER 596
COMPONENT ASSEMBLY 690a
COMPONENT ASSEMBLY 690b
COMPONENT ASSEMBLY 690c
COMPONENT ASSEMBLY 690d
COMPONENT ASSEMBLY 690e
COMPONENT ASSEMBLY 690f
LOAD MODULE EXECUTION MANAGER 569
EVENT RETURN

FIG. 15A

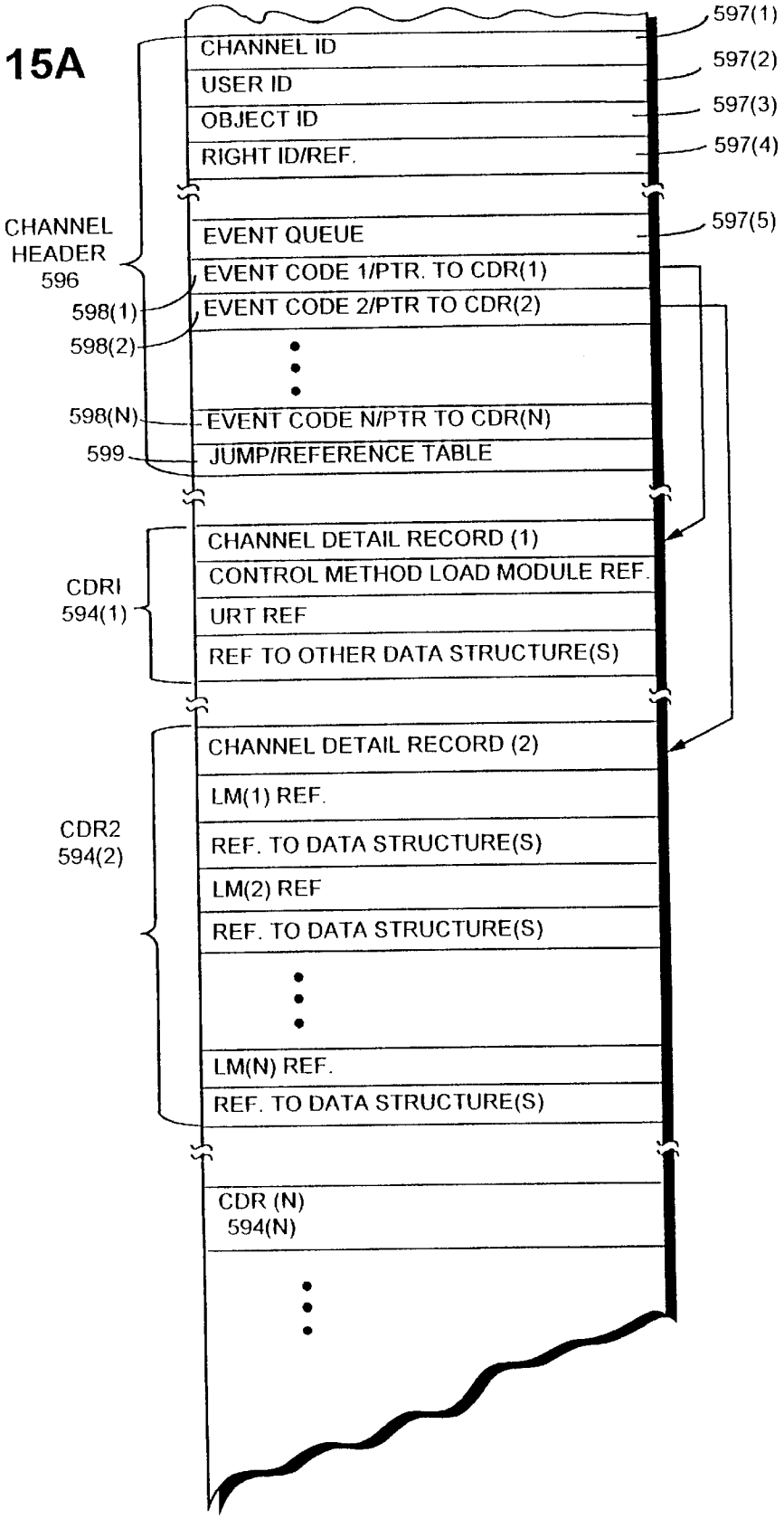

597(1)
CHANNEL ID
597(2)
USER ID
597(3)
OBJECT ID
597(4)
RIGHT ID/REF.
CHANNEL HEADER 596
597(5)
EVENT QUEUE
EVENT CODE 1/PTR. TO CDR(1)
598(1)
EVENT CODE 2/PTR TO CDR(2)
598(2)
598(N)
EVENT CODE N/PTR TO CDR(N)
599
JUMP/REFERENCE TABLE
CDRI 594(1)
CHANNEL DETAIL RECORD (1)
CONTROL METHOD LOAD MODULE REF.
URT REF
REF TO OTHER DATA STRUCTURE(S)
CDR2 594(2)
CHANNEL DETAIL RECORD (2)
LM(1) REF.
REF. TO DATA STRUCTURE(S)
LM(2) REF
REF. TO DATA STRUCTURE(S)
LM(N) REF.
REF. TO DATA STRUCTURE(S)
CDR (N) 594(N)

FIG. 15B

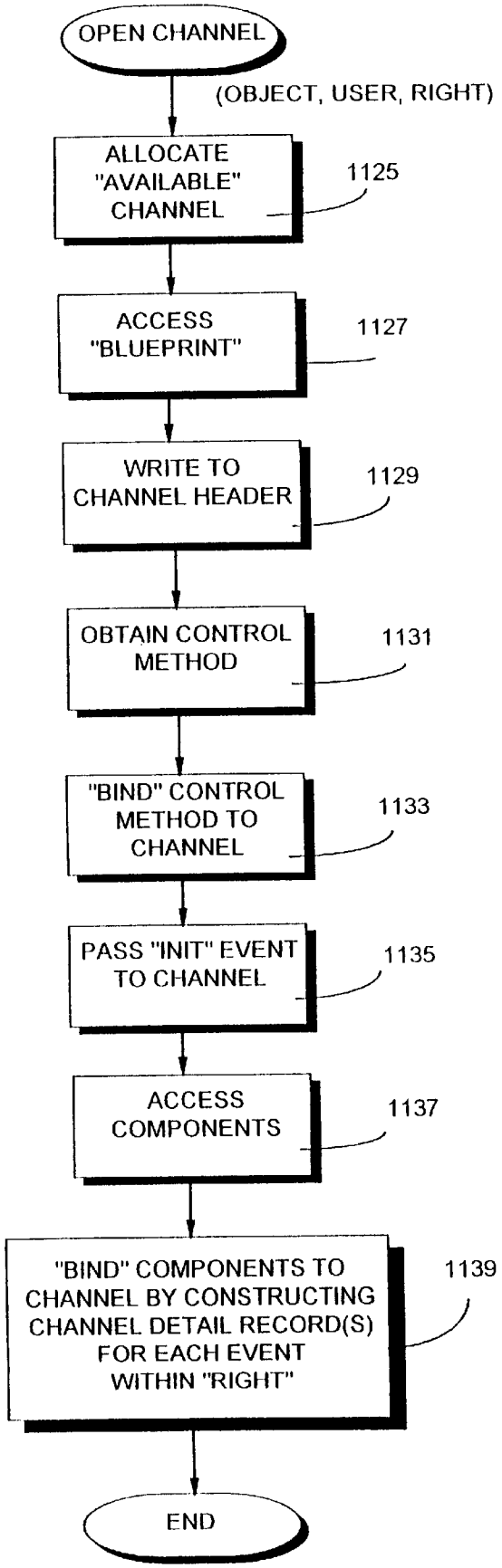
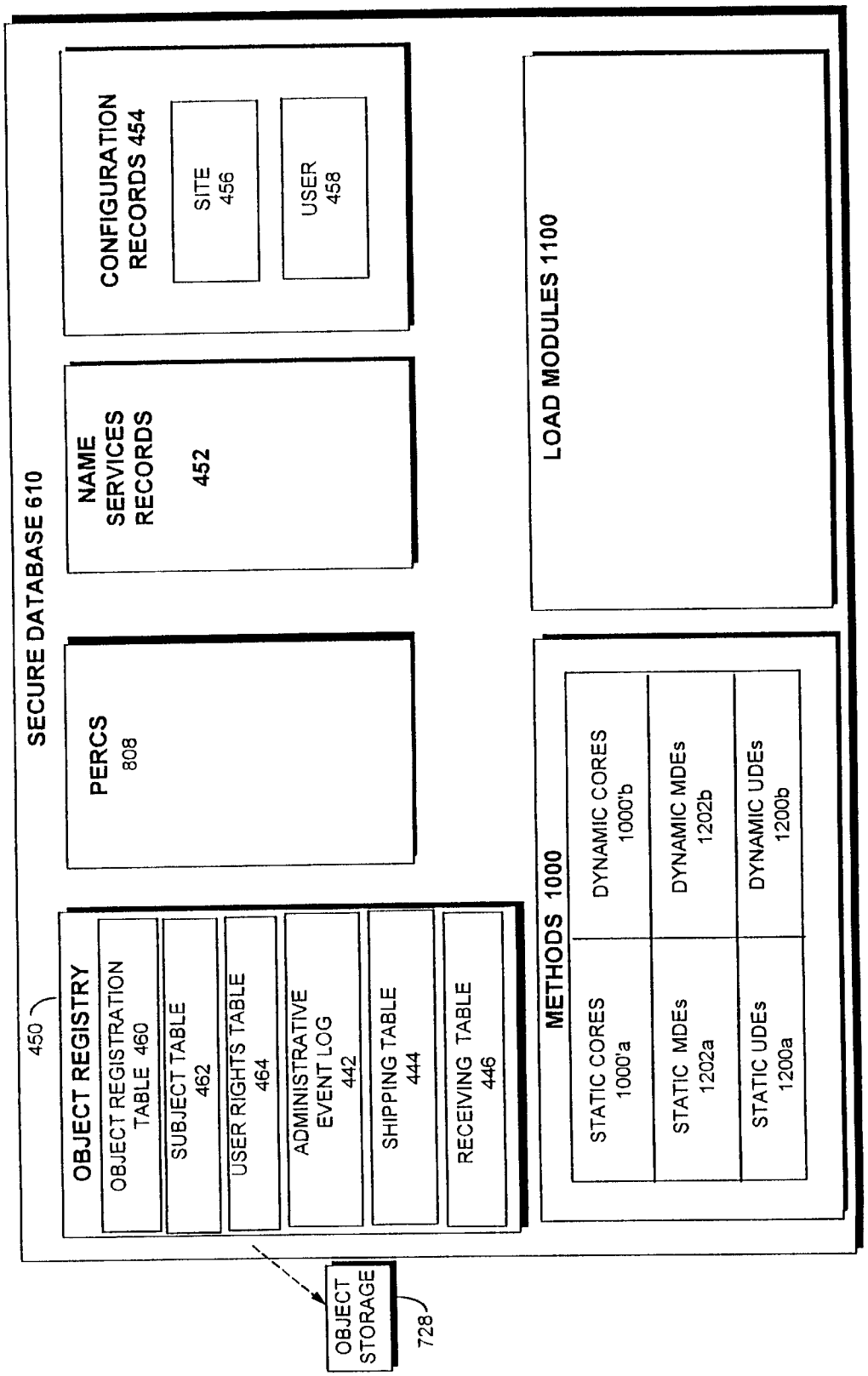

OPEN CHANNEL
(OBJECT, USER, RIGHT)
ALLOCATE "AVAILABLE" CHANNEL
1125
ACCESS "BLUEPRINT"
1127
WRITE TO CHANNEL HEADER
1129
OBTAIN CONTROL METHOD
1131
"BIND" CONTROL METHOD TO CHANNEL
1133
PASS "INIT" EVENT TO CHANNEL
1135
ACCESS COMPONENTS
1137
"BIND" COMPONENTS TO CHANNEL BY CONSTRUCTING CHANNEL DETAIL RECORD(S) FOR EACH EVENT WITHIN "RIGHT"
1139
END

FIG. 16


SECURE DATABASE 610
450
OBJECT REGISTRY
OBJECT REGISTRATION TABLE 460
SUBJECT TABLE 462
USER RIGHTS TABLE 464
ADMINISTRATIVE EVENT LOG 442
SHIPPING TABLE 444
RECEIVING TABLE 446
PERCS 808
NAME SERVICES RECORDS 452
CONFIGURATION RECORDS 454
SITE 456
USER 458
LOAD MODULES 1100
METHODS 1000
STATIC CORES 1000'a
DYNAMIC CORES 1000'b
STATIC MDEs 1202a
DYNAMIC MDEs 1202b
STATIC UDEs 1200a
DYNAMIC UDEs 1200b
OBJECT STORAGE
728

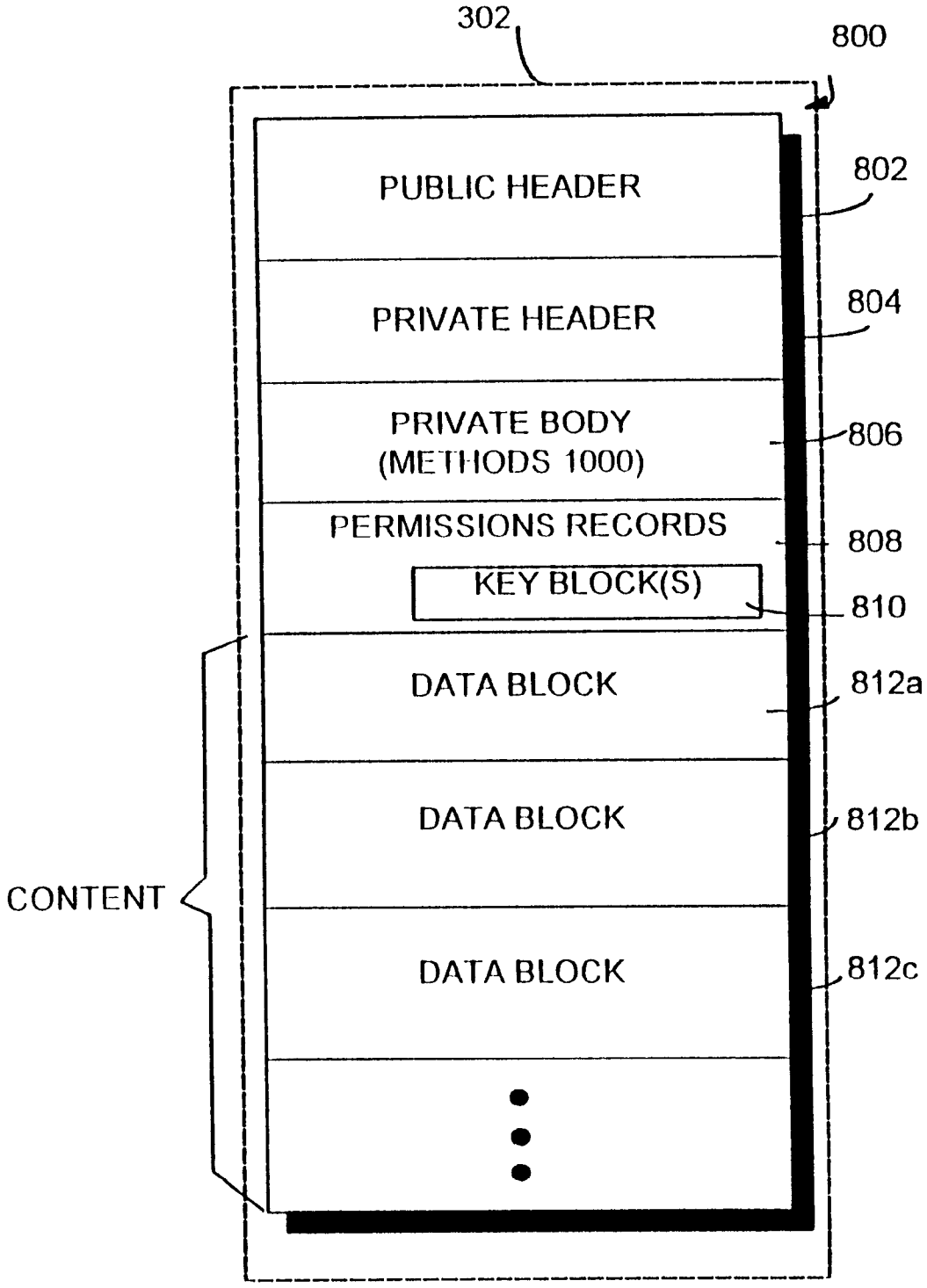

302
800
PUBLIC HEADER
802
PRIVATE HEADER
804
PRIVATE BODY
(METHODS 1000)
806
PERMISSIONS RECORDS
808
KEY BLOCK(S)
810
DATA BLOCK
812a
DATA BLOCK
812b
DATA BLOCK
812c
CONTENT

FIG. 17

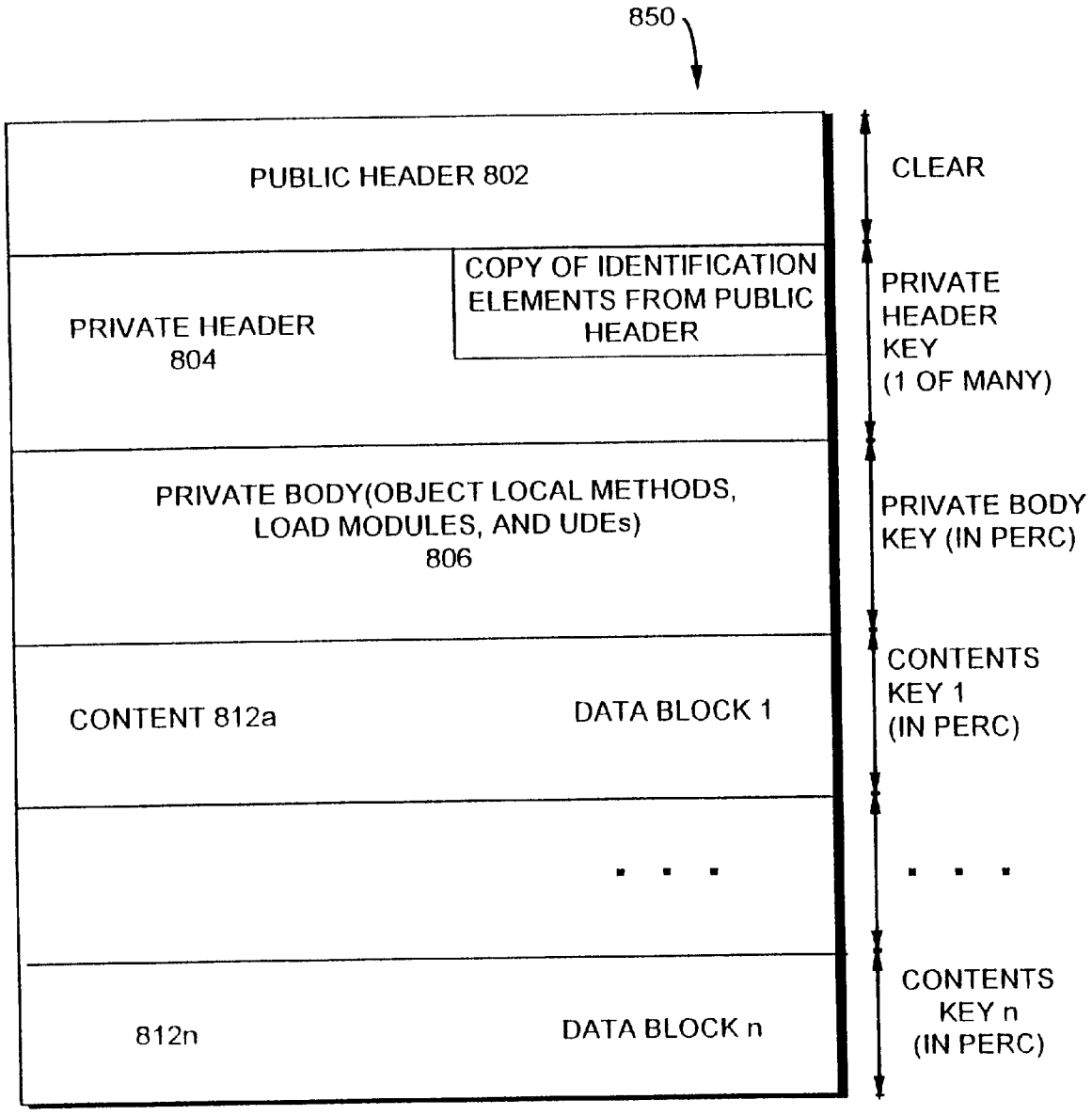

850
PUBLIC HEADER 802
CLEAR
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
PRIVATE HEADER 804
PRIVATE HEADER KEY (1 OF MANY)
PRIVATE BODY(OBJECT LOCAL METHODS, LOAD MODULES, AND UDEs) 806
PRIVATE BODY KEY (IN PERC)
CONTENT 812a
DATA BLOCK 1
CONTENTS KEY 1 (IN PERC)
. . .
. . .
812n
DATA BLOCK n
CONTENTS KEY n (IN PERC)

FIG. 18

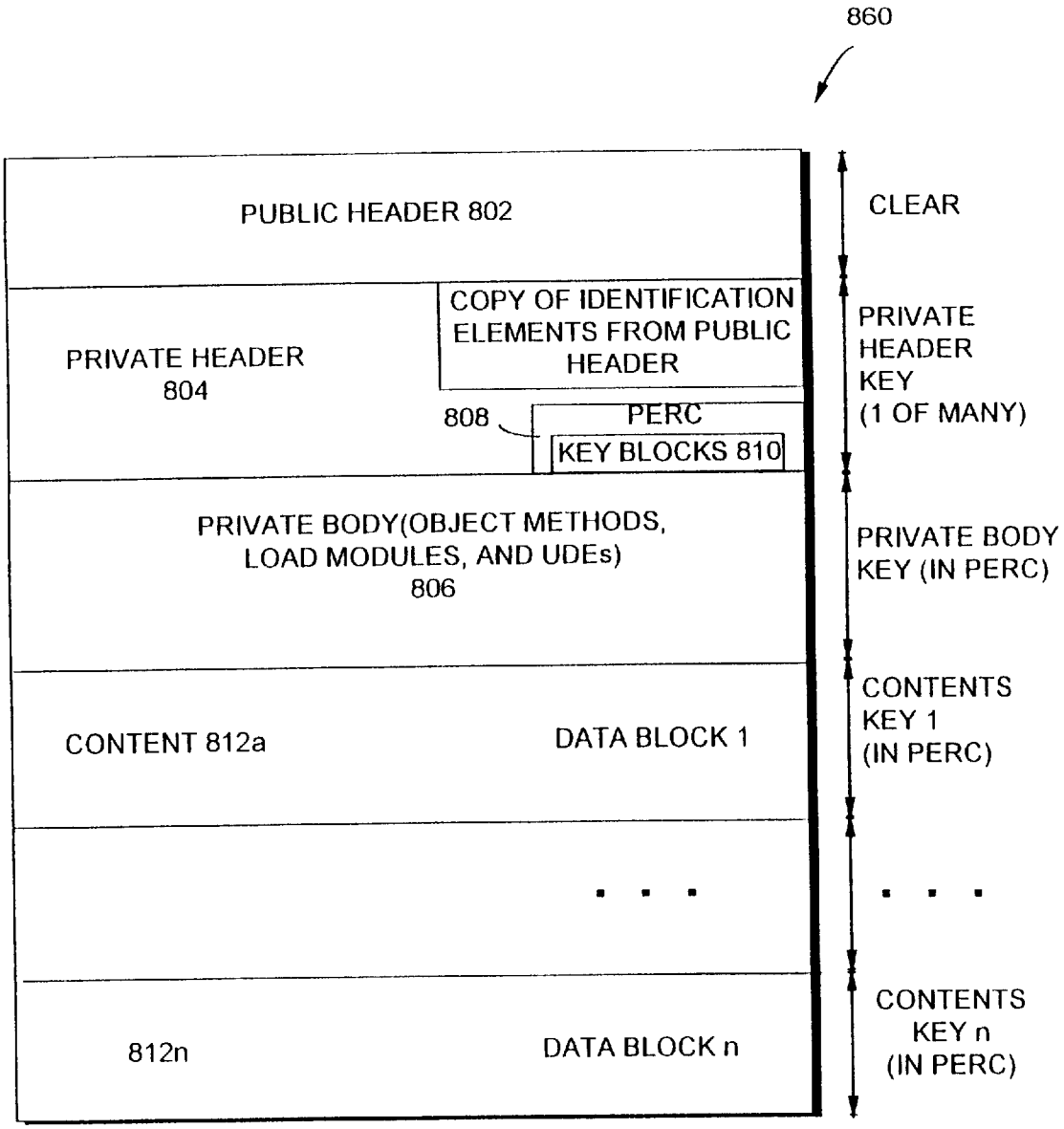

860
PUBLIC HEADER 802
PRIVATE HEADER 804
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
808
PERC
KEY BLOCKS 810
PRIVATE BODY(OBJECT METHODS, LOAD MODULES, AND UDEs) 806
CONTENT 812a
DATA BLOCK 1
812n
DATA BLOCK n
CLEAR
PRIVATE HEADER KEY (1 OF MANY)
PRIVATE BODY KEY (IN PERC)
CONTENTS KEY 1 (IN PERC)
CONTENTS KEY n (IN PERC)

FIG. 19

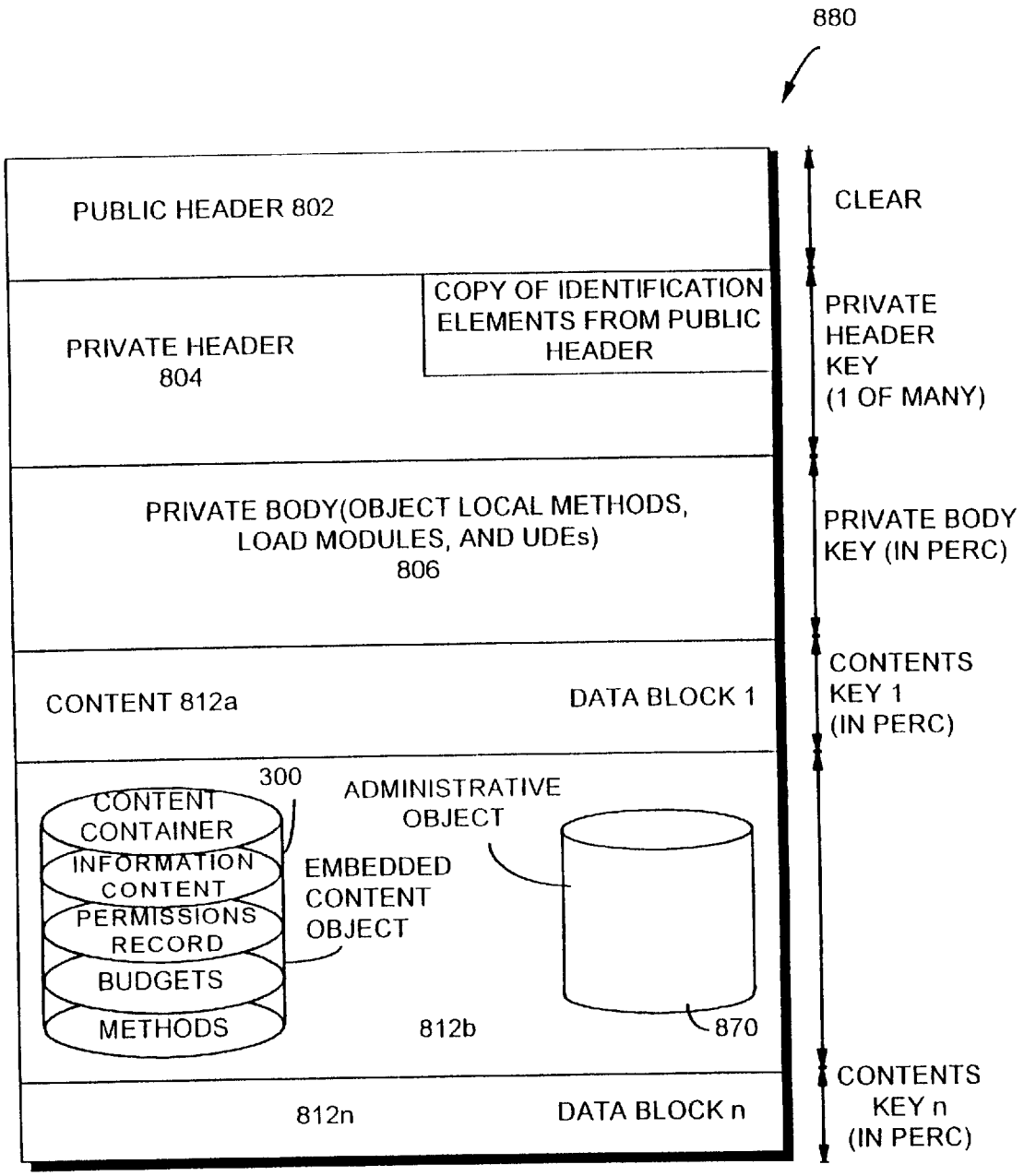

880
PUBLIC HEADER 802
CLEAR
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
PRIVATE HEADER 804
PRIVATE HEADER KEY (1 OF MANY)
PRIVATE BODY(OBJECT LOCAL METHODS, LOAD MODULES, AND UDEs) 806
PRIVATE BODY KEY (IN PERC)
CONTENT 812a
DATA BLOCK 1
CONTENTS KEY 1 (IN PERC)
300
CONTENT CONTAINER
INFORMATION CONTENT
PERMISSIONS RECORD
BUDGETS
METHODS
EMBEDDED CONTENT OBJECT
ADMINISTRATIVE OBJECT
812b
870
812n
DATA BLOCK n
CONTENTS KEY n (IN PERC)

FIG. 20

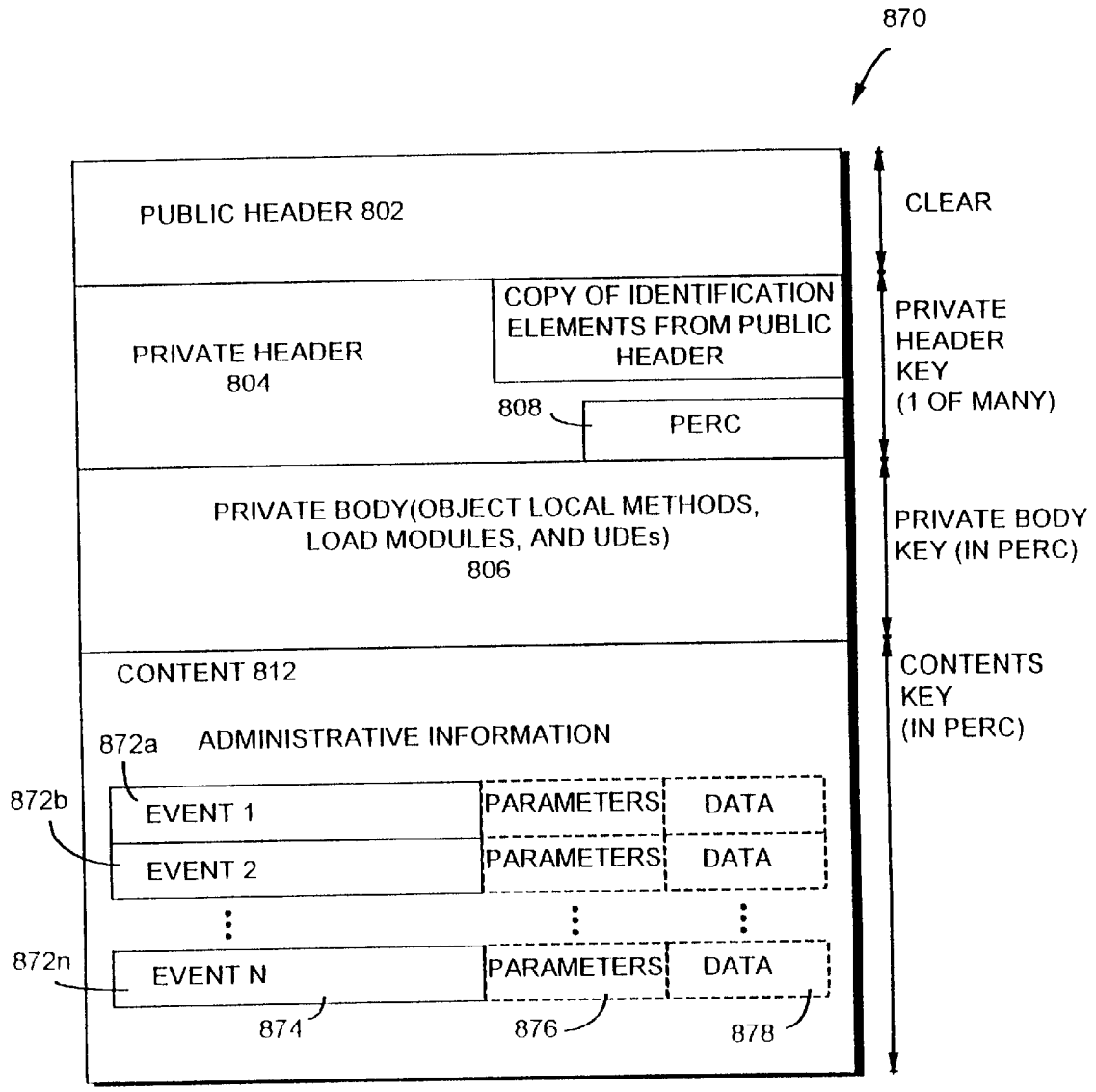

870
PUBLIC HEADER 802
CLEAR
PRIVATE HEADER 804
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
808
PERC
PRIVATE HEADER KEY (1 OF MANY)
PRIVATE BODY(OBJECT LOCAL METHODS, LOAD MODULES, AND UDEs) 806
PRIVATE BODY KEY (IN PERC)
CONTENT 812
CONTENTS KEY (IN PERC)
872a
ADMINISTRATIVE INFORMATION
872b
EVENT 1
PARAMETERS
DATA
EVENT 2
PARAMETERS
DATA
872n
EVENT N
PARAMETERS
DATA
874
876
878

FIG. 21

FIG. 22

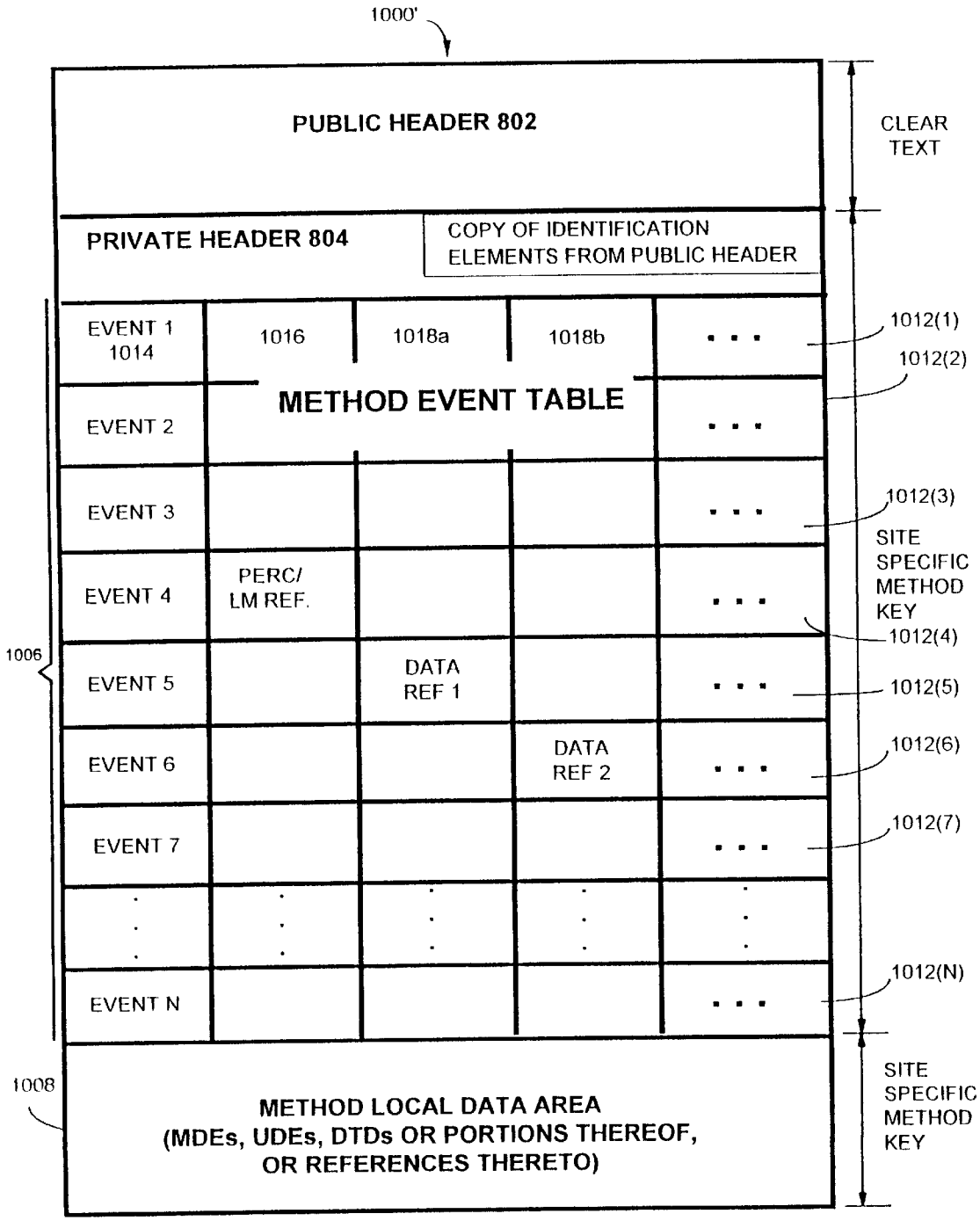

1000'
PUBLIC HEADER 802
CLEAR TEXT
PRIVATE HEADER 804
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
EVENT 1 1014
1016
1018a
1018b
1012(1)
1012(2)
METHOD EVENT TABLE
EVENT 2
EVENT 3
1012(3)
SITE SPECIFIC METHOD KEY
EVENT 4
PERC/ LM REF.
1012(4)
1006
EVENT 5
DATA REF 1
1012(5)
EVENT 6
DATA REF 2
1012(6)
EVENT 7
1012(7)
EVENT N
1012(N)
1008
METHOD LOCAL DATA AREA
(MDEs, UDEs, DTDs OR PORTIONS THEREOF, OR REFERENCES THERETO)
SITE SPECIFIC METHOD KEY

FIG. 23

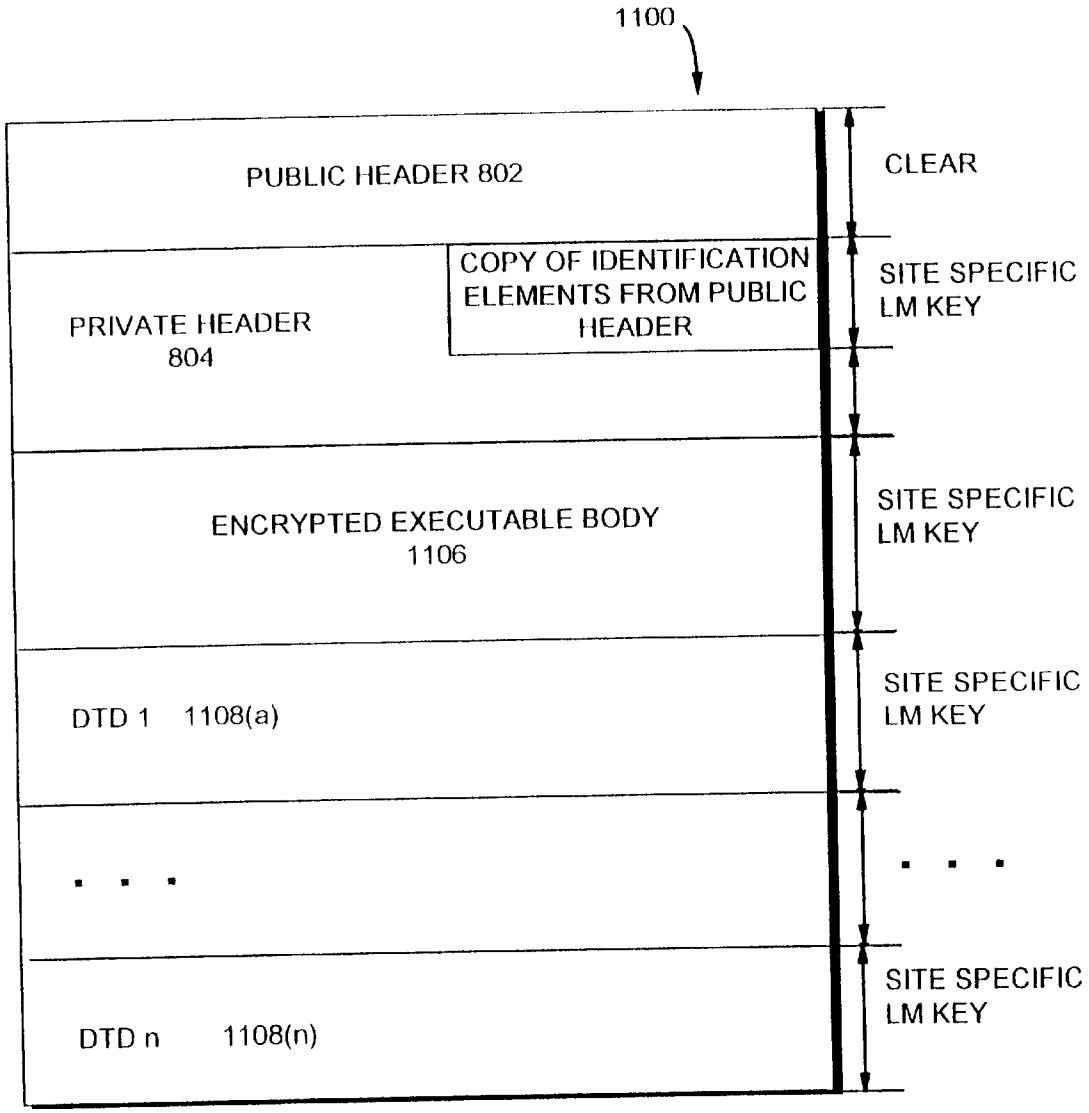

1100
PUBLIC HEADER 802
CLEAR
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
PRIVATE HEADER 804
SITE SPECIFIC LM KEY
ENCRYPTED EXECUTABLE BODY 1106
SITE SPECIFIC LM KEY
DTD 1 1108(a)
SITE SPECIFIC LM KEY
. . .
. . .
DTD n 1108(n)
SITE SPECIFIC LM KEY

FIG. 24

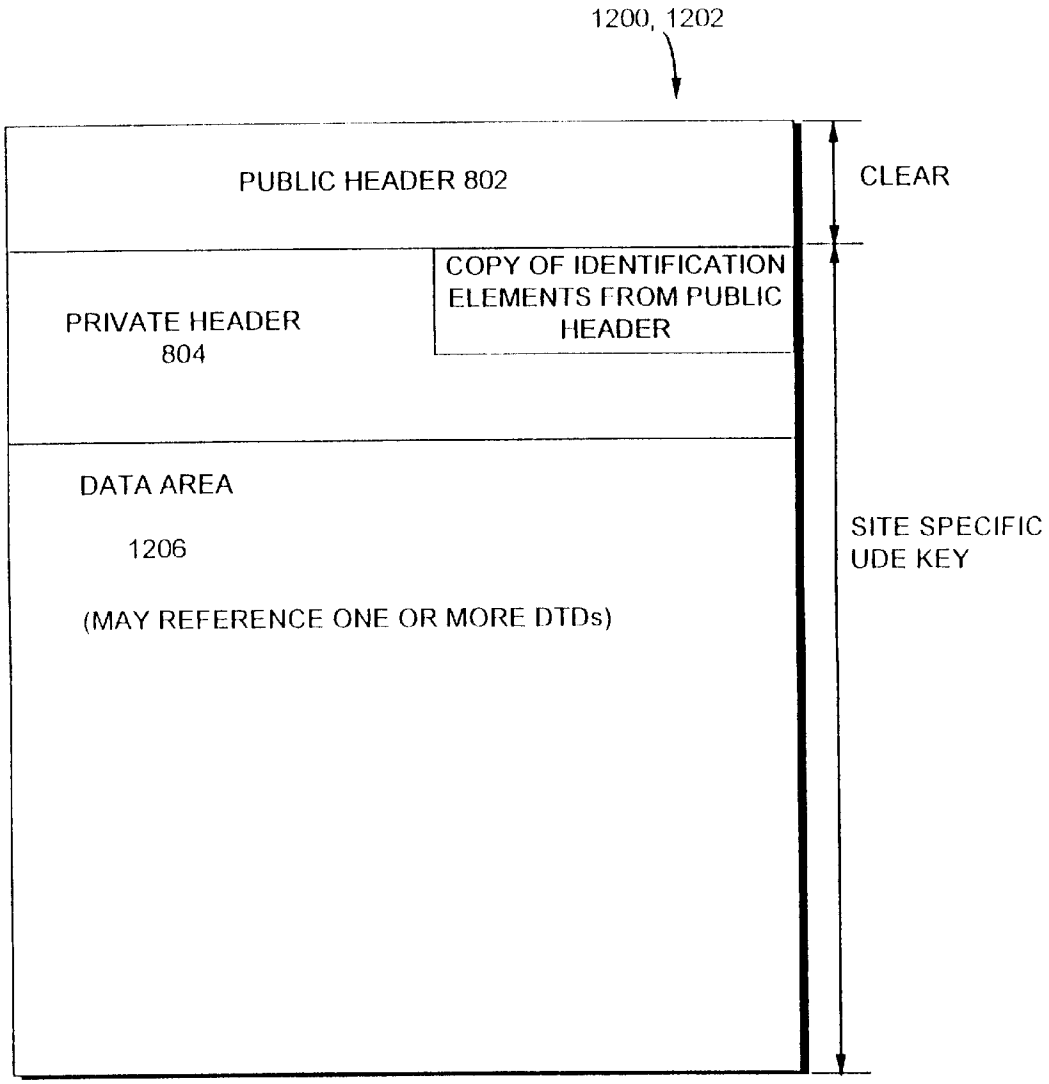

1200, 1202
PUBLIC HEADER 802
CLEAR
COPY OF IDENTIFICATION ELEMENTS FROM PUBLIC HEADER
PRIVATE HEADER 804
DATA AREA
1206
(MAY REFERENCE ONE OR MORE DTDs)
SITE SPECIFIC UDE KEY

FIG. 25A

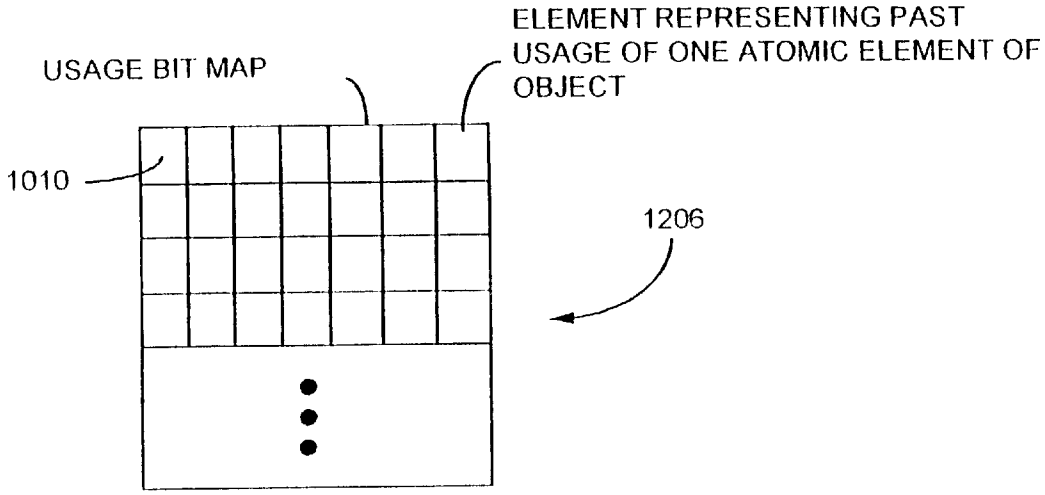


FIG. 25B

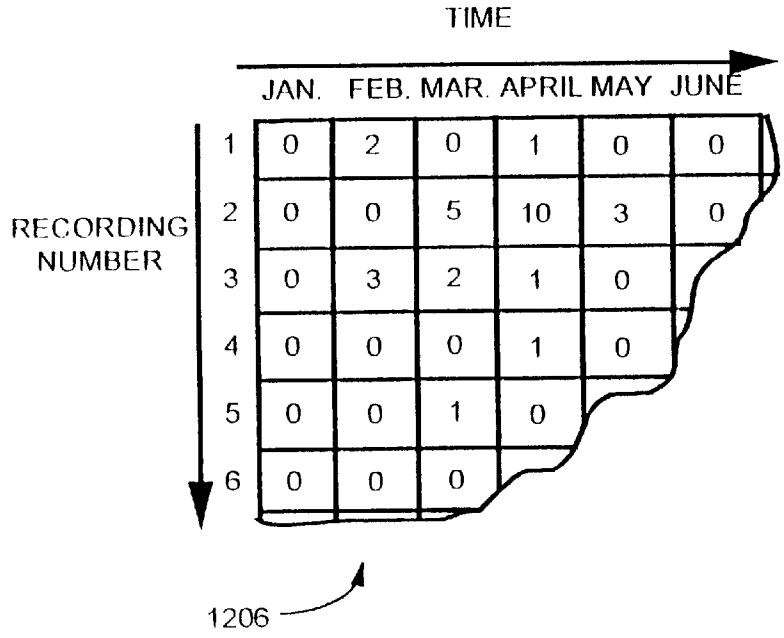


FIG. 25C

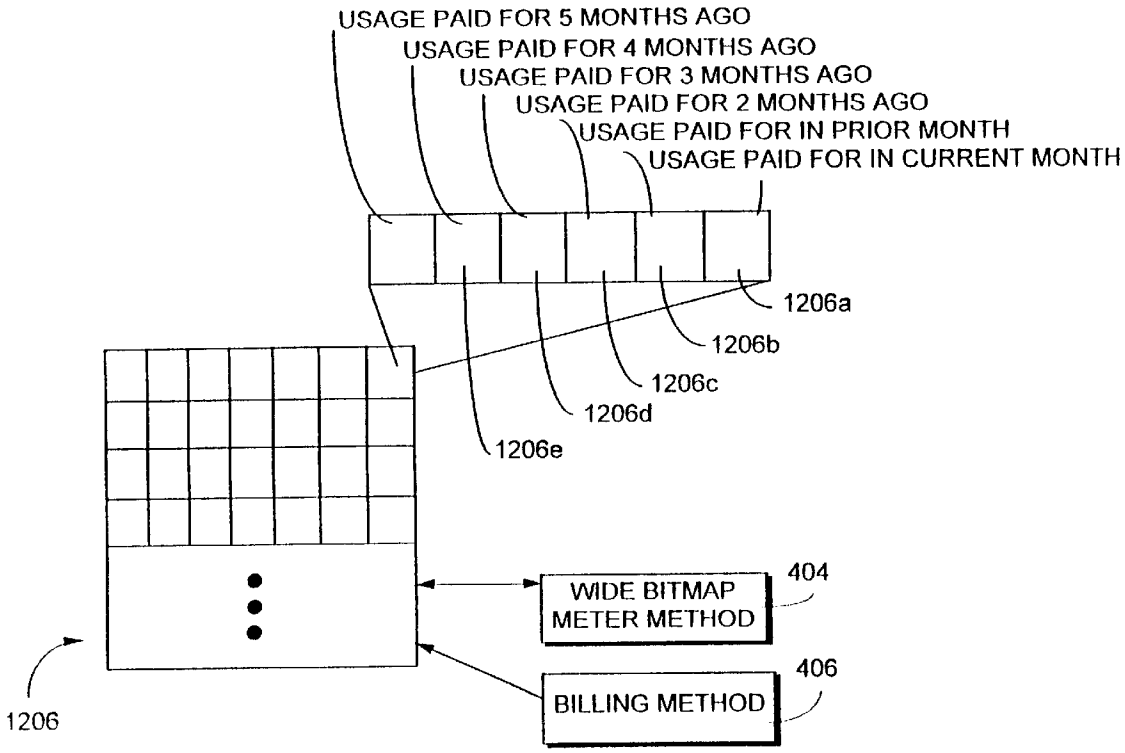

USAGE PAID FOR 5 MONTHS AGO
USAGE PAID FOR 4 MONTHS AGO
USAGE PAID FOR 3 MONTHS AGO
USAGE PAID FOR 2 MONTHS AGO
USAGE PAID FOR IN PRIOR MONTH
USAGE PAID FOR IN CURRENT MONTH
1206a
1206b
1206c
1206d
1206e
WIDE BITMAP METER METHOD
404
BILLING METHOD
406
1206

FIG. 26

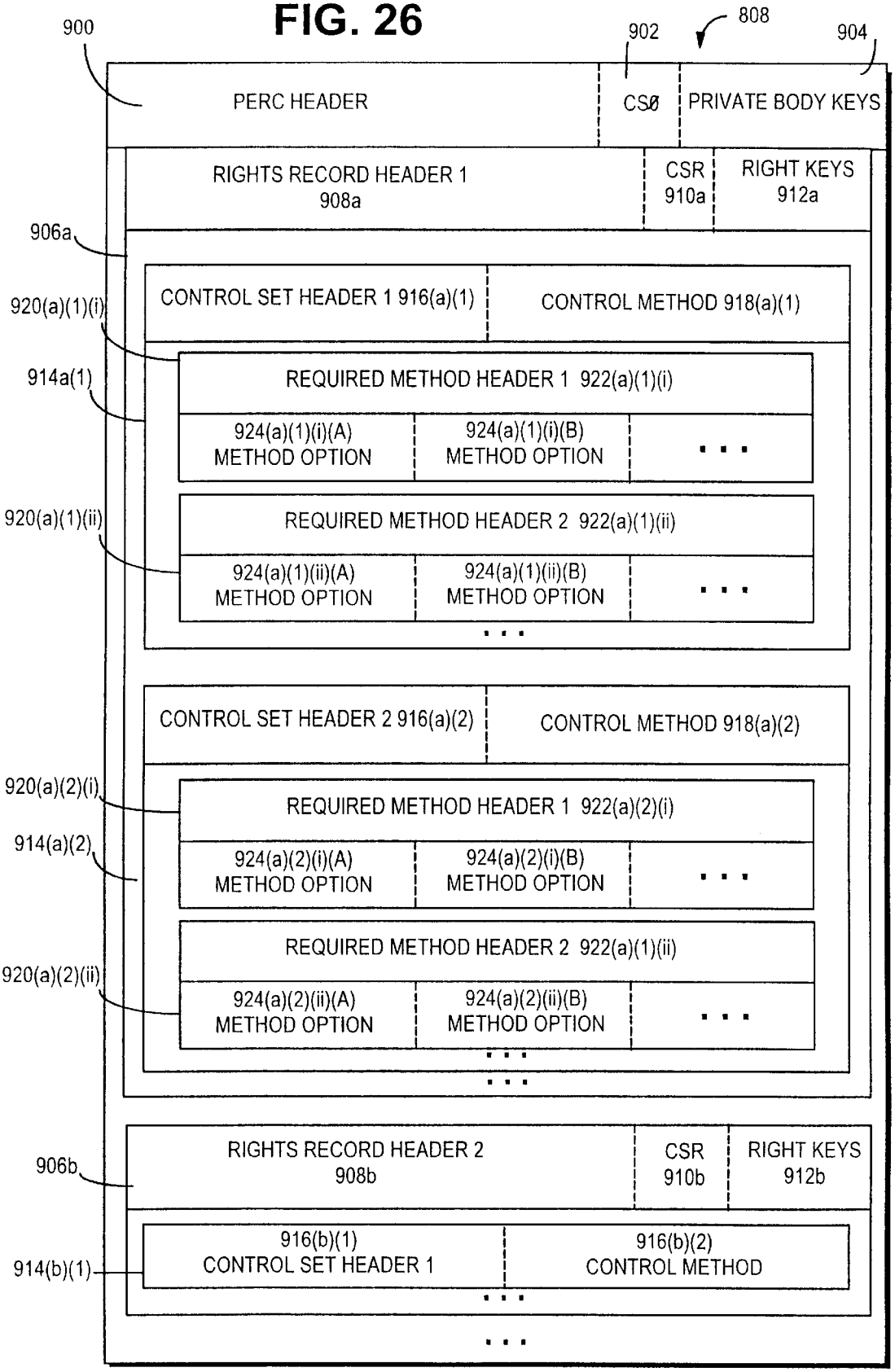

900
902
808
904
PERC HEADER
CSØ
PRIVATE BODY KEYS
RIGHTS RECORD HEADER 1
908a
CSR
910a
RIGHT KEYS
912a
906a
CONTROL SET HEADER 1 916(a)(1)
CONTROL METHOD 918(a)(1)
920(a)(1)(i)
914a(1)
REQUIRED METHOD HEADER 1 922(a)(1)(i)
924(a)(1)(i)(A)
METHOD OPTION
924(a)(1)(i)(B)
METHOD OPTION
920(a)(1)(ii)
REQUIRED METHOD HEADER 2 922(a)(1)(ii)
924(a)(1)(ii)(A)
METHOD OPTION
924(a)(1)(ii)(B)
METHOD OPTION
CONTROL SET HEADER 2 916(a)(2)
CONTROL METHOD 918(a)(2)
920(a)(2)(i)
914(a)(2)
REQUIRED METHOD HEADER 1 922(a)(2)(i)
924(a)(2)(i)(A)
METHOD OPTION
924(a)(2)(i)(B)
METHOD OPTION
920(a)(2)(ii)
REQUIRED METHOD HEADER 2 922(a)(1)(ii)
924(a)(2)(ii)(A)
METHOD OPTION
924(a)(2)(ii)(B)
METHOD OPTION
906b
RIGHTS RECORD HEADER 2
908b
CSR
910b
RIGHT KEYS
912b
914(b)(1)
916(b)(1)
CONTROL SET HEADER 1
916(b)(2)
CONTROL METHOD

FIG. 26A

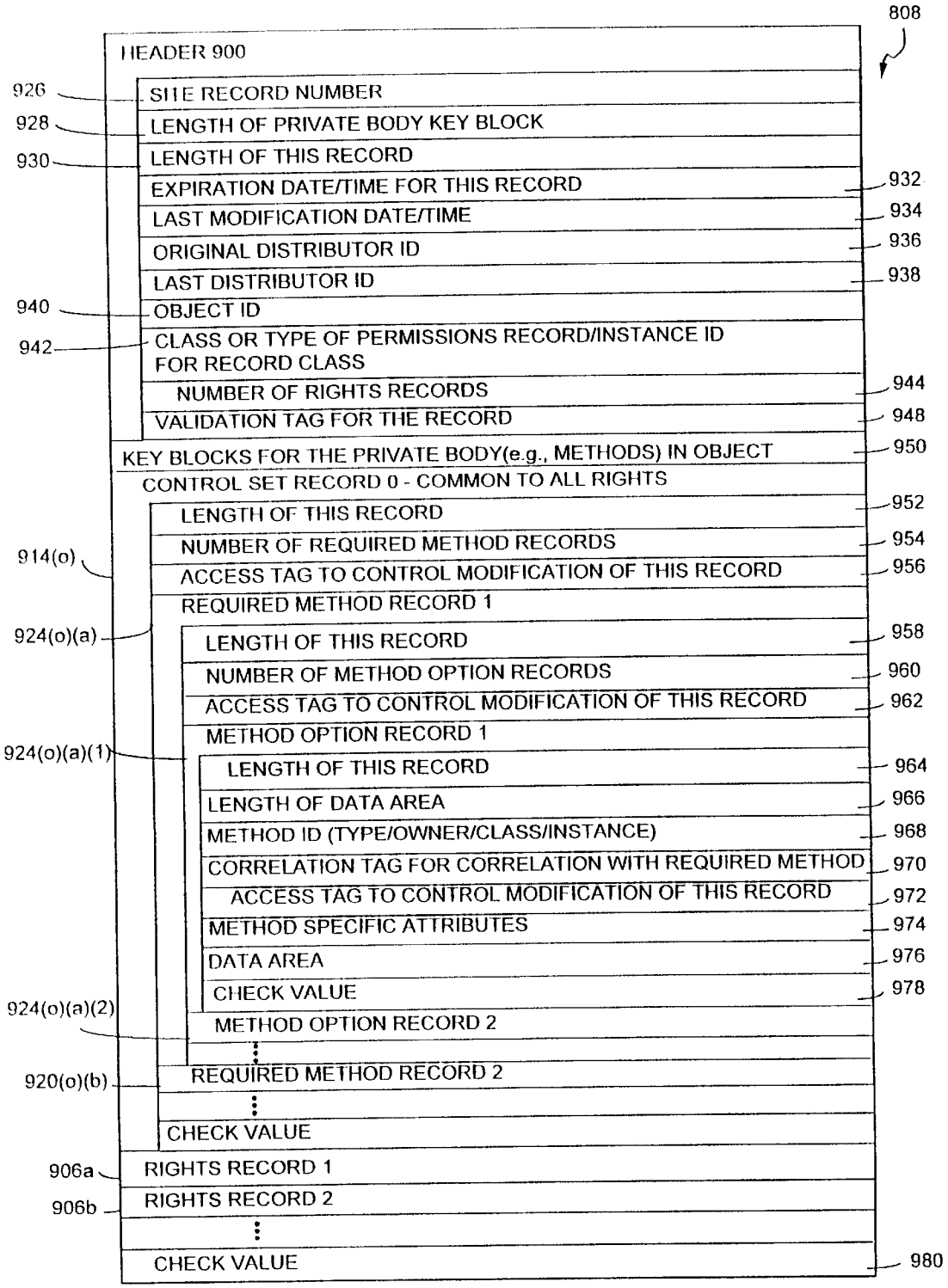

808
HEADER 900
926
SITE RECORD NUMBER
928
LENGTH OF PRIVATE BODY KEY BLOCK
930
LENGTH OF THIS RECORD
EXPIRATION DATE/TIME FOR THIS RECORD
932
LAST MODIFICATION DATE/TIME
934
ORIGINAL DISTRIBUTOR ID
936
LAST DISTRIBUTOR ID
938
940
OBJECT ID
942
CLASS OR TYPE OF PERMISSIONS RECORD/INSTANCE ID FOR RECORD CLASS
NUMBER OF RIGHTS RECORDS
944
VALIDATION TAG FOR THE RECORD
948
KEY BLOCKS FOR THE PRIVATE BODY(e.g., METHODS) IN OBJECT
950
CONTROL SET RECORD 0 - COMMON TO ALL RIGHTS
LENGTH OF THIS RECORD
952
NUMBER OF REQUIRED METHOD RECORDS
954
914(o)
ACCESS TAG TO CONTROL MODIFICATION OF THIS RECORD
956
REQUIRED METHOD RECORD 1
924(o)(a)
LENGTH OF THIS RECORD
958
NUMBER OF METHOD OPTION RECORDS
960
ACCESS TAG TO CONTROL MODIFICATION OF THIS RECORD
962
METHOD OPTION RECORD 1
924(o)(a)(1)
LENGTH OF THIS RECORD
964
LENGTH OF DATA AREA
966
METHOD ID (TYPE/OWNER/CLASS/INSTANCE)
968
CORRELATION TAG FOR CORRELATION WITH REQUIRED METHOD
970
ACCESS TAG TO CONTROL MODIFICATION OF THIS RECORD
972
METHOD SPECIFIC ATTRIBUTES
974
DATA AREA
976
CHECK VALUE
978
924(o)(a)(2)
METHOD OPTION RECORD 2
920(o)(b)
REQUIRED METHOD RECORD 2
CHECK VALUE
906a
RIGHTS RECORD 1
906b
RIGHTS RECORD 2
CHECK VALUE
980

FIG. 26B

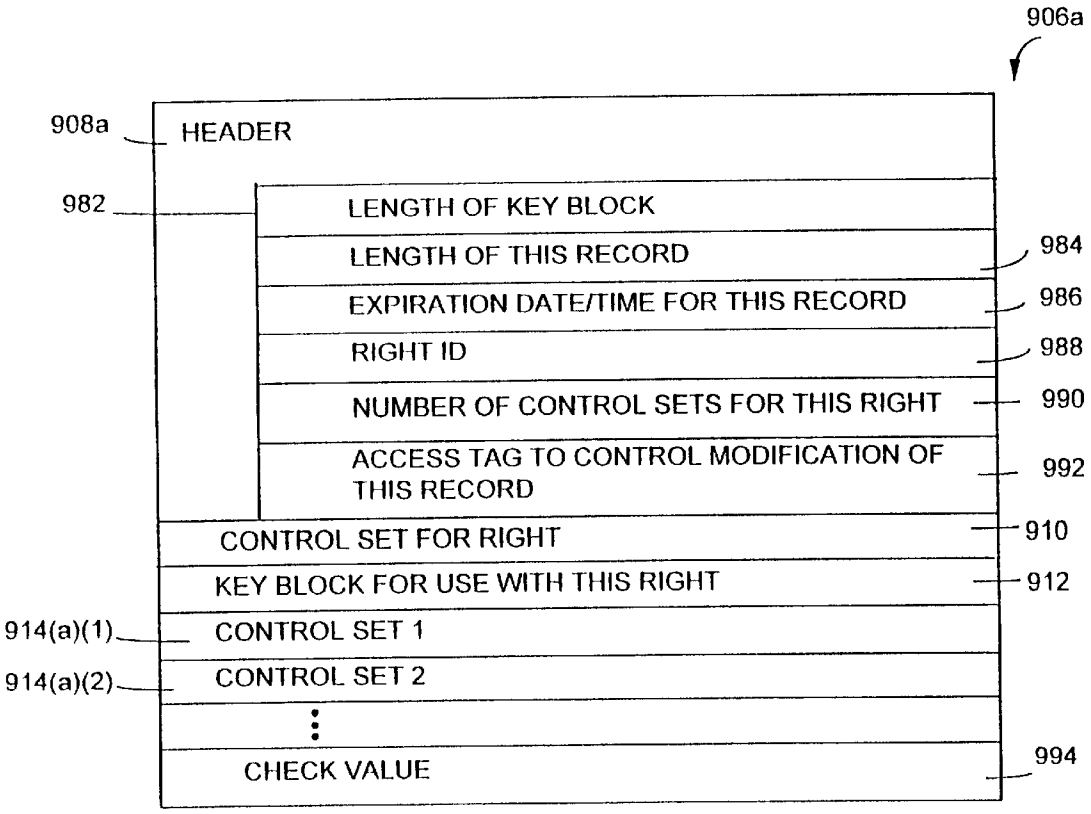

906a
908a
HEADER
982
LENGTH OF KEY BLOCK
LENGTH OF THIS RECORD
984
EXPIRATION DATE/TIME FOR THIS RECORD
986
RIGHT ID
988
NUMBER OF CONTROL SETS FOR THIS RIGHT
990
ACCESS TAG TO CONTROL MODIFICATION OF THIS RECORD
992
CONTROL SET FOR RIGHT
910
KEY BLOCK FOR USE WITH THIS RIGHT
912
914(a)(1)
CONTROL SET 1
914(a)(2)
CONTROL SET 2
CHECK VALUE
994

FIG. 27

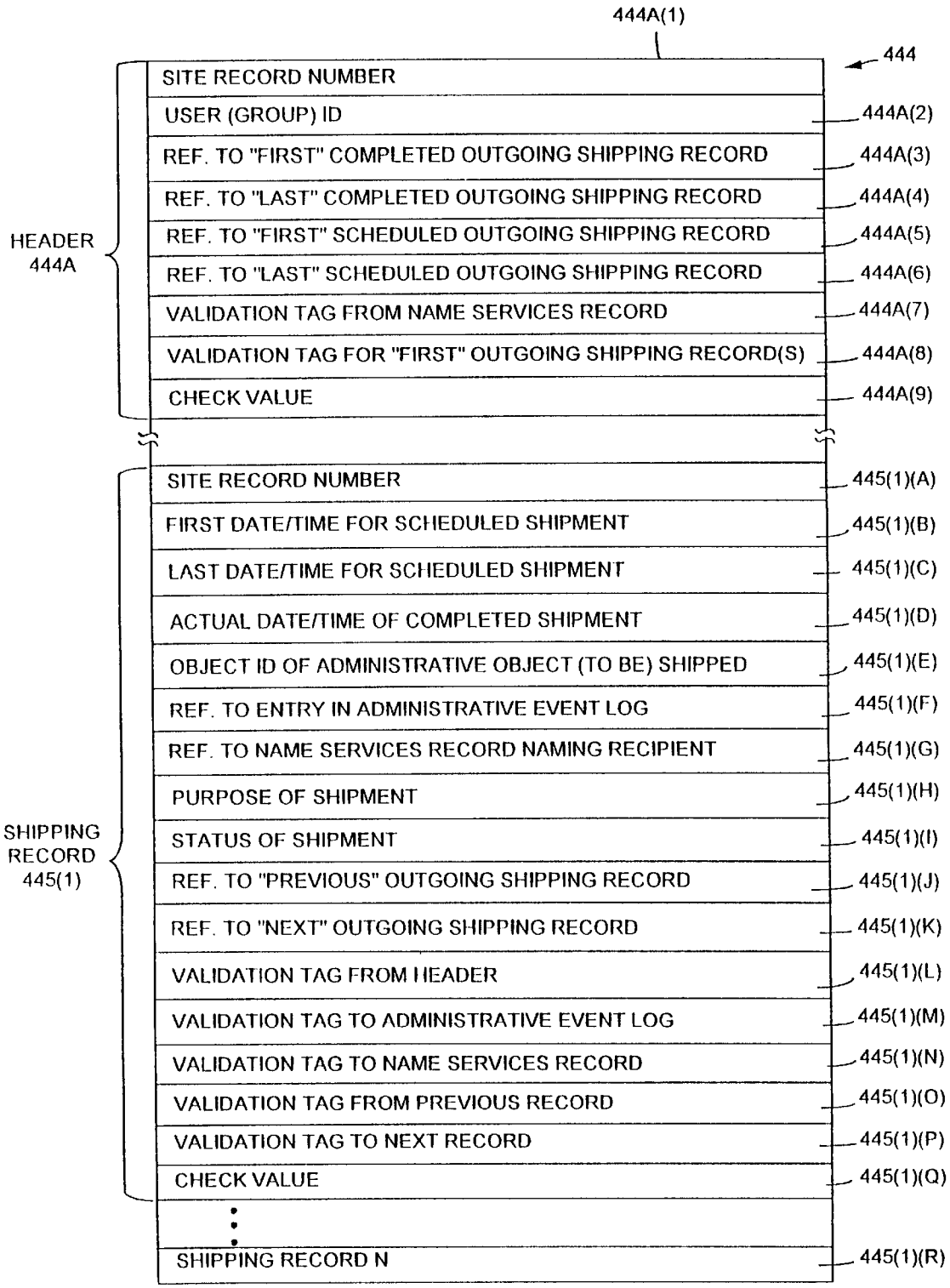

444A(1)
444
SITE RECORD NUMBER
USER (GROUP) ID
444A(2)
REF. TO "FIRST" COMPLETED OUTGOING SHIPPING RECORD
444A(3)
REF. TO "LAST" COMPLETED OUTGOING SHIPPING RECORD
444A(4)
REF. TO "FIRST" SCHEDULED OUTGOING SHIPPING RECORD
444A(5)
REF. TO "LAST" SCHEDULED OUTGOING SHIPPING RECORD
444A(6)
VALIDATION TAG FROM NAME SERVICES RECORD
444A(7)
VALIDATION TAG FOR "FIRST" OUTGOING SHIPPING RECORD(S)
444A(8)
CHECK VALUE
444A(9)
HEADER 444A
SITE RECORD NUMBER
445(1)(A)
FIRST DATE/TIME FOR SCHEDULED SHIPMENT
445(1)(B)
LAST DATE/TIME FOR SCHEDULED SHIPMENT
445(1)(C)
ACTUAL DATE/TIME OF COMPLETED SHIPMENT
445(1)(D)
OBJECT ID OF ADMINISTRATIVE OBJECT (TO BE) SHIPPED
445(1)(E)
REF. TO ENTRY IN ADMINISTRATIVE EVENT LOG
445(1)(F)
REF. TO NAME SERVICES RECORD NAMING RECIPIENT
445(1)(G)
PURPOSE OF SHIPMENT
445(1)(H)
STATUS OF SHIPMENT
445(1)(I)
REF. TO "PREVIOUS" OUTGOING SHIPPING RECORD
445(1)(J)
REF. TO "NEXT" OUTGOING SHIPPING RECORD
445(1)(K)
VALIDATION TAG FROM HEADER
445(1)(L)
VALIDATION TAG TO ADMINISTRATIVE EVENT LOG
445(1)(M)
VALIDATION TAG TO NAME SERVICES RECORD
445(1)(N)
VALIDATION TAG FROM PREVIOUS RECORD
445(1)(O)
VALIDATION TAG TO NEXT RECORD
445(1)(P)
CHECK VALUE
445(1)(Q)
SHIPPING RECORD 445(1)
SHIPPING RECORD N
445(1)(R)

FIG. 28

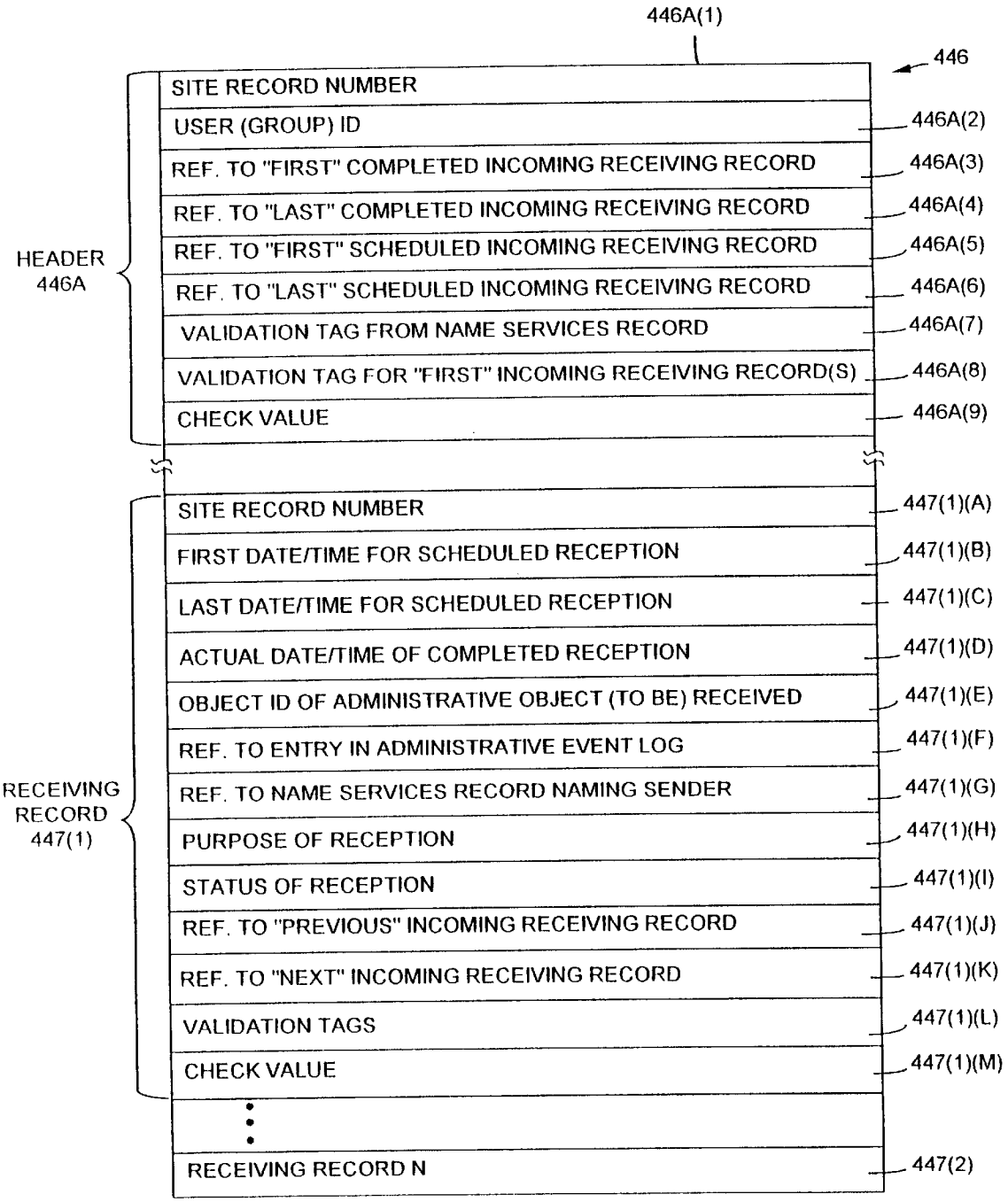
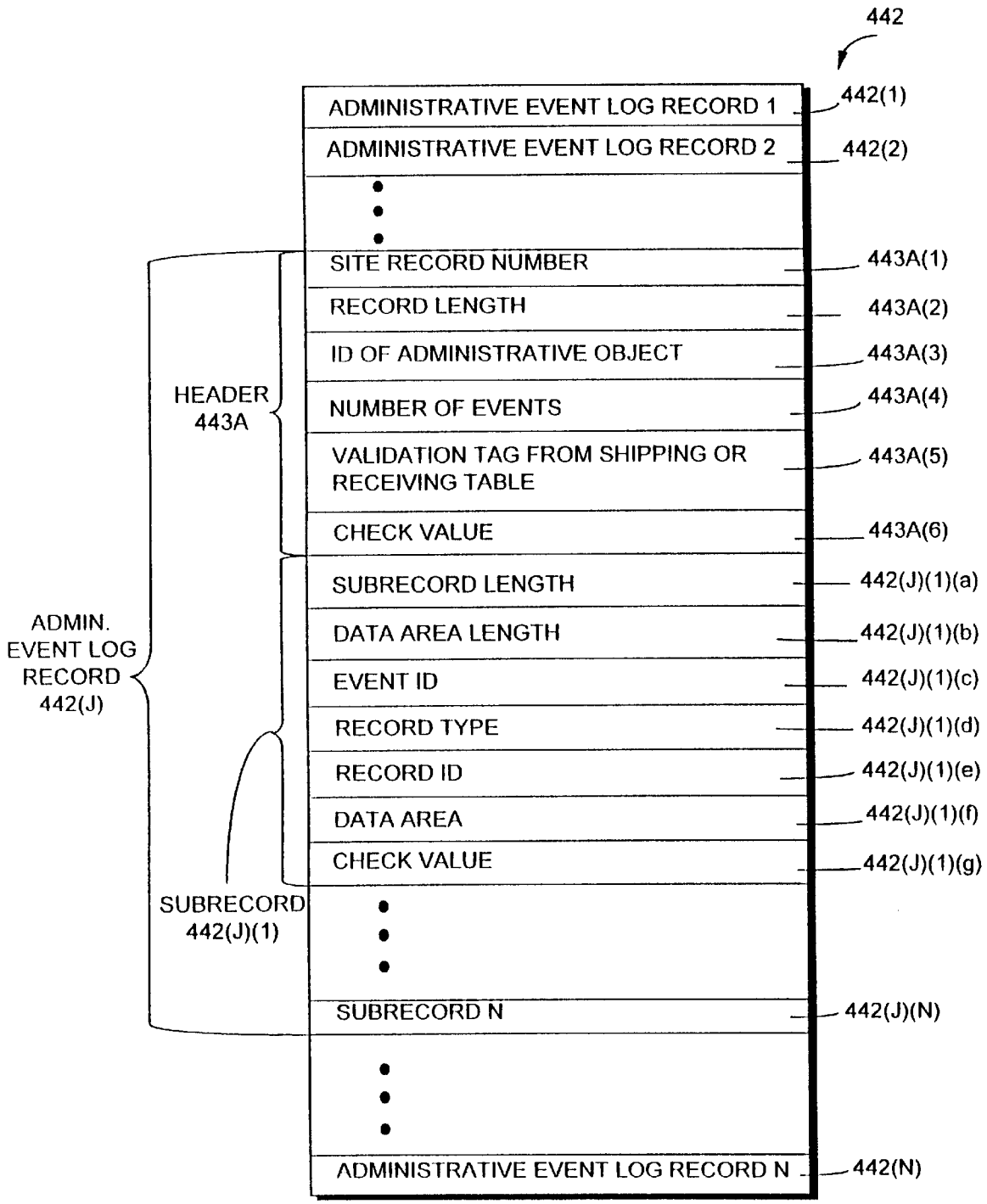

446A(1)
446
SITE RECORD NUMBER
USER (GROUP) ID
446A(2)
REF. TO "FIRST" COMPLETED INCOMING RECEIVING RECORD
446A(3)
REF. TO "LAST" COMPLETED INCOMING RECEIVING RECORD
446A(4)
REF. TO "FIRST" SCHEDULED INCOMING RECEIVING RECORD
446A(5)
REF. TO "LAST" SCHEDULED INCOMING RECEIVING RECORD
446A(6)
VALIDATION TAG FROM NAME SERVICES RECORD
446A(7)
VALIDATION TAG FOR "FIRST" INCOMING RECEIVING RECORD(S)
446A(8)
CHECK VALUE
446A(9)
HEADER 446A
SITE RECORD NUMBER
447(1)(A)
FIRST DATE/TIME FOR SCHEDULED RECEPTION
447(1)(B)
LAST DATE/TIME FOR SCHEDULED RECEPTION
447(1)(C)
ACTUAL DATE/TIME OF COMPLETED RECEPTION
447(1)(D)
OBJECT ID OF ADMINISTRATIVE OBJECT (TO BE) RECEIVED
447(1)(E)
REF. TO ENTRY IN ADMINISTRATIVE EVENT LOG
447(1)(F)
REF. TO NAME SERVICES RECORD NAMING SENDER
447(1)(G)
PURPOSE OF RECEPTION
447(1)(H)
STATUS OF RECEPTION
447(1)(I)
REF. TO "PREVIOUS" INCOMING RECEIVING RECORD
447(1)(J)
REF. TO "NEXT" INCOMING RECEIVING RECORD
447(1)(K)
VALIDATION TAGS
447(1)(L)
CHECK VALUE
447(1)(M)
RECEIVING RECORD 447(1)
RECEIVING RECORD N
447(2)

FIG. 29


442
ADMINISTRATIVE EVENT LOG RECORD 1
442(1)
ADMINISTRATIVE EVENT LOG RECORD 2
442(2)
SITE RECORD NUMBER
443A(1)
RECORD LENGTH
443A(2)
ID OF ADMINISTRATIVE OBJECT
443A(3)
NUMBER OF EVENTS
443A(4)
VALIDATION TAG FROM SHIPPING OR RECEIVING TABLE
443A(5)
CHECK VALUE
443A(6)
HEADER 443A
SUBRECORD LENGTH
442(J)(1)(a)
DATA AREA LENGTH
442(J)(1)(b)
EVENT ID
442(J)(1)(c)
RECORD TYPE
442(J)(1)(d)
RECORD ID
442(J)(1)(e)
DATA AREA
442(J)(1)(f)
CHECK VALUE
442(J)(1)(g)
ADMIN. EVENT LOG RECORD 442(J)
SUBRECORD 442(J)(1)
SUBRECORD N
442(J)(N)
ADMINISTRATIVE EVENT LOG RECORD N
442(N)

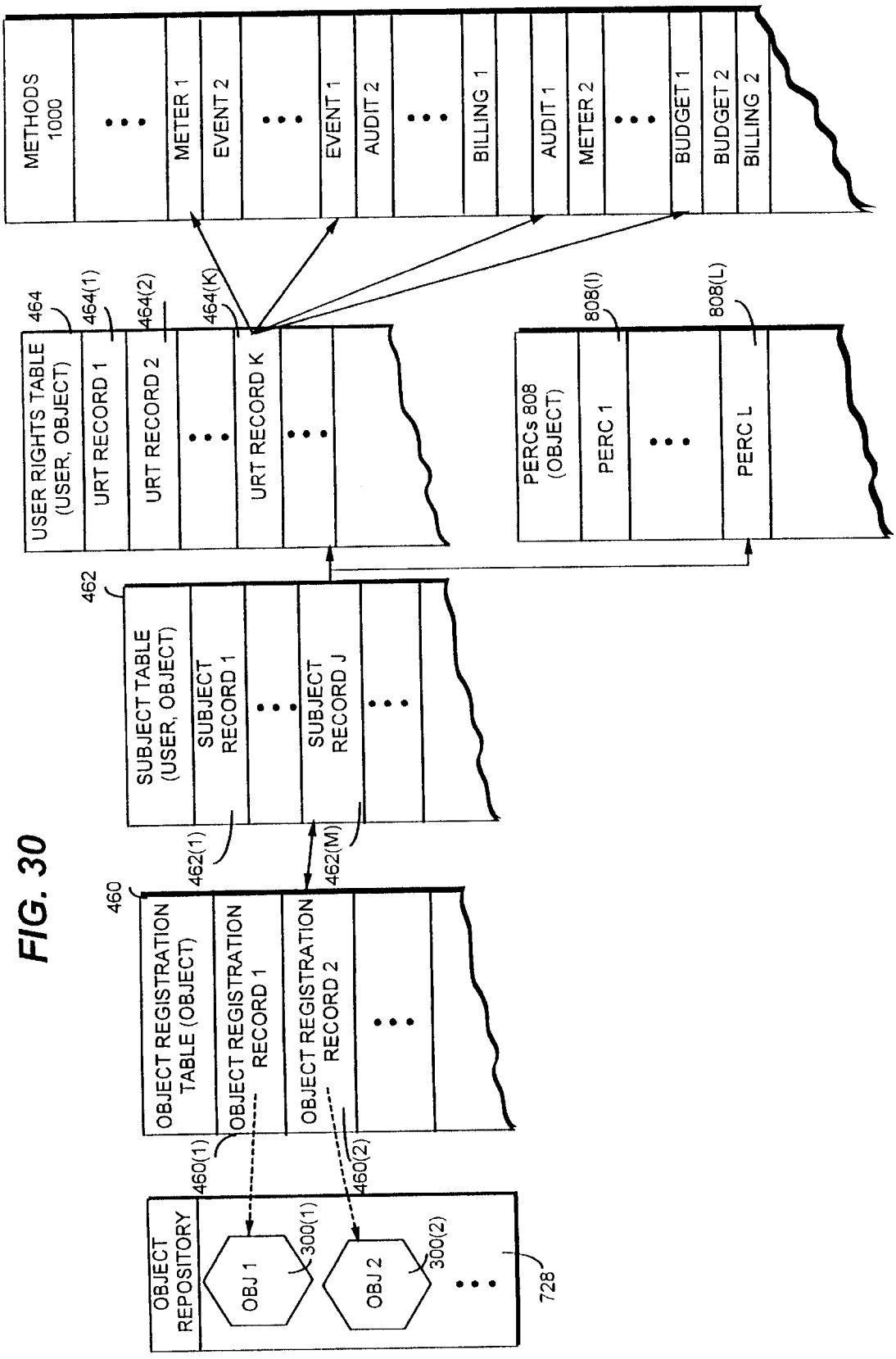

METHODS 1000
METER 1
EVENT 2
EVENT 1
AUDIT 2
BILLING 1
AUDIT 1
METER 2
BUDGET 1
BUDGET 2
BILLING 2
464
464(1)
464(2)
464(K)
USER RIGHTS TABLE (USER, OBJECT)
URT RECORD 1
URT RECORD 2
URT RECORD K
808(I)
808(L)
PERCs 808 (OBJECT)
PERC 1
PERC L
462
SUBJECT TABLE (USER, OBJECT)
SUBJECT RECORD 1
SUBJECT RECORD J
462(1)
462(M)
460
OBJECT REGISTRATION TABLE (OBJECT)
OBJECT REGISTRATION RECORD 1
OBJECT REGISTRATION RECORD 2
460(1)
460(2)
OBJECT REPOSITORY
OBJ 1
OBJ 2
300(1)
300(2)
728

FIG. 30

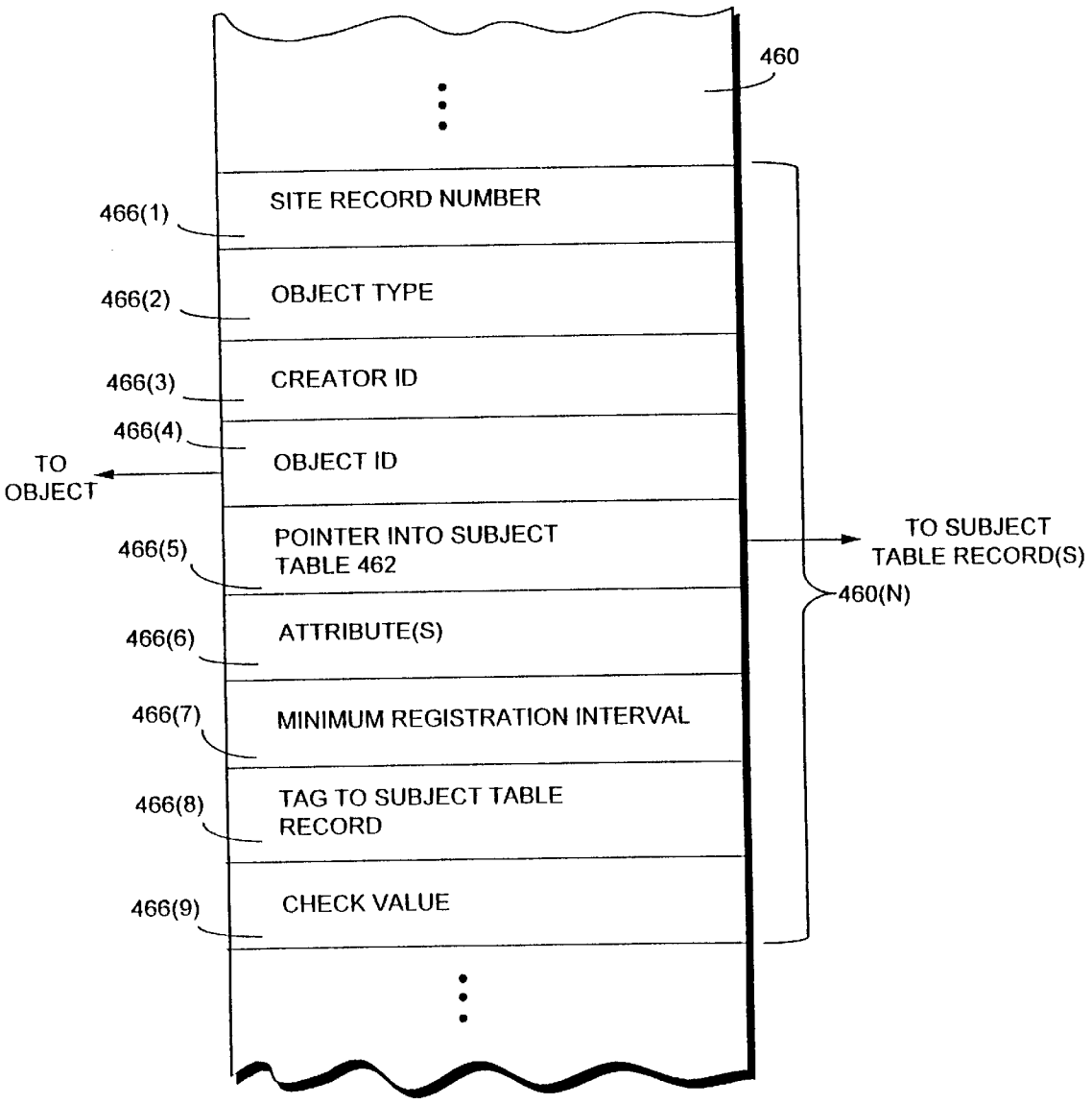

460
466(1)
SITE RECORD NUMBER
466(2)
OBJECT TYPE
466(3)
CREATOR ID
466(4)
TO OBJECT
OBJECT ID
466(5)
POINTER INTO SUBJECT TABLE 462
TO SUBJECT TABLE RECORD(S)
460(N)
466(6)
ATTRIBUTE(S)
466(7)
MINIMUM REGISTRATION INTERVAL
466(8)
TAG TO SUBJECT TABLE RECORD
466(9)
CHECK VALUE

FIG. 31

FIG. 32

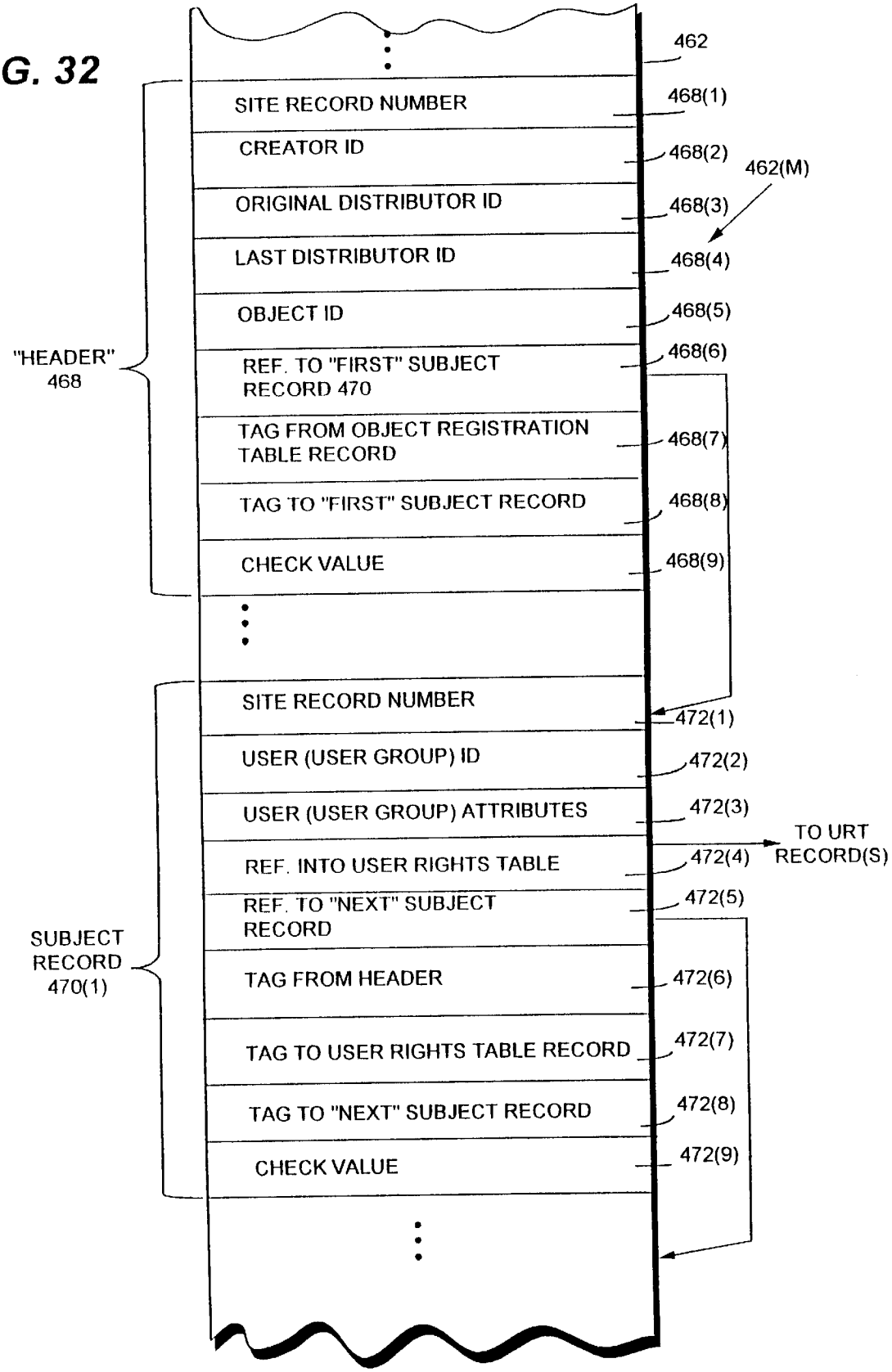

462
SITE RECORD NUMBER
468(1)
CREATOR ID
468(2)
462(M)
ORIGINAL DISTRIBUTOR ID
468(3)
LAST DISTRIBUTOR ID
468(4)
OBJECT ID
468(5)
"HEADER" 468
REF. TO "FIRST" SUBJECT RECORD 470
468(6)
TAG FROM OBJECT REGISTRATION TABLE RECORD
468(7)
TAG TO "FIRST" SUBJECT RECORD
468(8)
CHECK VALUE
468(9)
SITE RECORD NUMBER
472(1)
USER (USER GROUP) ID
472(2)
USER (USER GROUP) ATTRIBUTES
472(3)
REF. INTO USER RIGHTS TABLE
472(4)
TO URT RECORD(S)
REF. TO "NEXT" SUBJECT RECORD
472(5)
SUBJECT RECORD 470(1)
TAG FROM HEADER
472(6)
TAG TO USER RIGHTS TABLE RECORD
472(7)
TAG TO "NEXT" SUBJECT RECORD
472(8)
CHECK VALUE
472(9)

FIG. 33

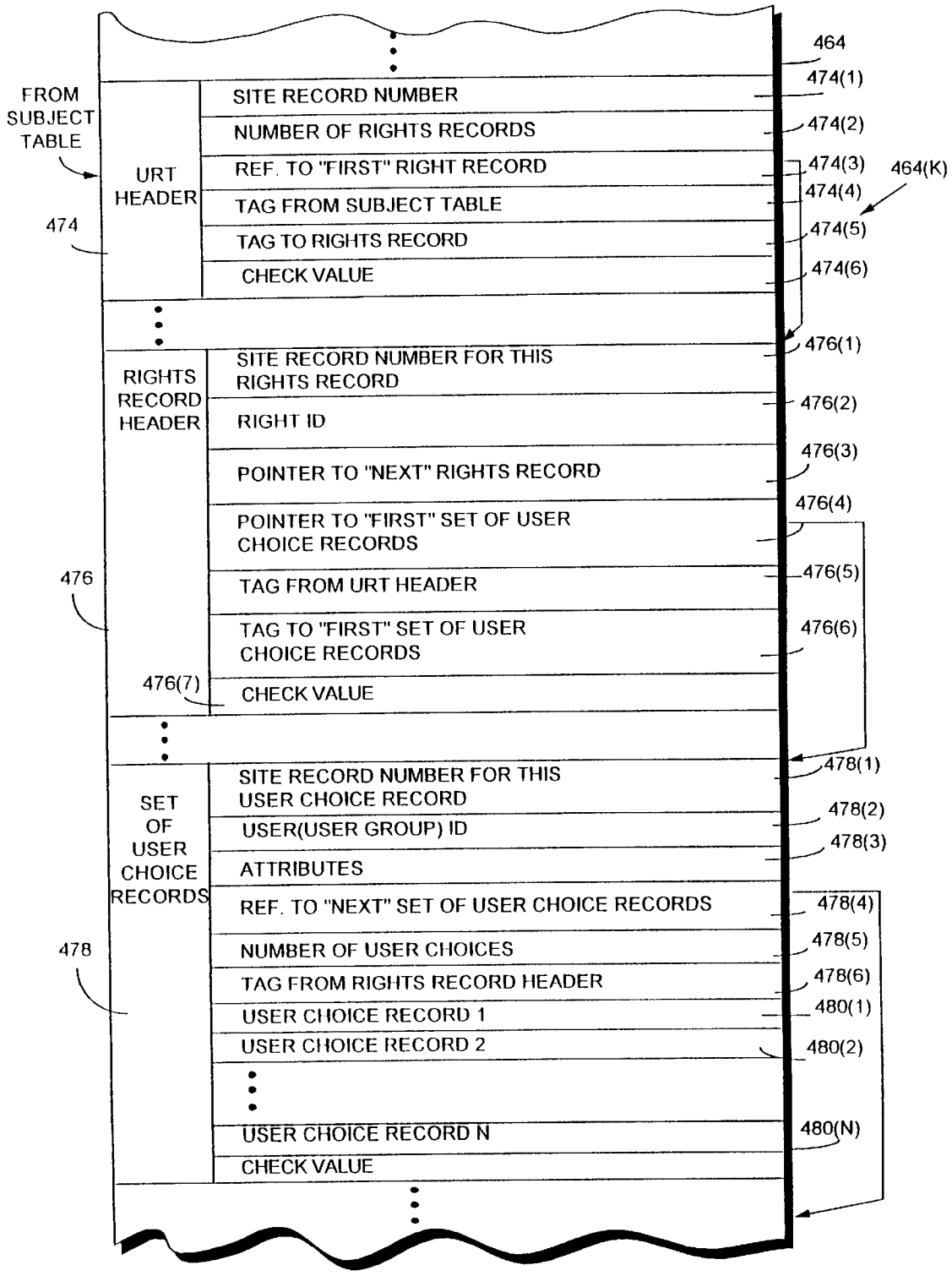

FROM SUBJECT TABLE
URT HEADER
474
SITE RECORD NUMBER
NUMBER OF RIGHTS RECORDS
REF. TO "FIRST" RIGHT RECORD
TAG FROM SUBJECT TABLE
TAG TO RIGHTS RECORD
CHECK VALUE
RIGHTS RECORD HEADER
476
SITE RECORD NUMBER FOR THIS RIGHTS RECORD
RIGHT ID
POINTER TO "NEXT" RIGHTS RECORD
POINTER TO "FIRST" SET OF USER CHOICE RECORDS
TAG FROM URT HEADER
TAG TO "FIRST" SET OF USER CHOICE RECORDS
476(7)
CHECK VALUE
SET OF USER CHOICE RECORDS
478
SITE RECORD NUMBER FOR THIS USER CHOICE RECORD
USER(USER GROUP) ID
ATTRIBUTES
REF. TO "NEXT" SET OF USER CHOICE RECORDS
NUMBER OF USER CHOICES
TAG FROM RIGHTS RECORD HEADER
USER CHOICE RECORD 1
USER CHOICE RECORD 2
USER CHOICE RECORD N
CHECK VALUE
464
474(1)
474(2)
474(3)
464(K)
474(4)
474(5)
474(6)
476(1)
476(2)
476(3)
476(4)
476(5)
476(6)
478(1)
478(2)
478(3)
478(4)
478(5)
478(6)
480(1)
480(2)
480(N)

FIG. 34

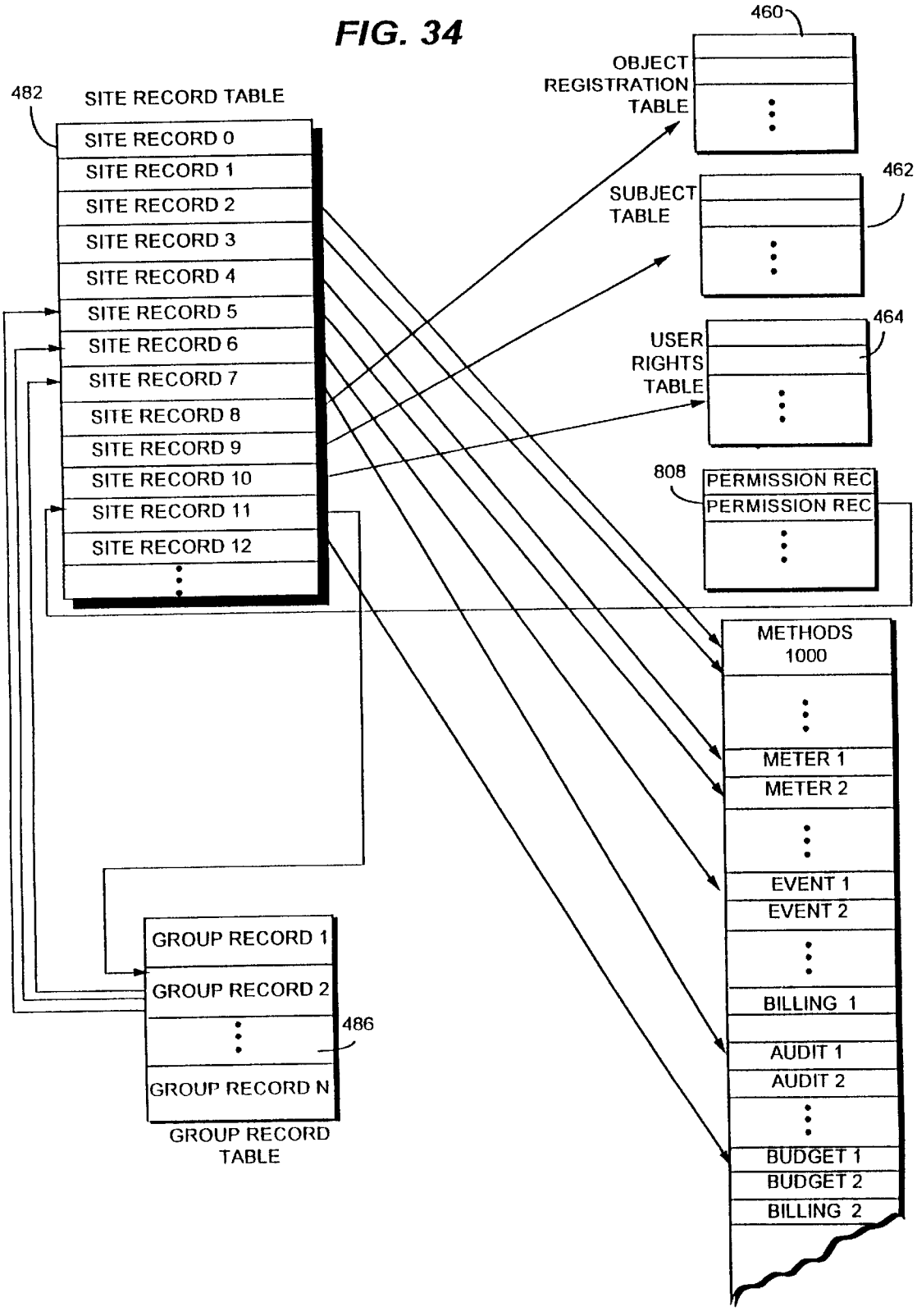

460
OBJECT REGISTRATION TABLE
482
SITE RECORD TABLE
SITE RECORD 0
SITE RECORD 1
SITE RECORD 2
SITE RECORD 3
SITE RECORD 4
SITE RECORD 5
SITE RECORD 6
SITE RECORD 7
SITE RECORD 8
SITE RECORD 9
SITE RECORD 10
SITE RECORD 11
SITE RECORD 12
462
SUBJECT TABLE
464
USER RIGHTS TABLE
808
PERMISSION REC
PERMISSION REC
METHODS 1000
METER 1
METER 2
EVENT 1
EVENT 2
BILLING 1
AUDIT 1
AUDIT 2
BUDGET 1
BUDGET 2
BILLING 2
GROUP RECORD 1
GROUP RECORD 2
GROUP RECORD N
486
GROUP RECORD TABLE

FIG. 34A

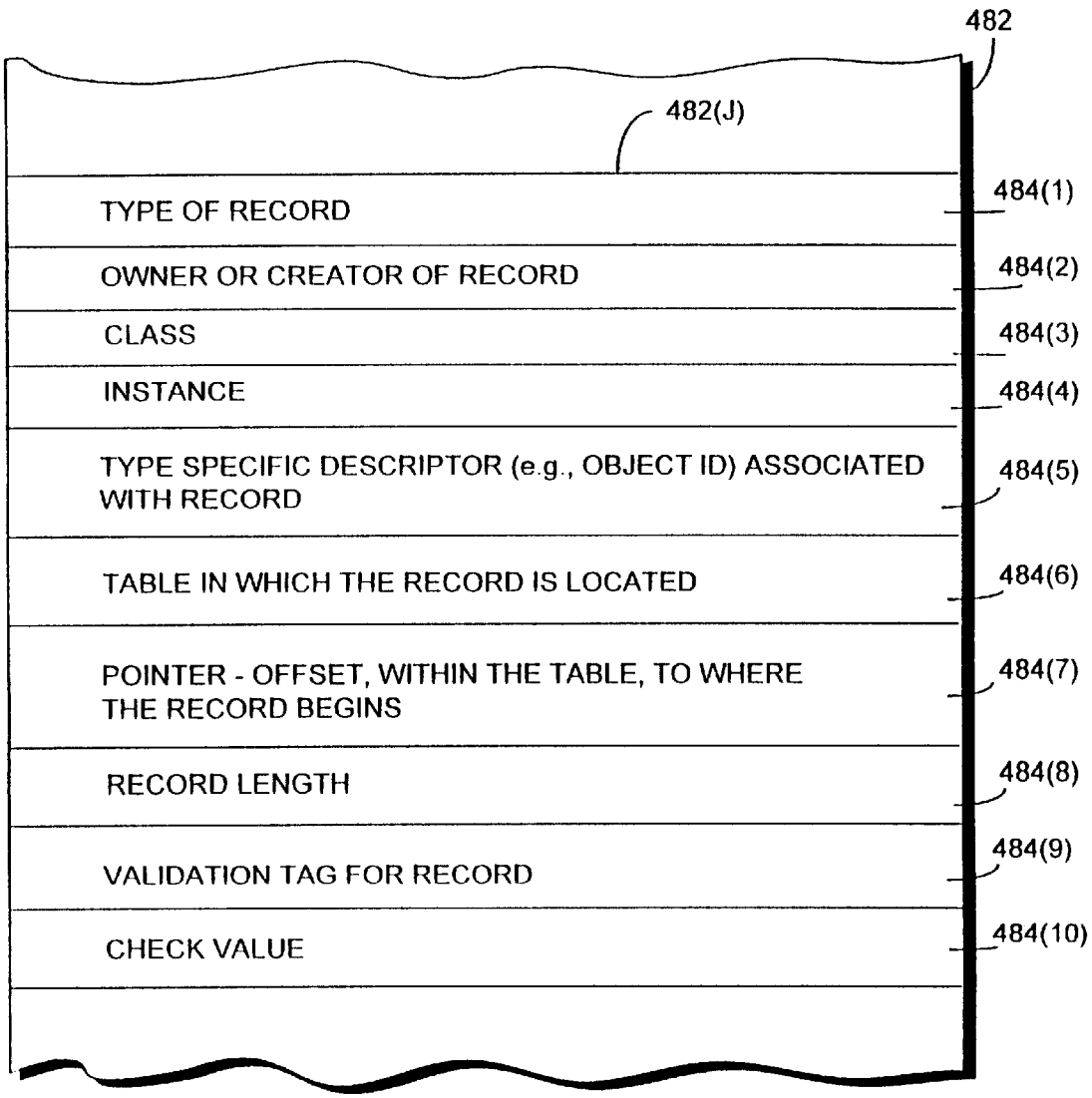

482
482(J)
TYPE OF RECORD
484(1)
OWNER OR CREATOR OF RECORD
484(2)
CLASS
484(3)
INSTANCE
484(4)
TYPE SPECIFIC DESCRIPTOR (e.g., OBJECT ID) ASSOCIATED WITH RECORD
484(5)
TABLE IN WHICH THE RECORD IS LOCATED
484(6)
POINTER - OFFSET, WITHIN THE TABLE, TO WHERE THE RECORD BEGINS
484(7)
RECORD LENGTH
484(8)
VALIDATION TAG FOR RECORD
484(9)
CHECK VALUE
484(10)

FIG. 34B

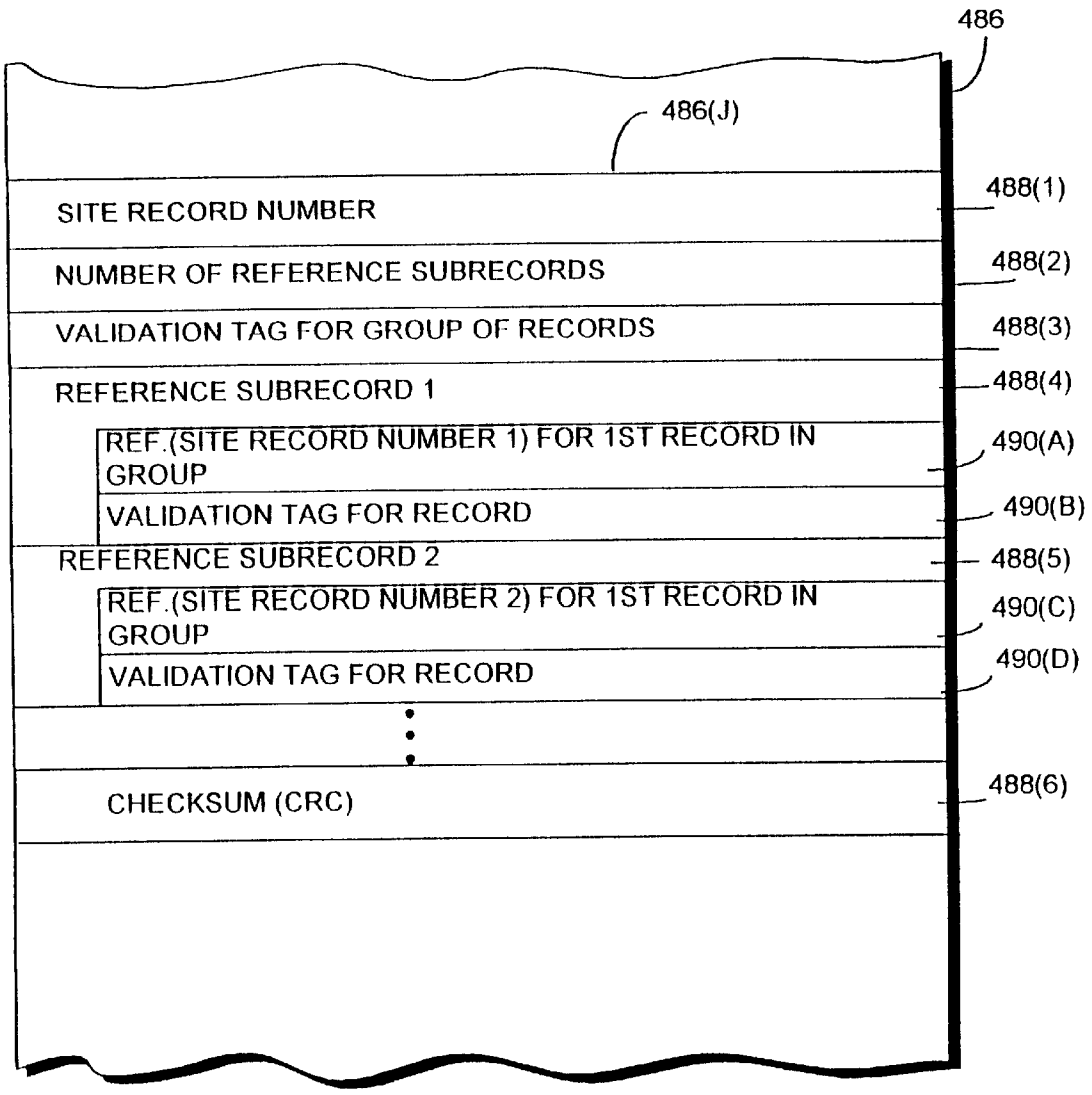

486
486(J)
SITE RECORD NUMBER
488(1)
NUMBER OF REFERENCE SUBRECORDS
488(2)
VALIDATION TAG FOR GROUP OF RECORDS
488(3)
REFERENCE SUBRECORD 1
488(4)
REF.(SITE RECORD NUMBER 1) FOR 1ST RECORD IN GROUP
490(A)
VALIDATION TAG FOR RECORD
490(B)
REFERENCE SUBRECORD 2
488(5)
REF.(SITE RECORD NUMBER 2) FOR 1ST RECORD IN GROUP
490(C)
VALIDATION TAG FOR RECORD
490(D)
CHECKSUM (CRC)
488(6)

FIG. 35

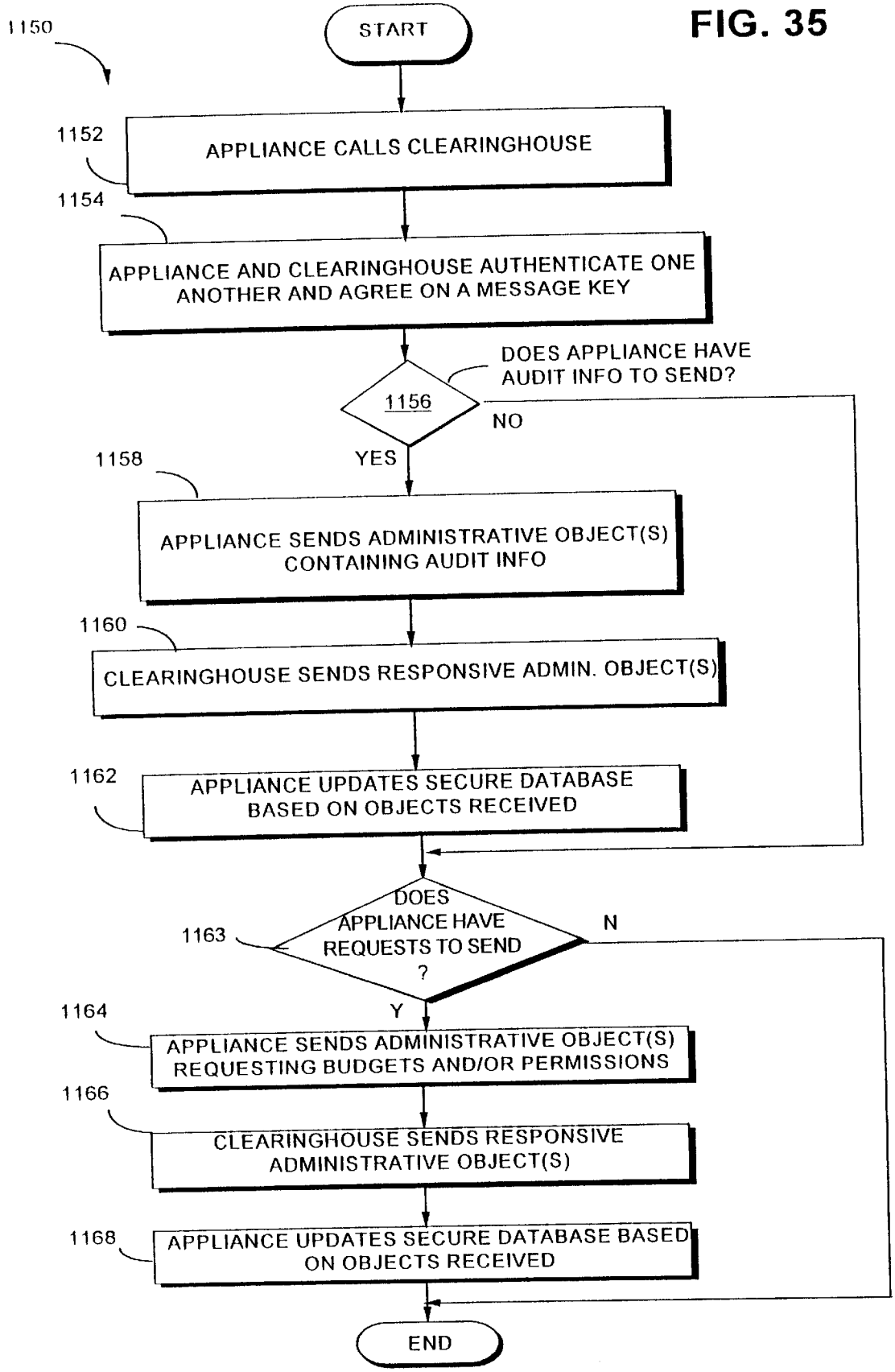
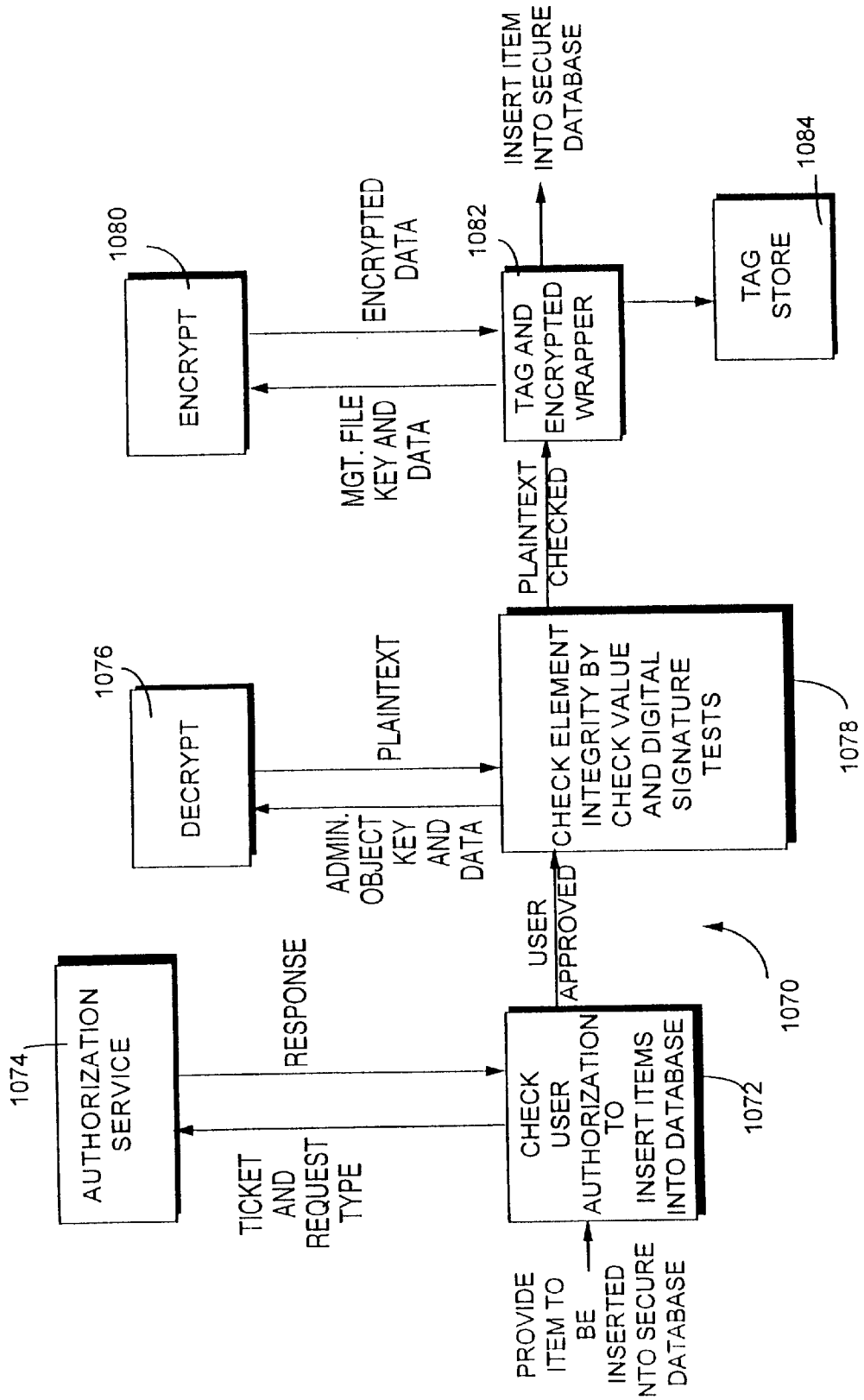

1150
START
1152
APPLIANCE CALLS CLEARINGHOUSE
1154
APPLIANCE AND CLEARINGHOUSE AUTHENTICATE ONE ANOTHER AND AGREE ON A MESSAGE KEY
DOES APPLIANCE HAVE AUDIT INFO TO SEND?
1156
NO
YES
1158
APPLIANCE SENDS ADMINISTRATIVE OBJECT(S) CONTAINING AUDIT INFO
1160
CLEARINGHOUSE SENDS RESPONSIVE ADMIN. OBJECT(S)
1162
APPLIANCE UPDATES SECURE DATABASE BASED ON OBJECTS RECEIVED
DOES APPLIANCE HAVE REQUESTS TO SEND ?
1163
N
Y
1164
APPLIANCE SENDS ADMINISTRATIVE OBJECT(S) REQUESTING BUDGETS AND/OR PERMISSIONS
1166
CLEARINGHOUSE SENDS RESPONSIVE ADMINISTRATIVE OBJECT(S)
1168
APPLIANCE UPDATES SECURE DATABASE BASED ON OBJECTS RECEIVED
END

FIG. 36


PROVIDE ITEM TO BE INSERTED INTO SECURE DATABASE
CHECK USER AUTHORIZATION TO INSERT ITEMS INTO DATABASE
1072
1070
TICKET AND REQUEST TYPE
RESPONSE
AUTHORIZATION SERVICE
1074
USER APPROVED
ADMIN. OBJECT KEY AND DATA
PLAINTEXT
DECRYPT
1076
CHECK ELEMENT INTEGRITY BY CHECK VALUE AND DIGITAL SIGNATURE TESTS
1078
PLAINTEXT CHECKED
MGT. FILE KEY AND DATA
ENCRYPTED DATA
ENCRYPT
1080
TAG AND ENCRYPTED WRAPPER
1082
INSERT ITEM INTO SECURE DATABASE
TAG STORE
1084

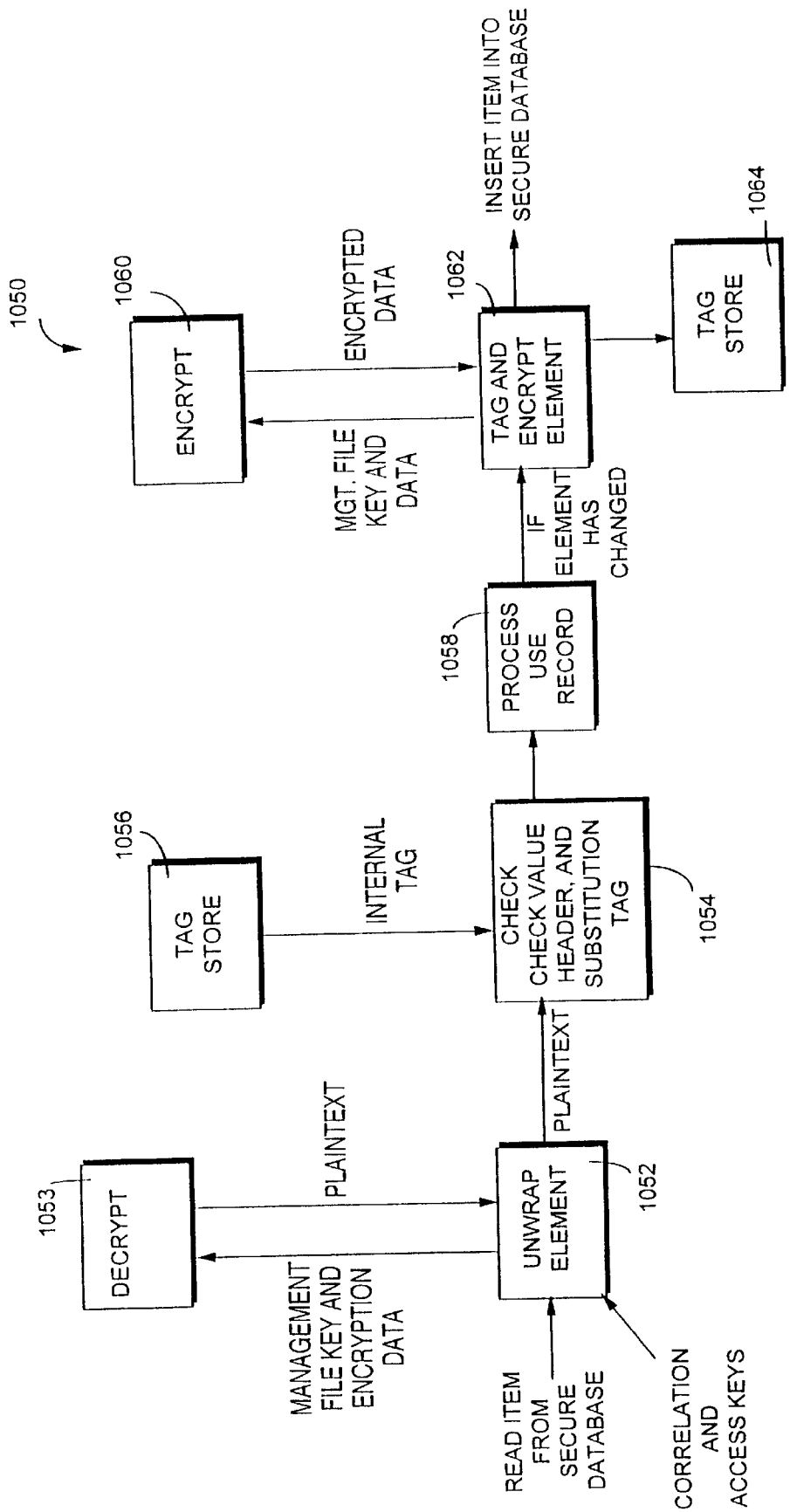

1050
1060
ENCRYPT
ENCRYPTED DATA
MGT. FILE KEY AND DATA
1062
TAG AND ENCRYPT ELEMENT
INSERT ITEM INTO SECURE DATABASE
1064
TAG STORE
IF ELEMENT HAS CHANGED
1058
PROCESS USE RECORD
1056
TAG STORE
INTERNAL TAG
CHECK CHECK VALUE HEADER, AND SUBSTITUTION TAG
1054
1053
DECRYPT
PLAINTEXT
UNWRAP ELEMENT
1052
PLAINTEXT
MANAGEMENT FILE KEY AND ENCRYPTION DATA
READ ITEM FROM SECURE DATABASE
CORRELATION AND ACCESS KEYS

FIG. 37

FIG. 38

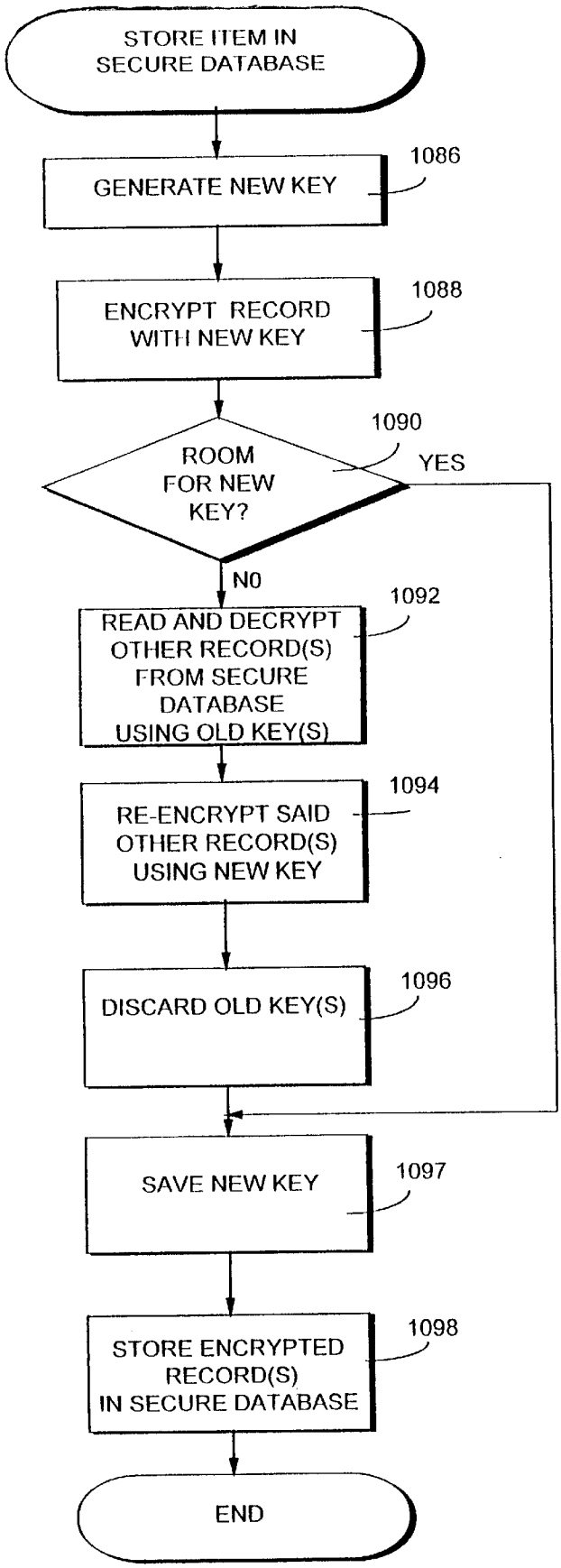

STORE ITEM IN SECURE DATABASE
GENERATE NEW KEY
1086
ENCRYPT RECORD WITH NEW KEY
1088
ROOM FOR NEW KEY?
1090
YES
NO
READ AND DECRYPT OTHER RECORD(S) FROM SECURE DATABASE USING OLD KEY(S)
1092
RE-ENCRYPT SAID OTHER RECORD(S) USING NEW KEY
1094
DISCARD OLD KEY(S)
1096
SAVE NEW KEY
1097
STORE ENCRYPTED RECORD(S) IN SECURE DATABASE
1098
END

FIG. 39

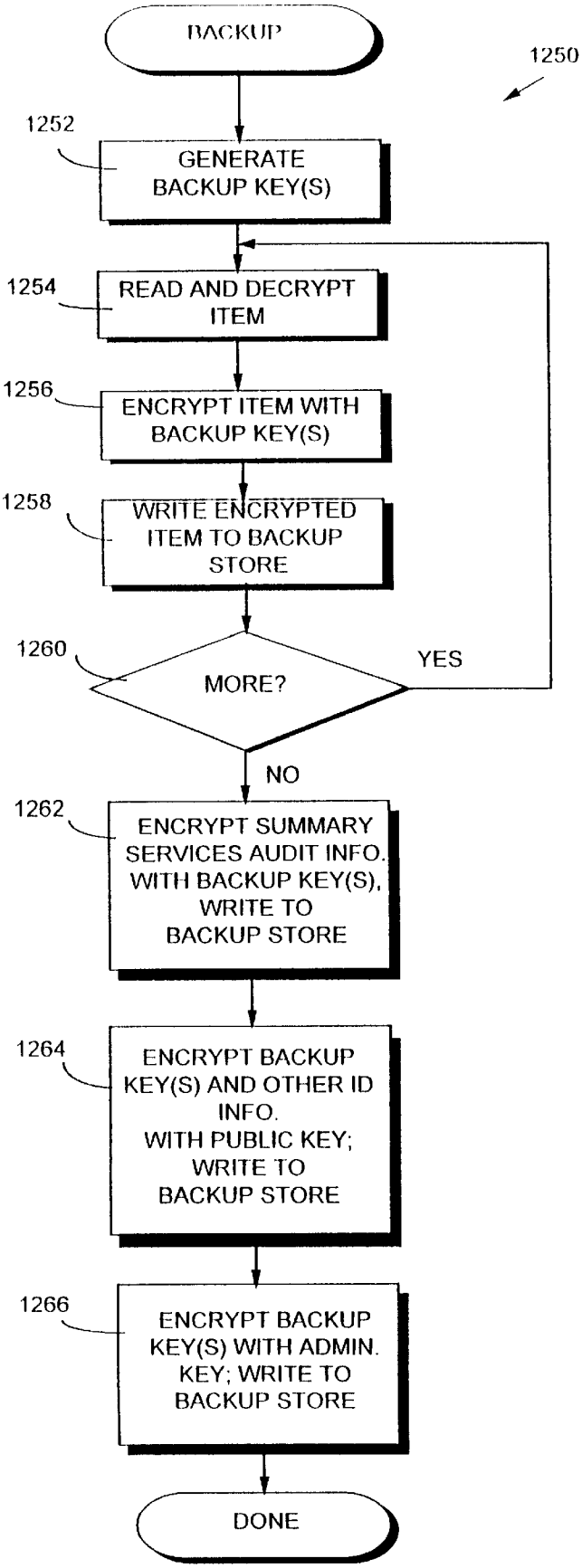
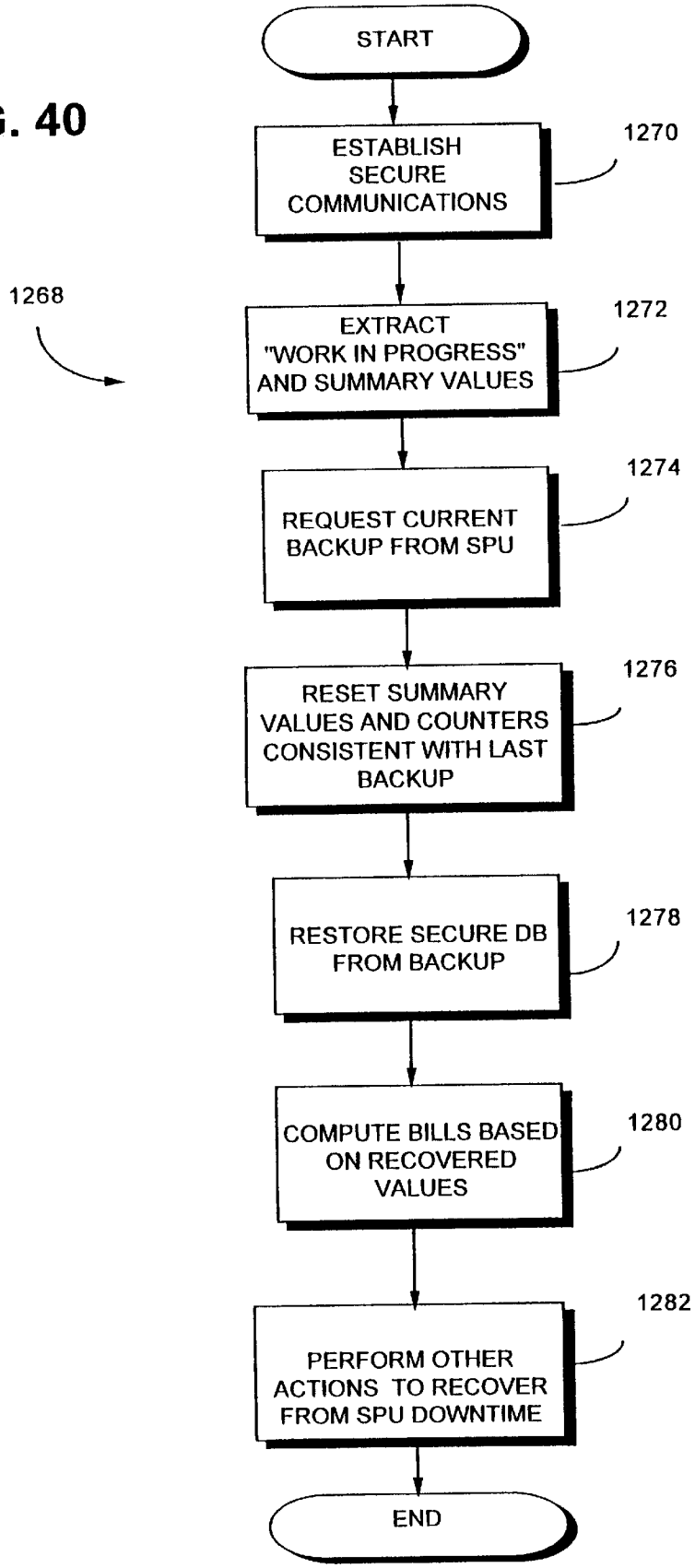

BACKUP
1250
1252
GENERATE BACKUP KEY(S)
1254
READ AND DECRYPT ITEM
1256
ENCRYPT ITEM WITH BACKUP KEY(S)
1258
WRITE ENCRYPTED ITEM TO BACKUP STORE
1260
MORE?
YES
NO
1262
ENCRYPT SUMMARY SERVICES AUDIT INFO. WITH BACKUP KEY(S), WRITE TO BACKUP STORE
1264
ENCRYPT BACKUP KEY(S) AND OTHER ID INFO. WITH PUBLIC KEY; WRITE TO BACKUP STORE
1266
ENCRYPT BACKUP KEY(S) WITH ADMIN. KEY; WRITE TO BACKUP STORE
DONE

FIG. 40


START
ESTABLISH SECURE COMMUNICATIONS
1270
1268
EXTRACT "WORK IN PROGRESS" AND SUMMARY VALUES
1272
REQUEST CURRENT BACKUP FROM SPU
1274
RESET SUMMARY VALUES AND COUNTERS CONSISTENT WITH LAST BACKUP
1276
RESTORE SECURE DB FROM BACKUP
1278
COMPUTE BILLS BASED ON RECOVERED VALUES
1280
PERFORM OTHER ACTIONS TO RECOVER FROM SPU DOWNTIME
1282
END

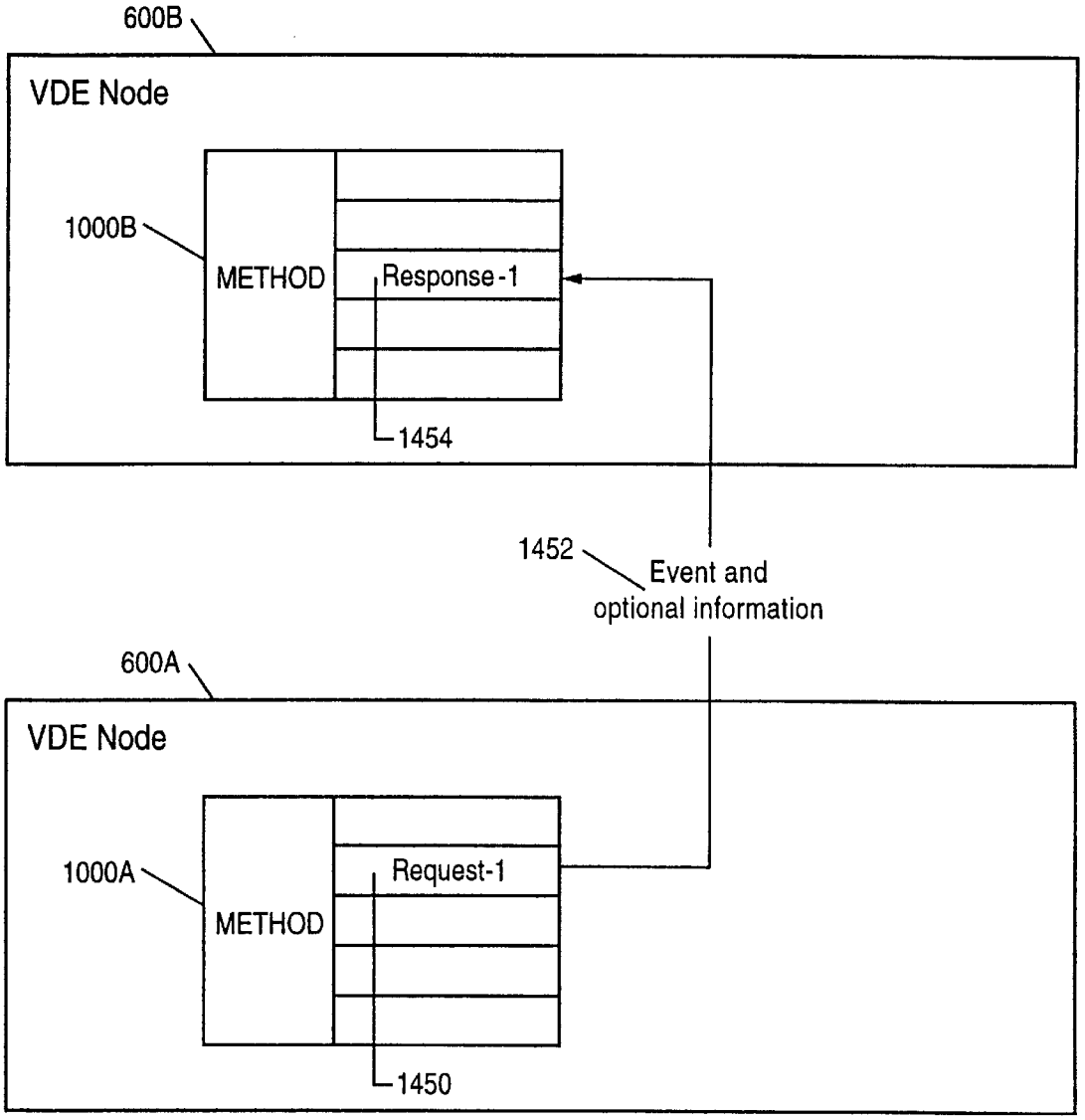

600B
VDE Node
1000B
METHOD
Response-1
1454
1452
Event and
optional information
600A
VDE Node
1000A
METHOD
Request-1
1450

FIG. 41a

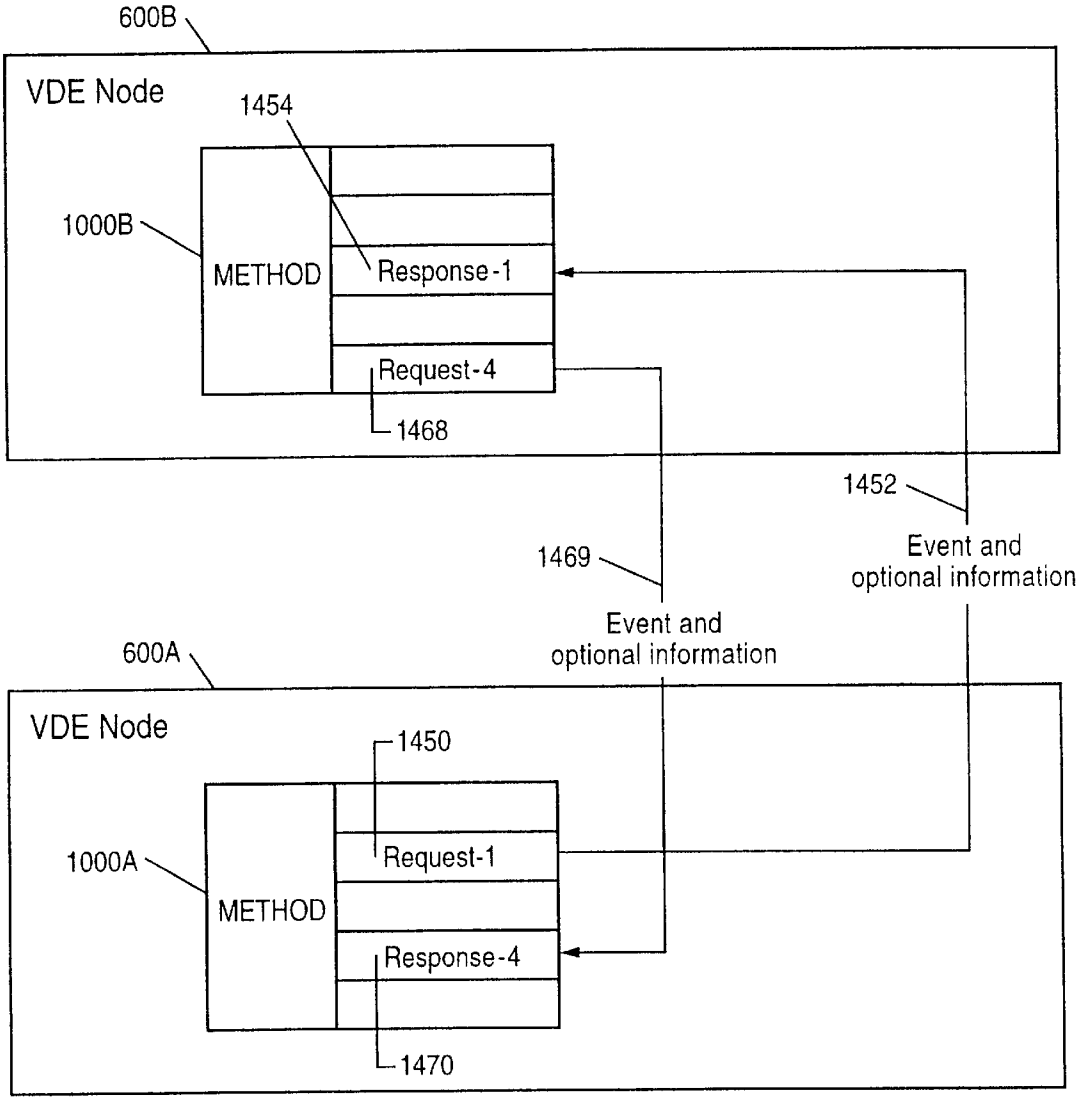

600B
VDE Node
1454
1000B
METHOD
Response-1
Request-4
1468
1452
Event and
optional information
1469
Event and
optional information
600A
VDE Node
1450
1000A
METHOD
Request-1
Response-4
1470

FIG. 41b

FIG. 41c

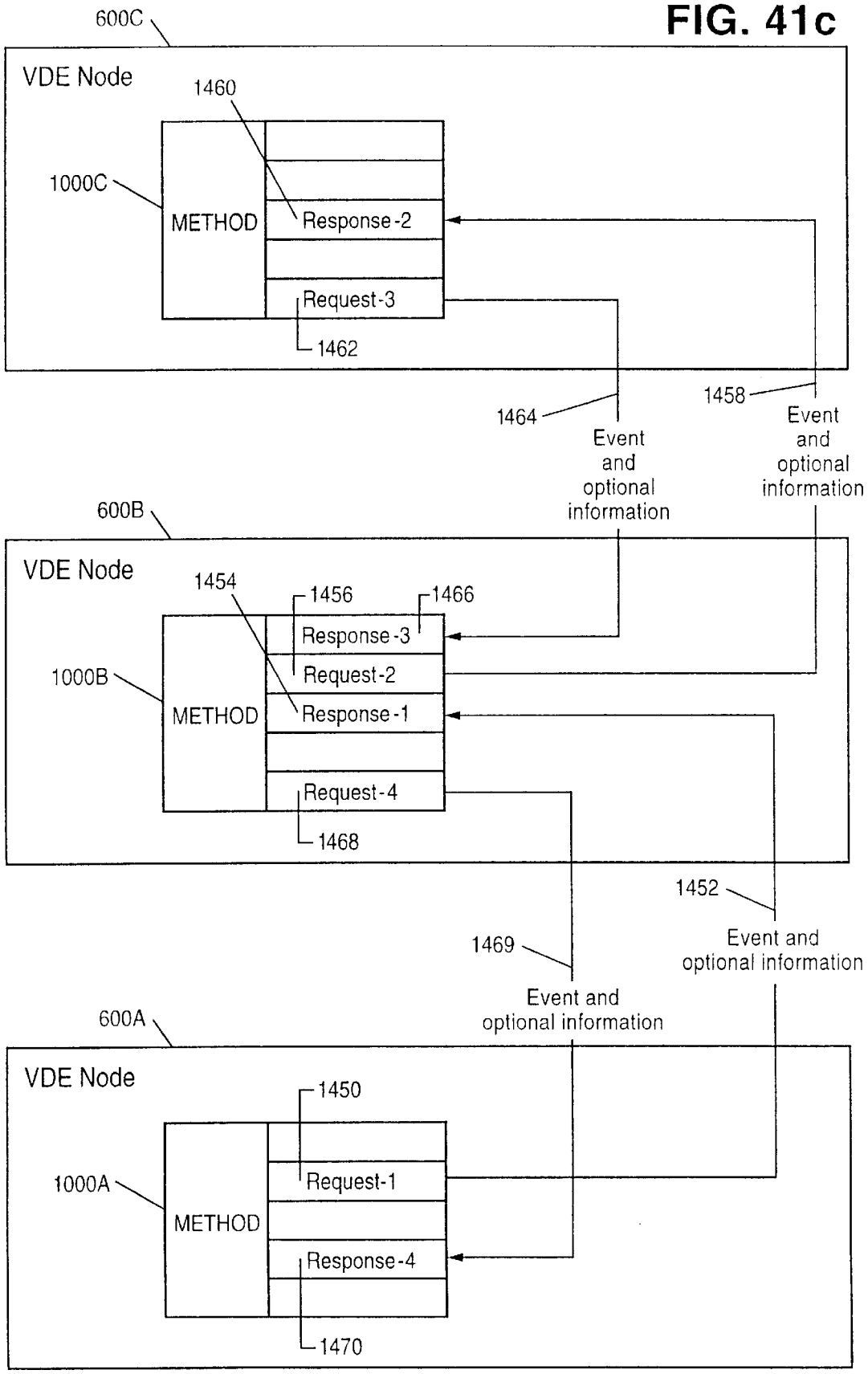
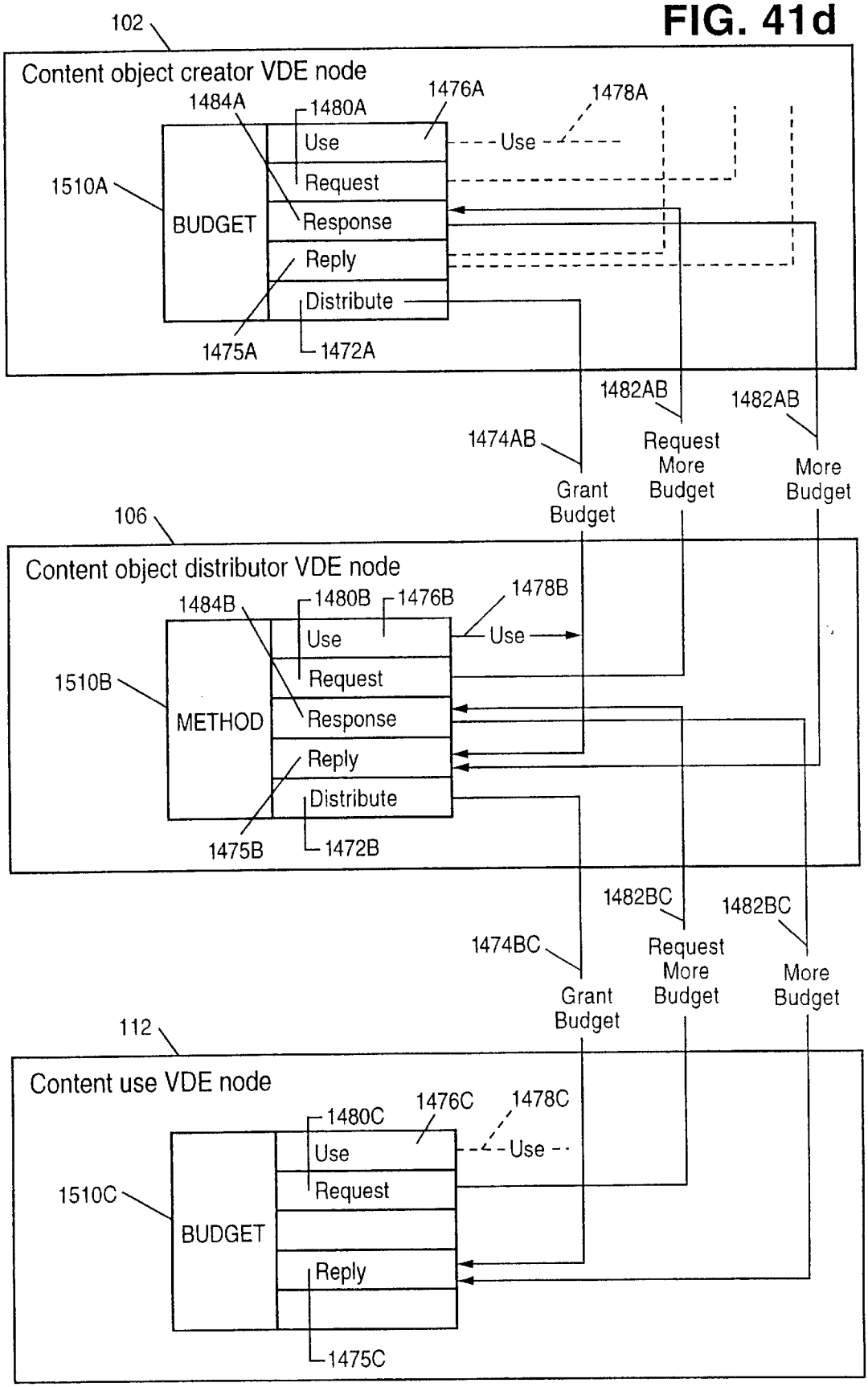

600C
VDE Node
1460
1000C
METHOD
Response-2
Request-3
1462
1464
Event and optional information
1458
Event and optional information
600B
VDE Node
1454
1456
1466
1000B
METHOD
Response-3
Request-2
Response-1
Request-4
1468
1452
Event and optional information
1469
Event and optional information
600A
VDE Node
1450
1000A
METHOD
Request-1
Response-4
1470

FIG. 41d


102
Content object creator VDE node
1484A
1480A
1476A
1478A
Use
Use
Request
1510A
BUDGET
Response
Reply
Distribute
1475A
1472A
1482AB
1482AB
1474AB
Request More Budget
More Budget
Grant Budget
106
Content object distributor VDE node
1484B
1480B
1476B
1478B
Use
Use
Request
1510B
METHOD
Response
Reply
Distribute
1475B
1472B
1482BC
1482BC
1474BC
Request More Budget
More Budget
Grant Budget
112
Content use VDE node
1480C
1476C
1478C
Use
Use
Request
1510C
BUDGET
Reply
1475C

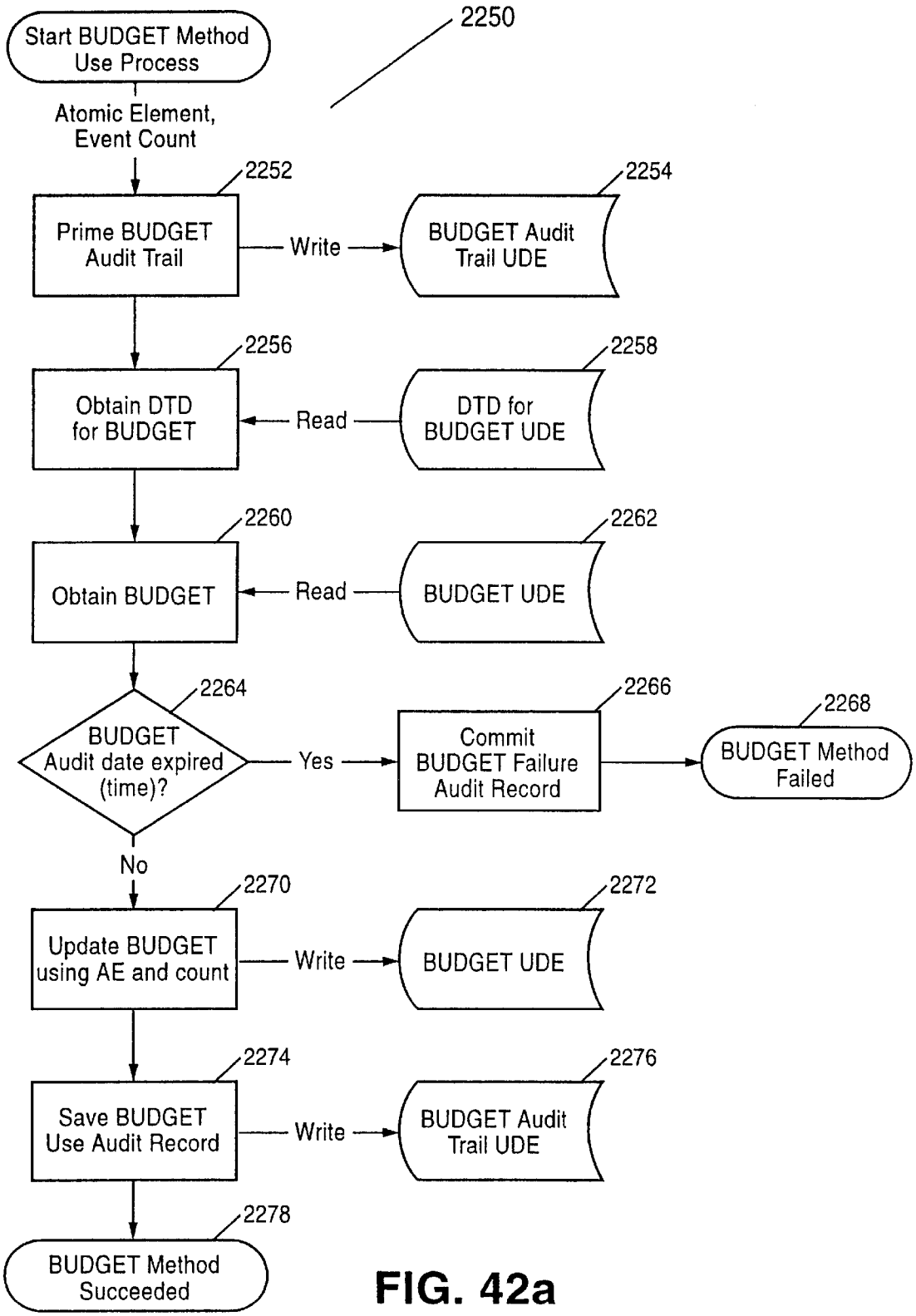

2250
Start BUDGET Method Use Process
Atomic Element, Event Count
2252
Prime BUDGET Audit Trail
Write
2254
BUDGET Audit Trail UDE
2256
Obtain DTD for BUDGET
Read
2258
DTD for BUDGET UDE
2260
Obtain BUDGET
Read
2262
BUDGET UDE
2264
BUDGET Audit date expired (time)?
Yes
2266
Commit BUDGET Failure Audit Record
2268
BUDGET Method Failed
No
2270
Update BUDGET using AE and count
Write
2272
BUDGET UDE
2274
Save BUDGET Use Audit Record
Write
2276
BUDGET Audit Trail UDE
2278
BUDGET Method Succeeded

FIG. 42a

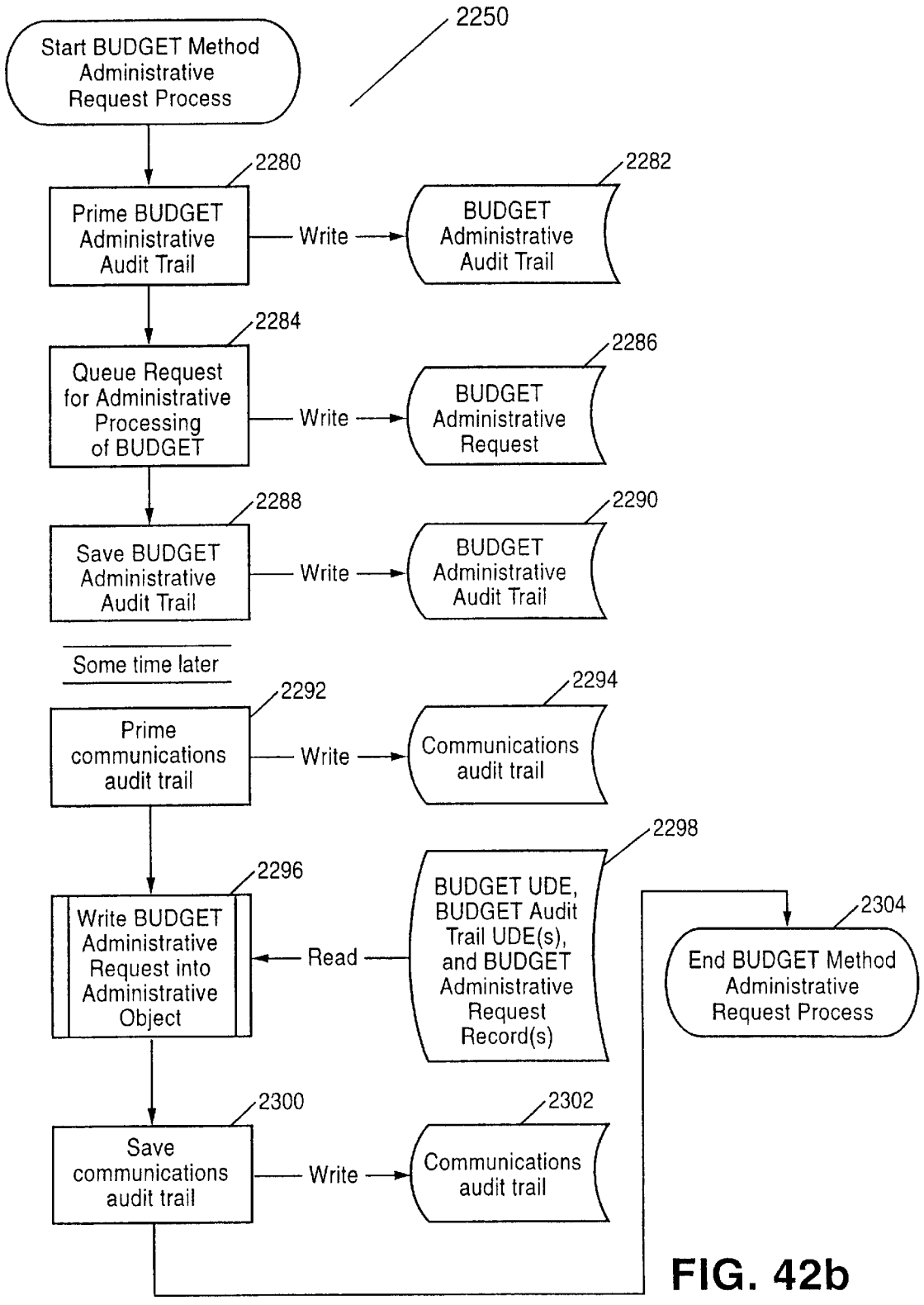

2250
Start BUDGET Method Administrative Request Process
2280
Prime BUDGET Administrative Audit Trail
Write
2282
BUDGET Administrative Audit Trail
2284
Queue Request for Administrative Processing of BUDGET
Write
2286
BUDGET Administrative Request
2288
Save BUDGET Administrative Audit Trail
Write
2290
BUDGET Administrative Audit Trail
Some time later
2292
Prime communications audit trail
Write
2294
Communications audit trail
2298
2296
Write BUDGET Administrative Request into Administrative Object
Read
BUDGET UDE, BUDGET Audit Trail UDE(s), and BUDGET Administrative Request Record(s)
2304
End BUDGET Method Administrative Request Process
2300
Save communications audit trail
Write
2302
Communications audit trail

FIG. 42b

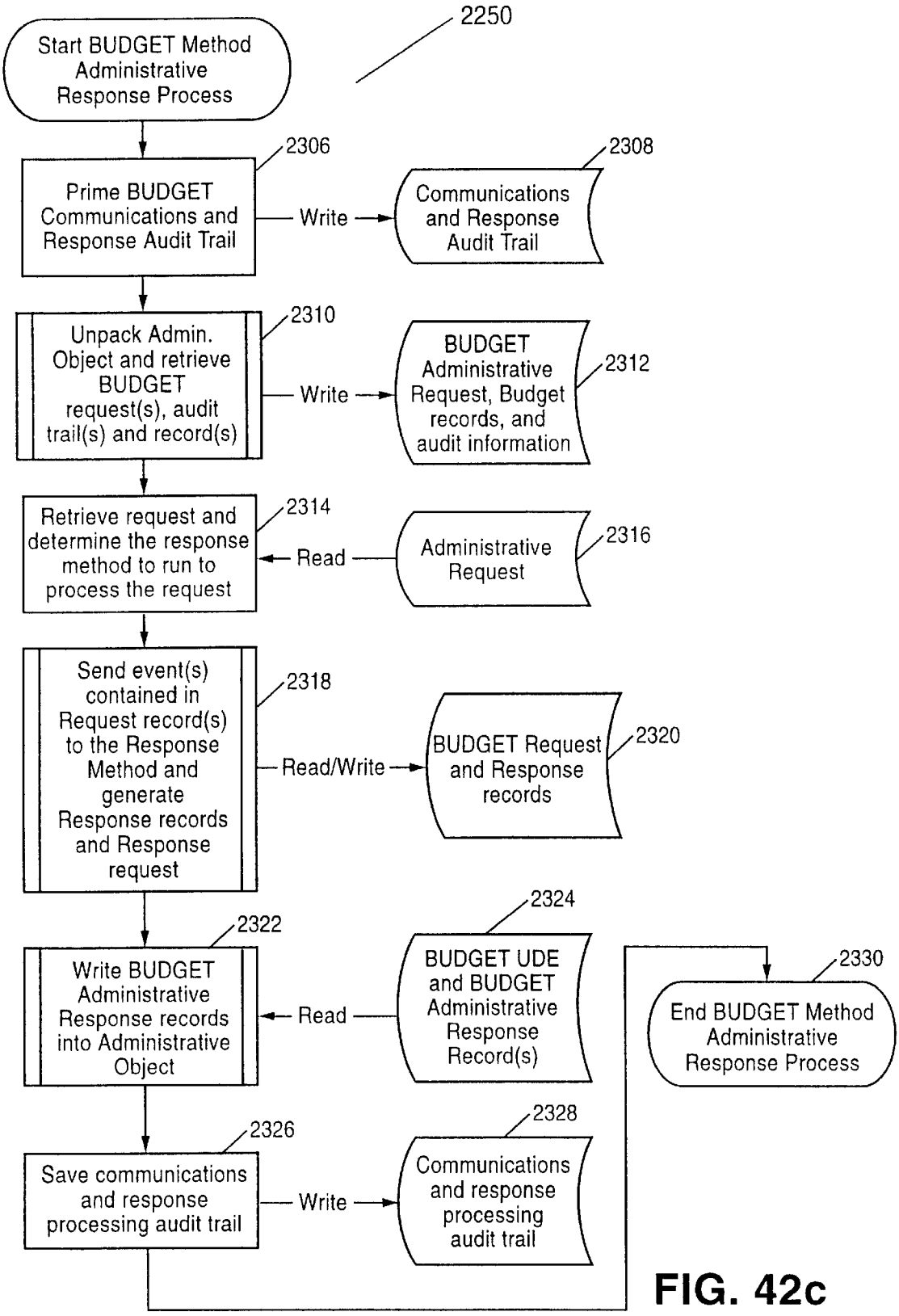

2250
Start BUDGET Method Administrative Response Process
2306
Prime BUDGET Communications and Response Audit Trail
Write
2308
Communications and Response Audit Trail
2310
Unpack Admin. Object and retrieve BUDGET request(s), audit trail(s) and record(s)
Write
BUDGET Administrative Request, Budget records, and audit information
2312
2314
Retrieve request and determine the response method to run to process the request
Read
Administrative Request
2316
2318
Send event(s) contained in Request record(s) to the Response Method and generate Response records and Response request
Read/Write
BUDGET Request and Response records
2320
2322
Write BUDGET Administrative Response records into Administrative Object
Read
2324
BUDGET UDE and BUDGET Administrative Response Record(s)
2326
Save communications and response processing audit trail
Write
2328
Communications and response processing audit trail
2330
End BUDGET Method Administrative Response Process

FIG. 42c

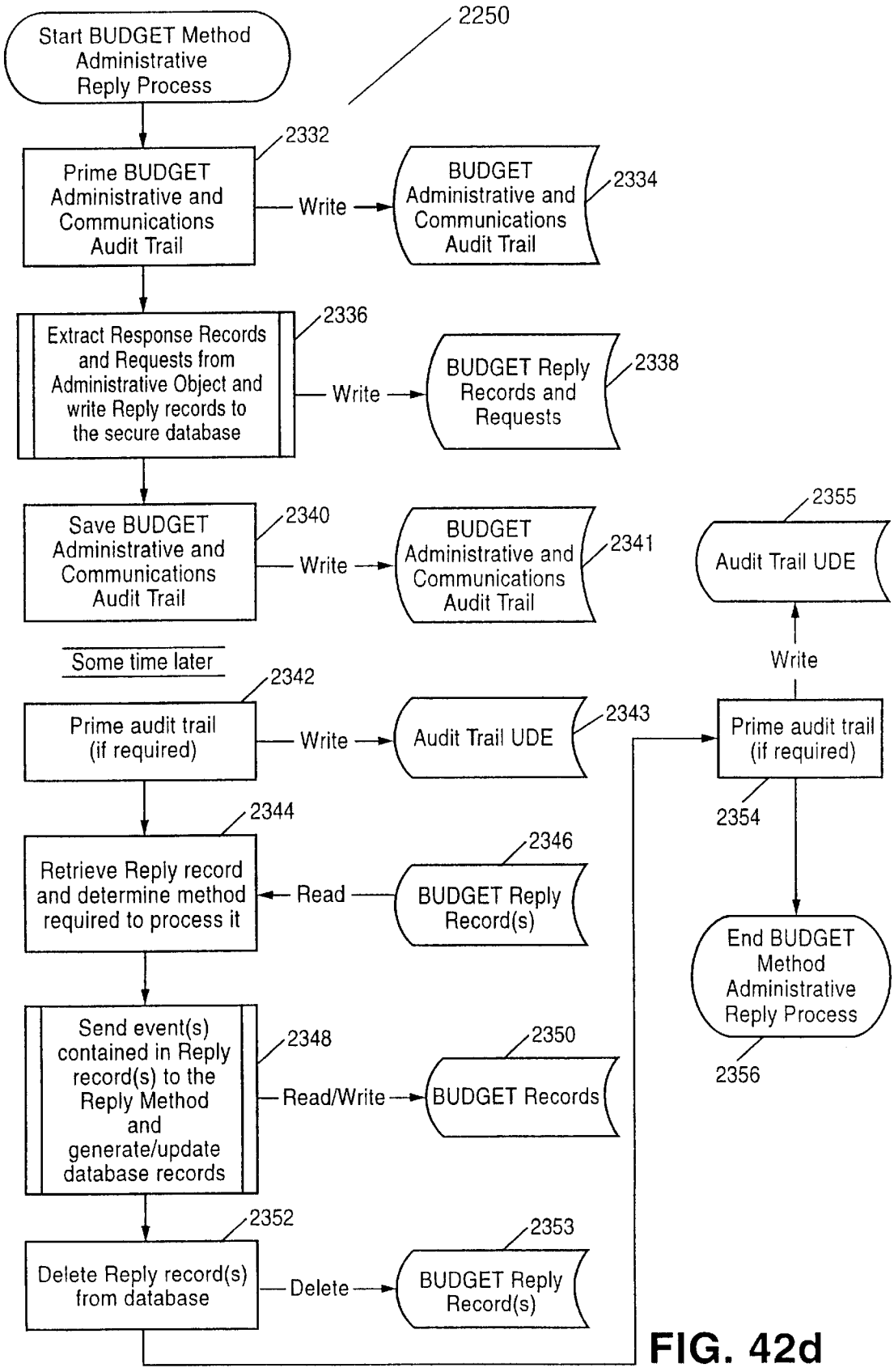

Start BUDGET Method Administrative Reply Process
2250
2332
Prime BUDGET Administrative and Communications Audit Trail
Write
BUDGET Administrative and Communications Audit Trail
2334
2336
Extract Response Records and Requests from Administrative Object and write Reply records to the secure database
Write
BUDGET Reply Records and Requests
2338
Save BUDGET Administrative and Communications Audit Trail
2340
Write
BUDGET Administrative and Communications Audit Trail
2341
2355
Audit Trail UDE
Write
Some time later
2342
Prime audit trail (if required)
Write
Audit Trail UDE
2343
Prime audit trail (if required)
2354
2344
Retrieve Reply record and determine method required to process it
Read
2346
BUDGET Reply Record(s)
End BUDGET Method Administrative Reply Process
2356
Send event(s) contained in Reply record(s) to the Reply Method and generate/update database records
2348
Read/Write
2350
BUDGET Records
2352
Delete Reply record(s) from database
Delete
2353
BUDGET Reply Record(s)

FIG. 42d

FIG. 43a

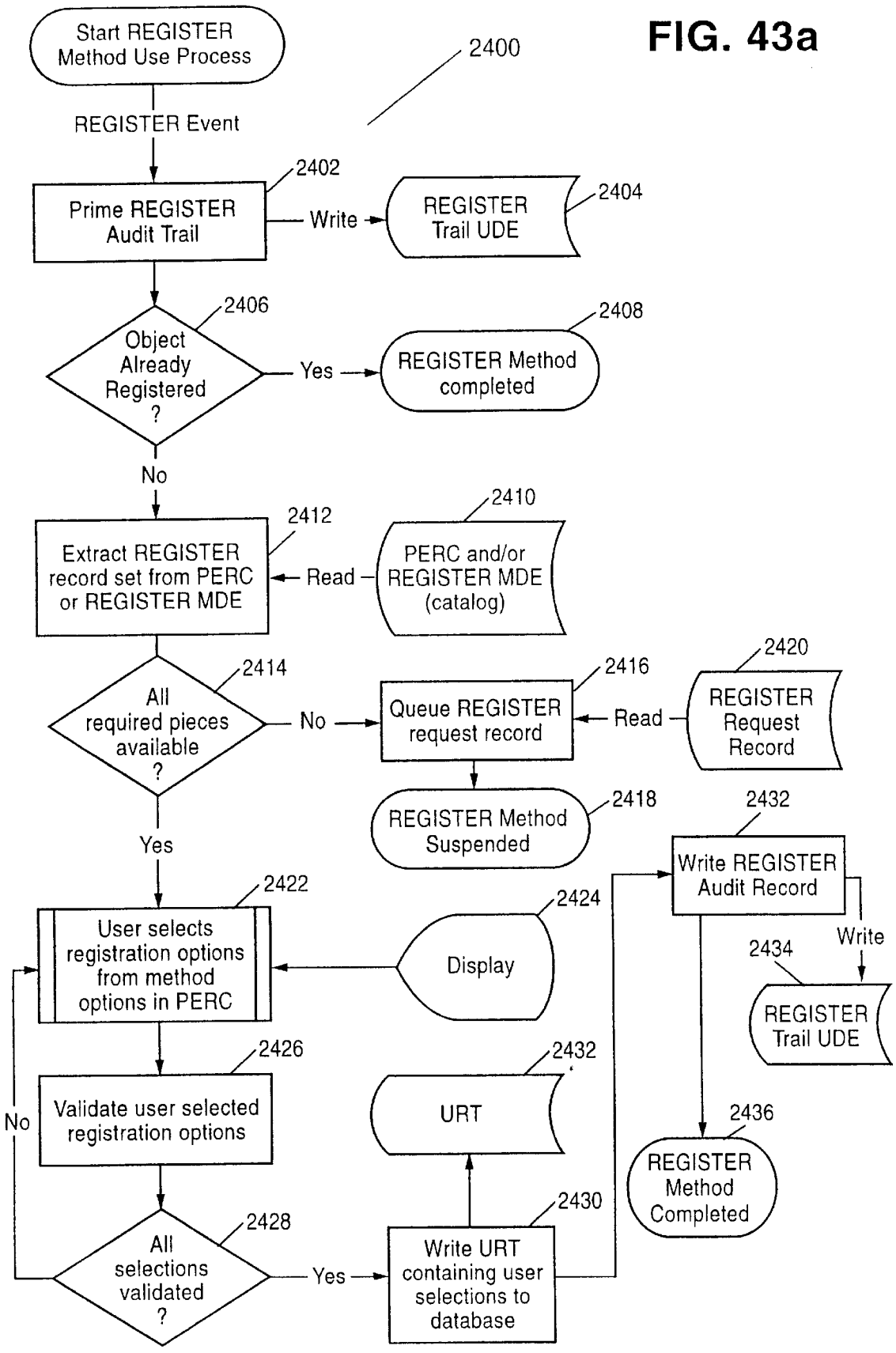
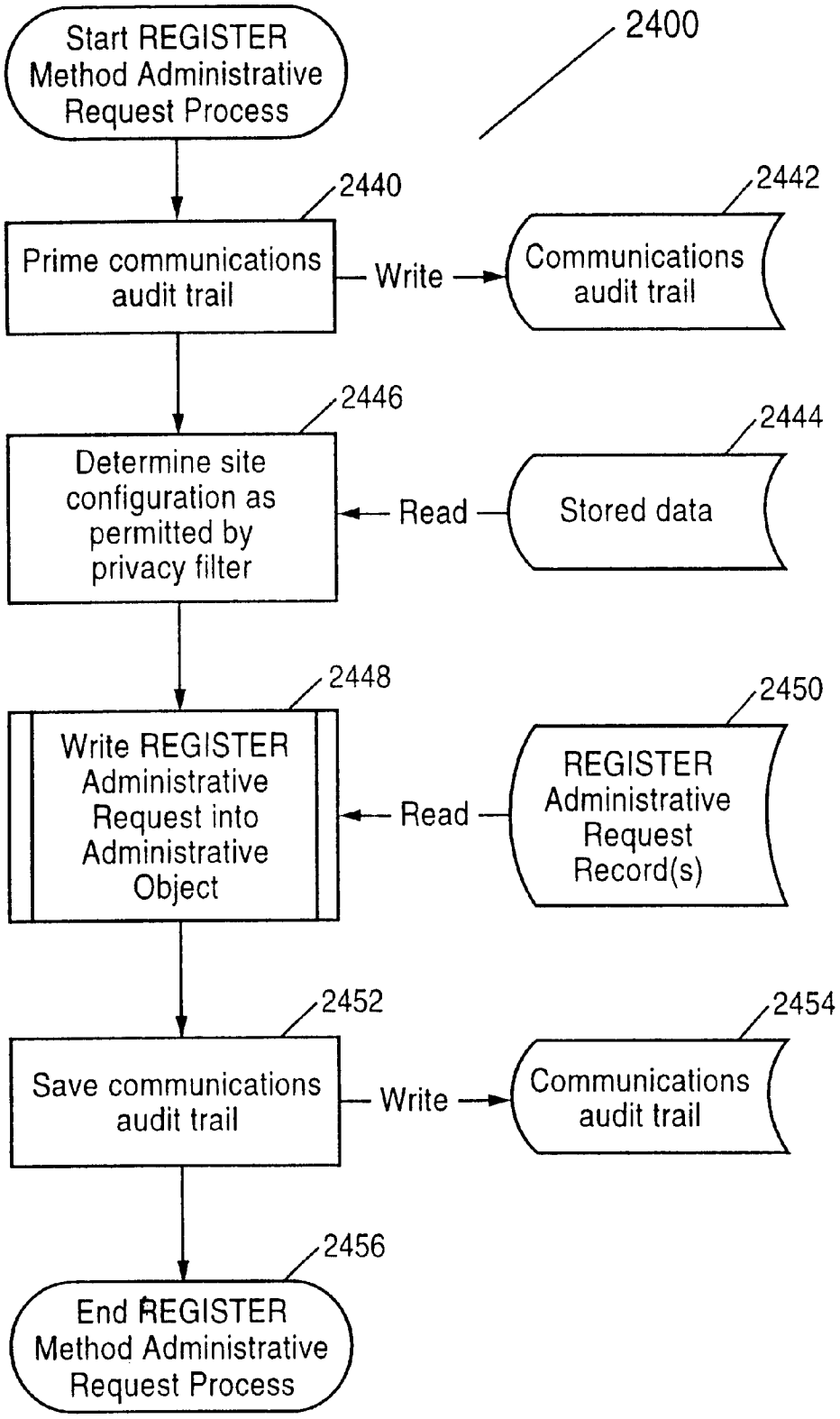

Start REGISTER Method Use Process
2400
REGISTER Event
2402
Prime REGISTER Audit Trail
Write
REGISTER Trail UDE
2404
2406
Object Already Registered ?
Yes
REGISTER Method completed
2408
No
2412
Extract REGISTER record set from PERC or REGISTER MDE
Read
PERC and/or REGISTER MDE (catalog)
2410
2414
All required pieces available ?
No
Queue REGISTER request record
2416
Read
REGISTER Request Record
2420
REGISTER Method Suspended
2418
Yes
2422
User selects registration options from method options in PERC
Display
2424
2426
Validate user selected registration options
No
2428
All selections validated ?
Yes
Write URT containing user selections to database
2430
URT
2432
Write REGISTER Audit Record
2432
Write
2434
REGISTER Trail UDE
2436
REGISTER Method Completed

FIG. 43b


Start REGISTER Method Administrative Request Process
2400
2440
Prime communications audit trail
Write
2442
Communications audit trail
2446
Determine site configuration as permitted by privacy filter
Read
2444
Stored data
2448
Write REGISTER Administrative Request into Administrative Object
Read
2450
REGISTER Administrative Request Record(s)
2452
Save communications audit trail
Write
2454
Communications audit trail
2456
End REGISTER Method Administrative Request Process

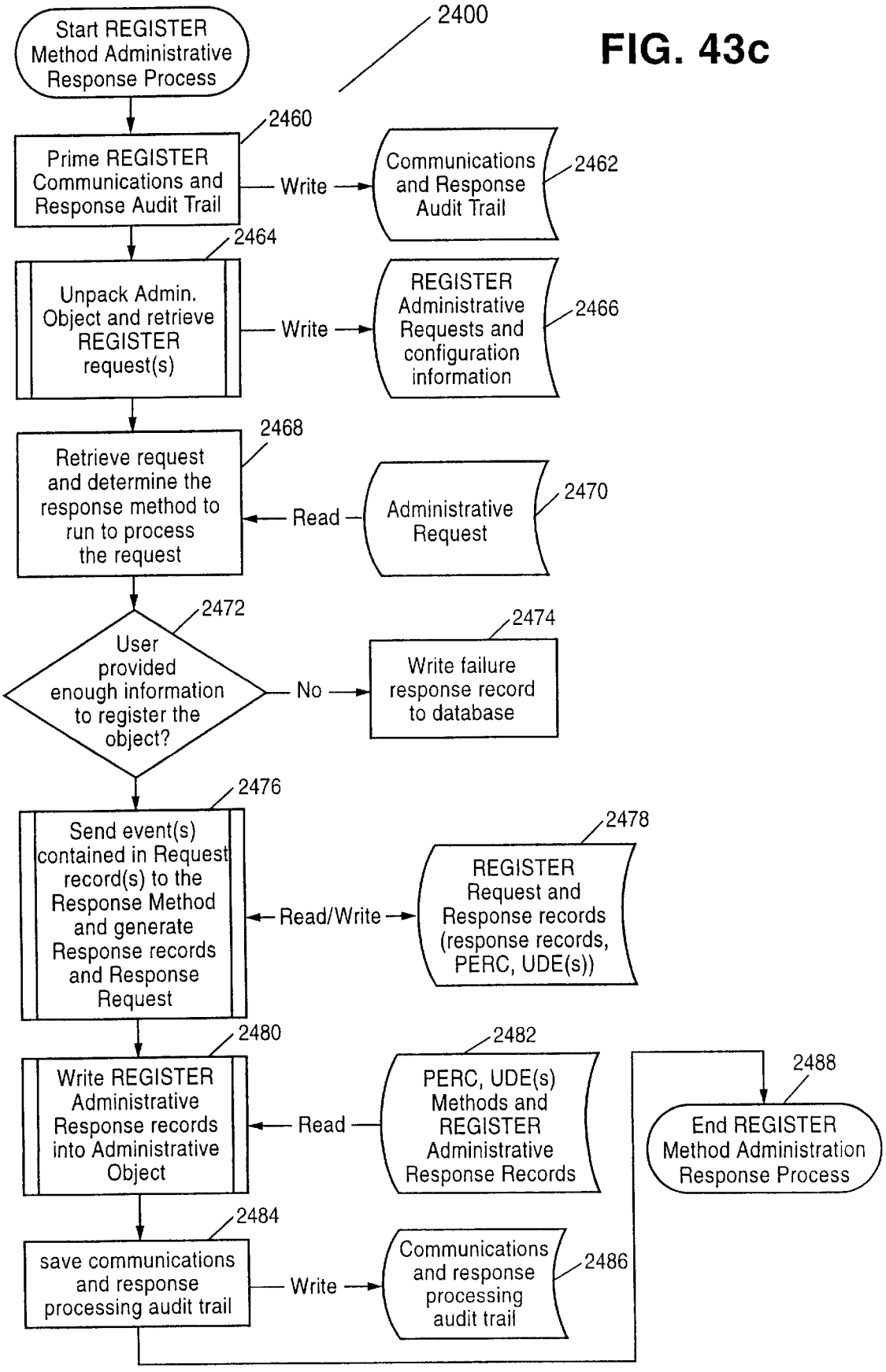

2400
Start REGISTER Method Administrative Response Process
2460
Prime REGISTER Communications and Response Audit Trail
Write
Communications and Response Audit Trail
2462
2464
Unpack Admin. Object and retrieve REGISTER request(s)
Write
REGISTER Administrative Requests and configuration information
2466
2468
Retrieve request and determine the response method to run to process the request
Read
Administrative Request
2470
2472
User provided enough information to register the object?
No
2474
Write failure response record to database
2476
Send event(s) contained in Request record(s) to the Response Method and generate Response records and Response Request
Read/Write
2478
REGISTER Request and Response records (response records, PERC, UDE(s))
2480
Write REGISTER Administrative Response records into Administrative Object
Read
2482
PERC, UDE(s) Methods and REGISTER Administrative Response Records
2488
End REGISTER Method Administration Response Process
2484
save communications and response processing audit trail
Write
Communications and response processing audit trail
2486

FIG. 43c

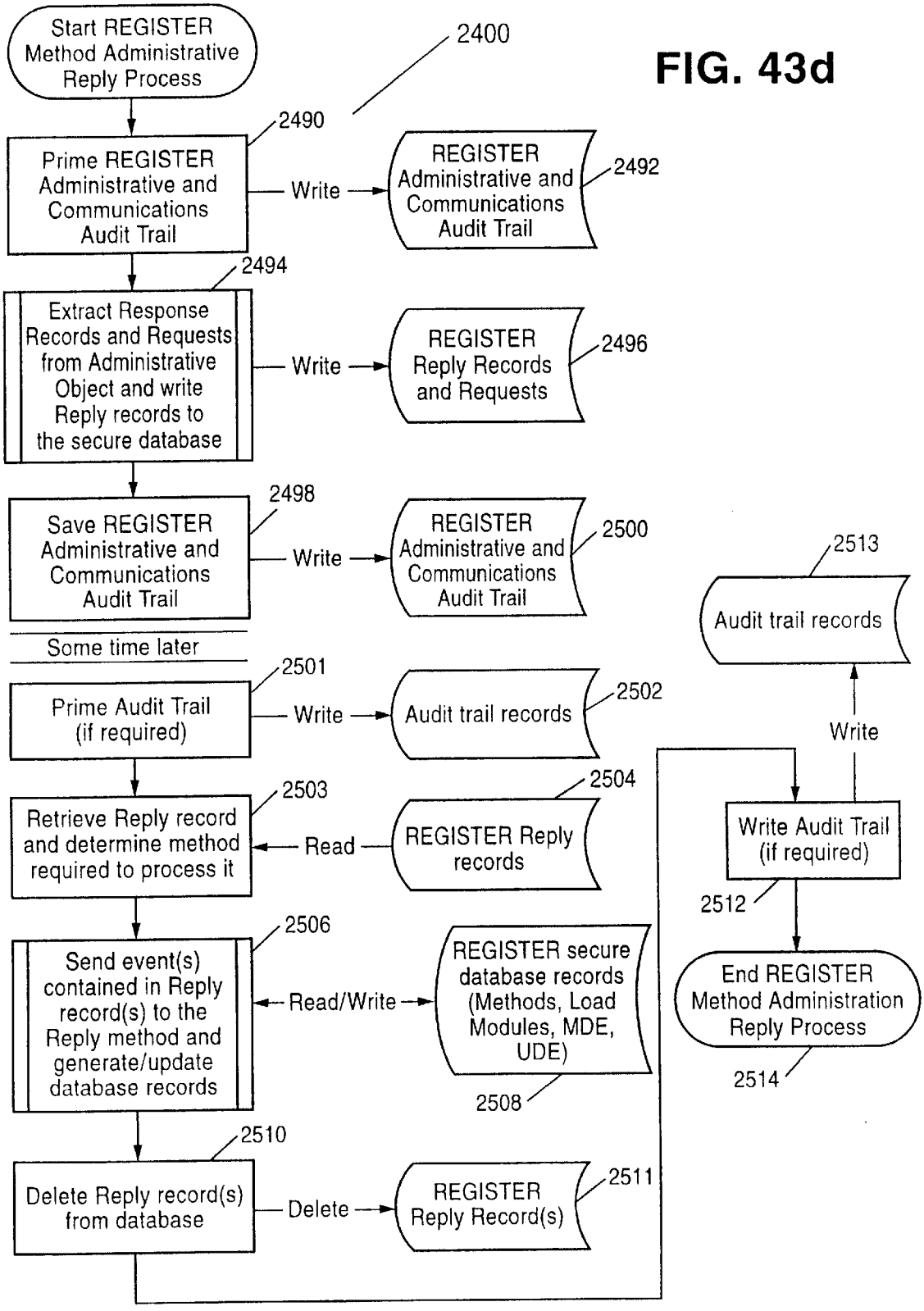

Start REGISTER Method Administrative Reply Process
2400
2490
Prime REGISTER Administrative and Communications Audit Trail
Write
REGISTER Administrative and Communications Audit Trail
2492
2494
Extract Response Records and Requests from Administrative Object and write Reply records to the secure database
Write
REGISTER Reply Records and Requests
2496
2498
Save REGISTER Administrative and Communications Audit Trail
Write
REGISTER Administrative and Communications Audit Trail
2500
Some time later
2501
Prime Audit Trail (if required)
Write
Audit trail records
2502
2503
Retrieve Reply record and determine method required to process it
Read
REGISTER Reply records
2504
2506
Send event(s) contained in Reply record(s) to the Reply method and generate/update database records
Read/Write
REGISTER secure database records (Methods, Load Modules, MDE, UDE)
2508
2510
Delete Reply record(s) from database
Delete
REGISTER Reply Record(s)
2511
2513
Audit trail records
Write
Write Audit Trail (if required)
2512
End REGISTER Method Administration Reply Process
2514

FIG. 43d

FIG. 44a

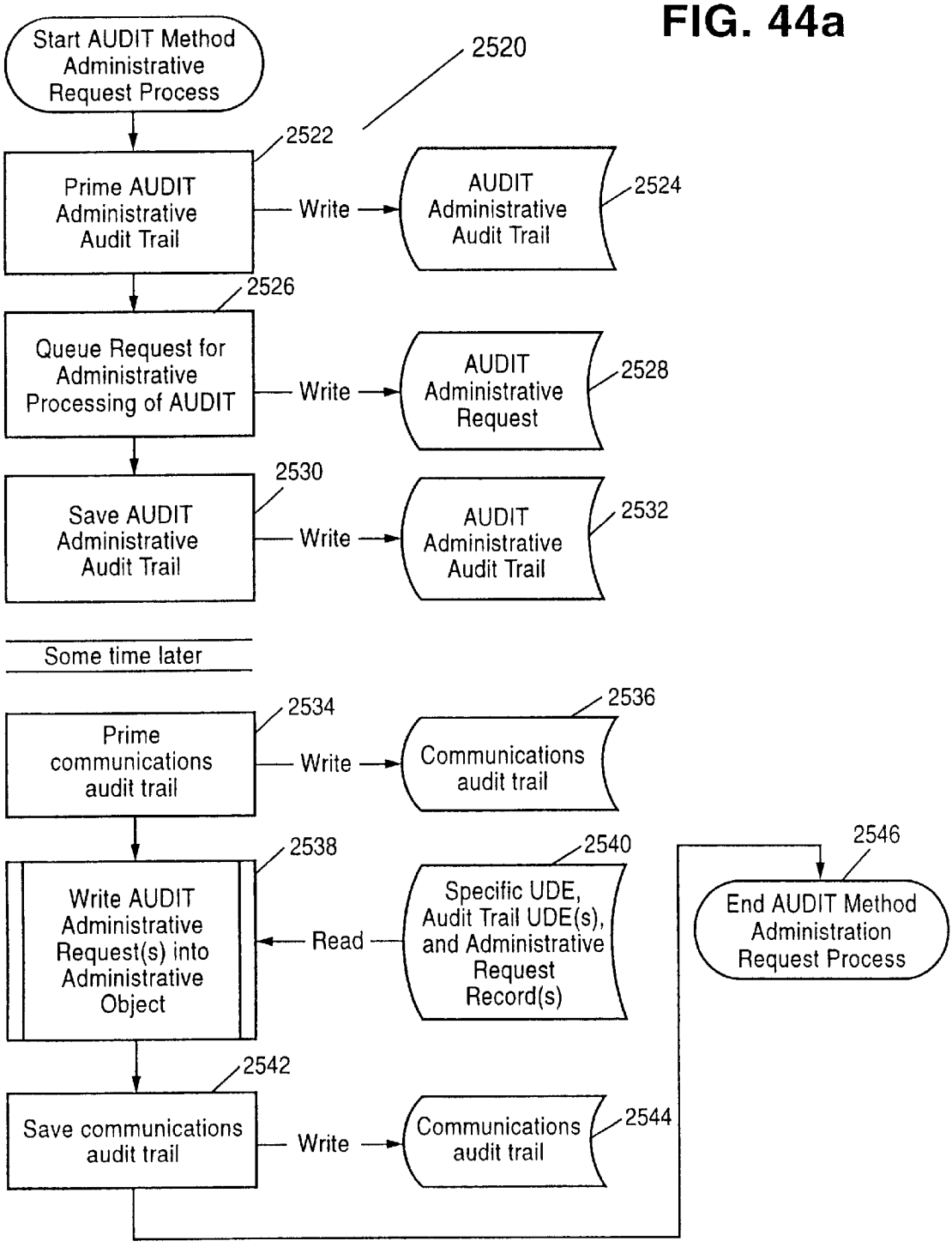

Start AUDIT Method Administrative Request Process
2520
2522
Prime AUDIT Administrative Audit Trail
Write
AUDIT Administrative Audit Trail
2524
2526
Queue Request for Administrative Processing of AUDIT
Write
AUDIT Administrative Request
2528
2530
Save AUDIT Administrative Audit Trail
Write
AUDIT Administrative Audit Trail
2532
Some time later
2534
Prime communications audit trail
Write
Communications audit trail
2536
2538
Write AUDIT Administrative Request(s) into Administrative Object
Read
Specific UDE, Audit Trail UDE(s), and Administrative Request Record(s)
2540
2546
End AUDIT Method Administration Request Process
2542
Save communications audit trail
Write
Communications audit trail
2544

FIG. 44b

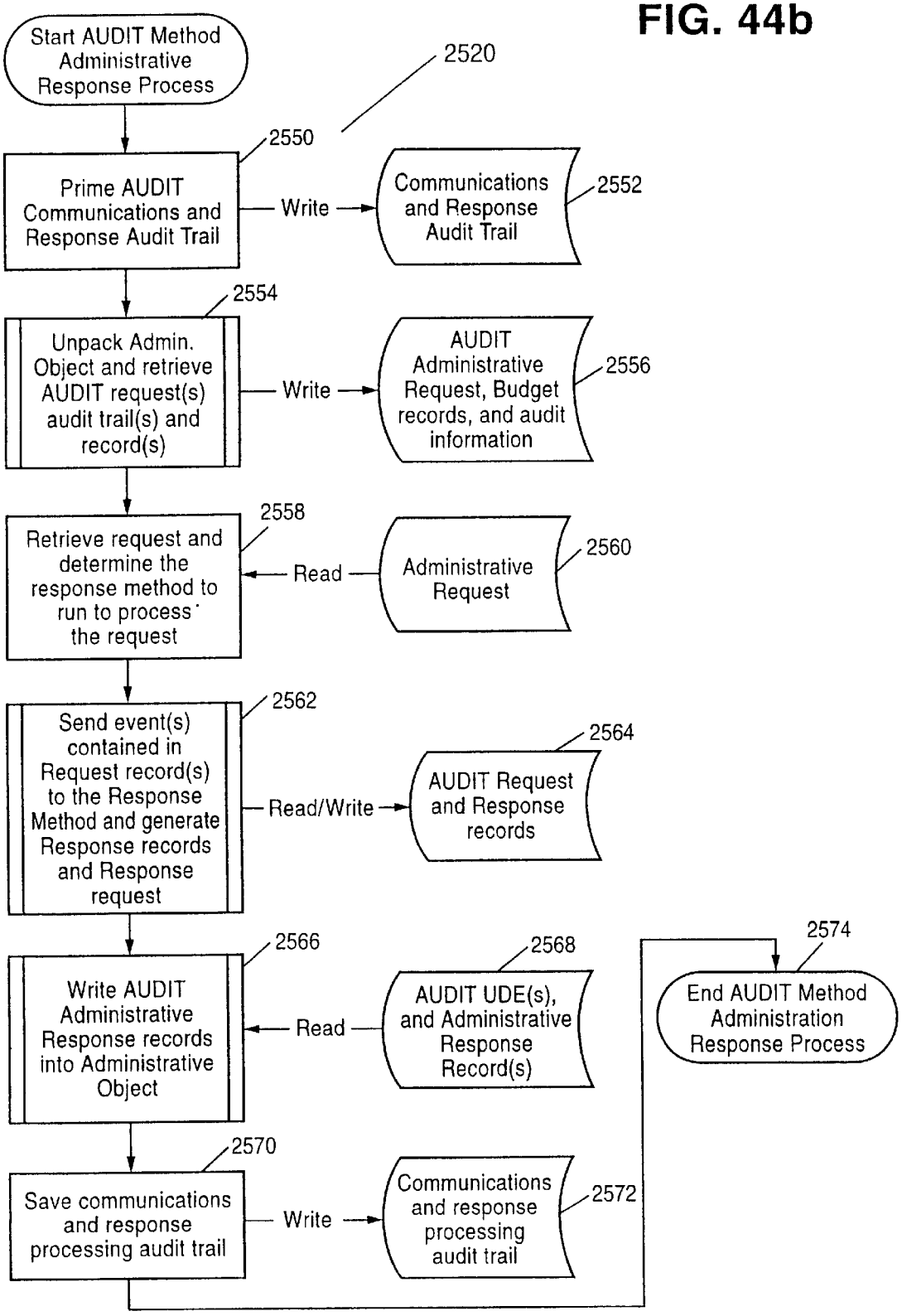
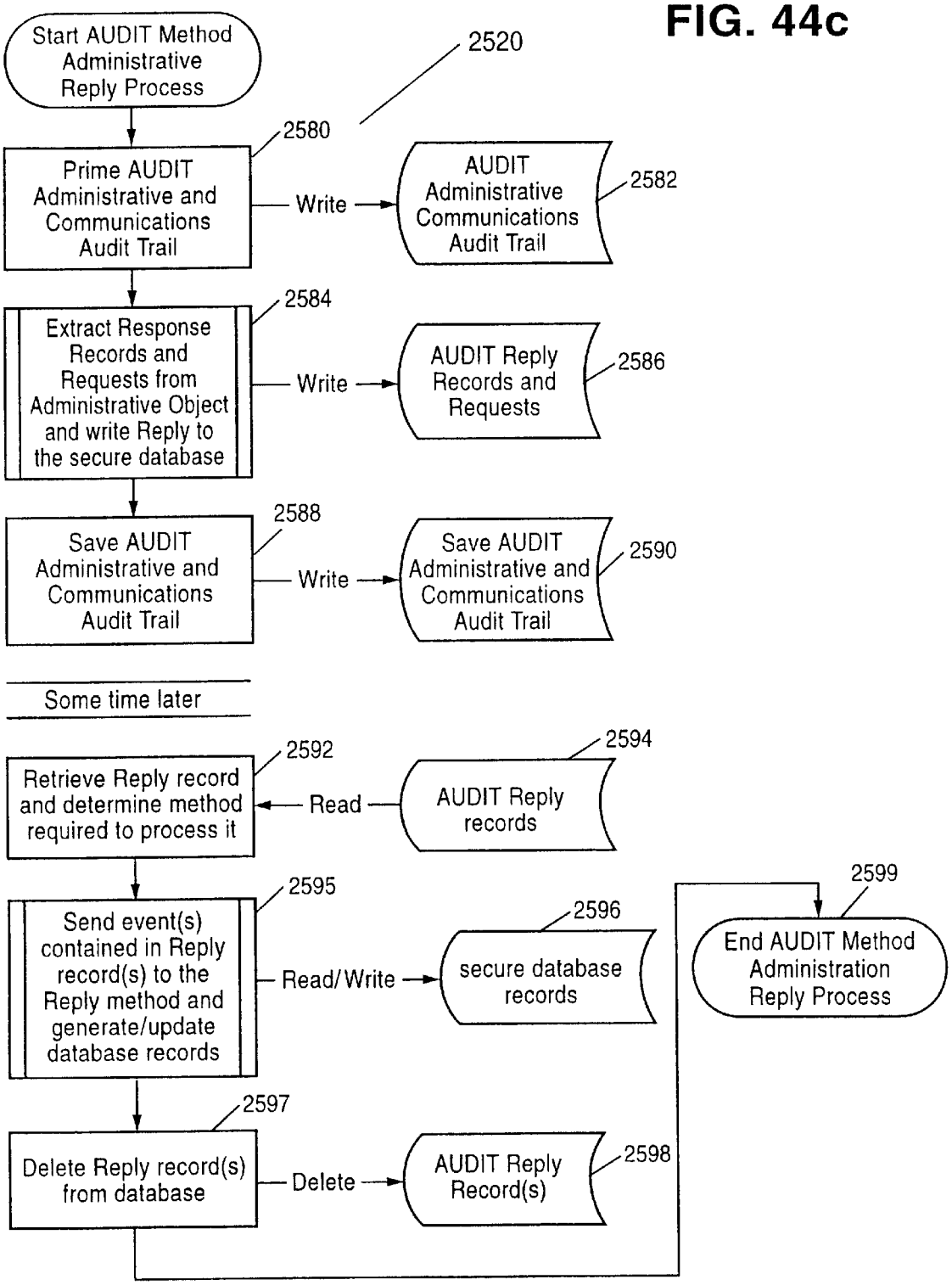

Start AUDIT Method Administrative Response Process
2520
2550
Prime AUDIT Communications and Response Audit Trail
Write
Communications and Response Audit Trail
2552
2554
Unpack Admin. Object and retrieve AUDIT request(s) audit trail(s) and record(s)
Write
AUDIT Administrative Request, Budget records, and audit information
2556
2558
Retrieve request and determine the response method to run to process the request
Read
Administrative Request
2560
2562
Send event(s) contained in Request record(s) to the Response Method and generate Response records and Response request
Read/Write
AUDIT Request and Response records
2564
2566
Write AUDIT Administrative Response records into Administrative Object
Read
AUDIT UDE(s), and Administrative Response Record(s)
2568
2570
Save communications and response processing audit trail
Write
Communications and response processing audit trail
2572
2574
End AUDIT Method Administration Response Process

FIG. 44c


Start AUDIT Method Administrative Reply Process
2520
2580
Prime AUDIT Administrative and Communications Audit Trail
Write
AUDIT Administrative Communications Audit Trail
2582
2584
Extract Response Records and Requests from Administrative Object and write Reply to the secure database
Write
AUDIT Reply Records and Requests
2586
2588
Save AUDIT Administrative and Communications Audit Trail
Write
Save AUDIT Administrative and Communications Audit Trail
2590
Some time later
2592
Retrieve Reply record and determine method required to process it
Read
AUDIT Reply records
2594
2595
Send event(s) contained in Reply record(s) to the Reply method and generate/update database records
Read/Write
secure database records
2596
2599
End AUDIT Method Administration Reply Process
2597
Delete Reply record(s) from database
Delete
AUDIT Reply Record(s)
2598

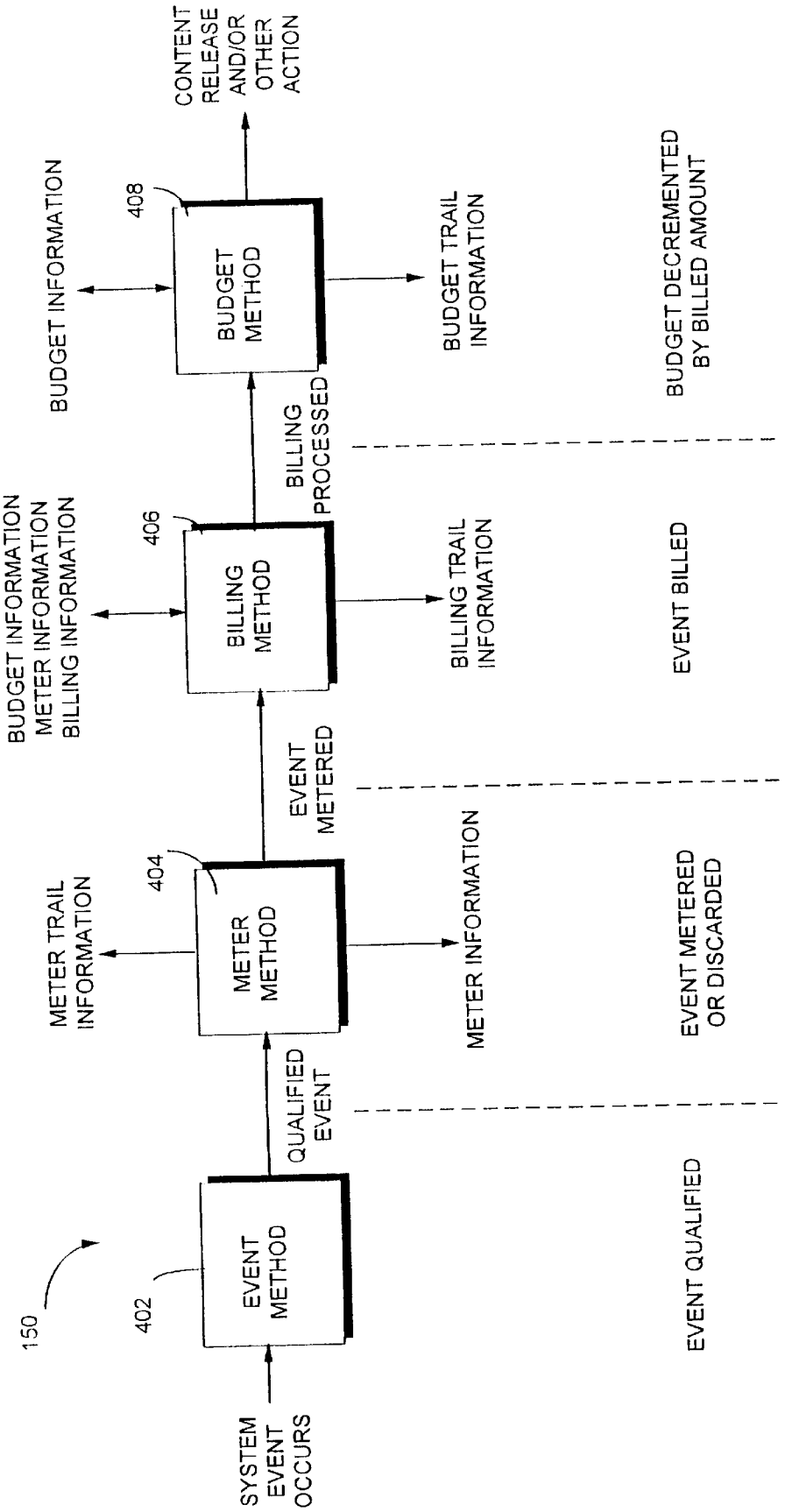

150
SYSTEM EVENT OCCURS
402
EVENT METHOD
QUALIFIED EVENT
EVENT QUALIFIED
404
METER METHOD
METER TRAIL INFORMATION
METER INFORMATION
EVENT METERED OR DISCARDED
EVENT METERED
406
BILLING METHOD
BUDGET INFORMATION
METER INFORMATION
BILLING INFORMATION
BILLING TRAIL INFORMATION
EVENT BILLED
BILLING PROCESSED
408
BUDGET METHOD
BUDGET INFORMATION
BUDGET TRAIL INFORMATION
BUDGET DECREMENTED BY BILLED AMOUNT
CONTENT RELEASE AND/OR OTHER ACTION

FIG. 45

FIG. 46

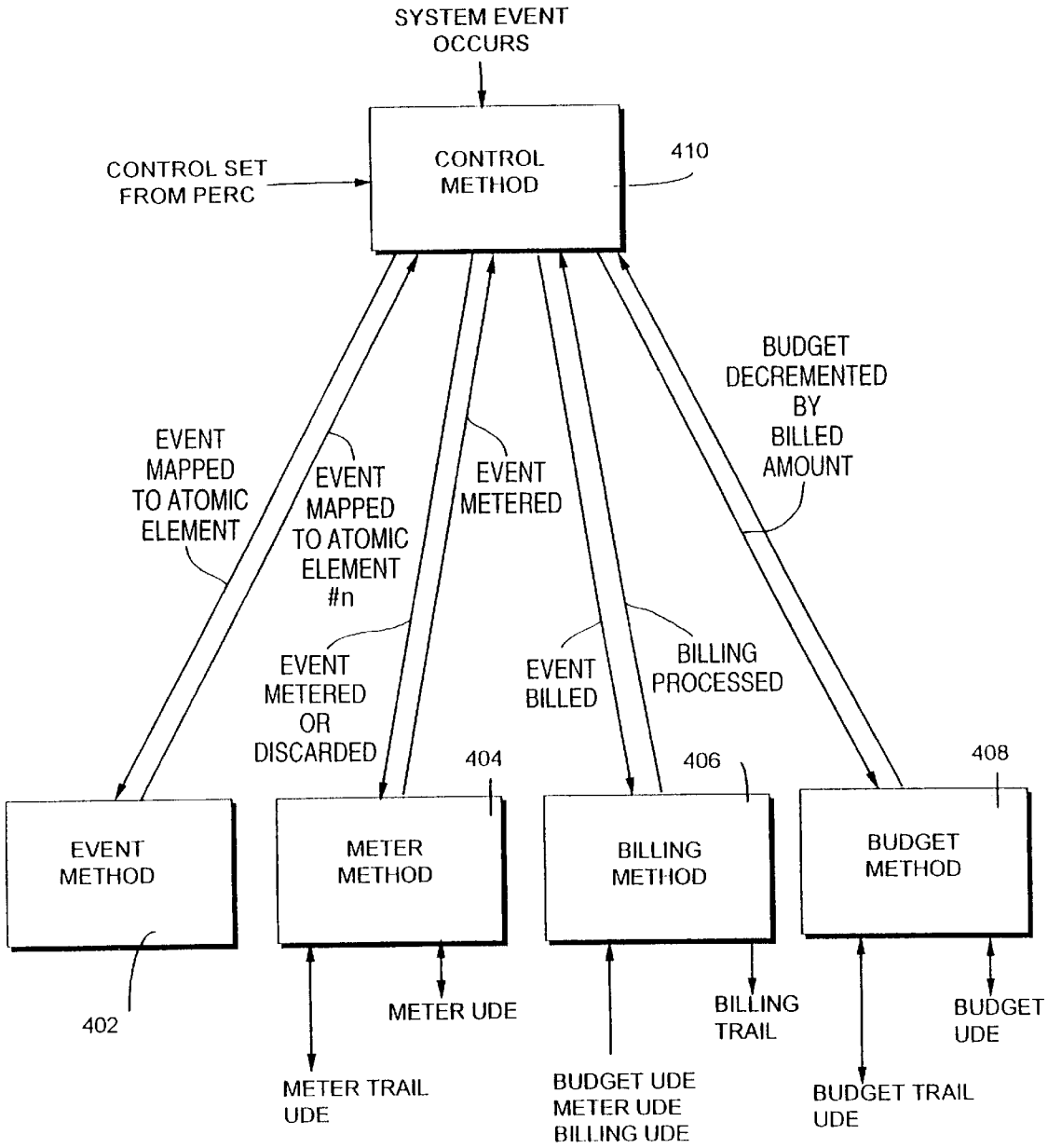
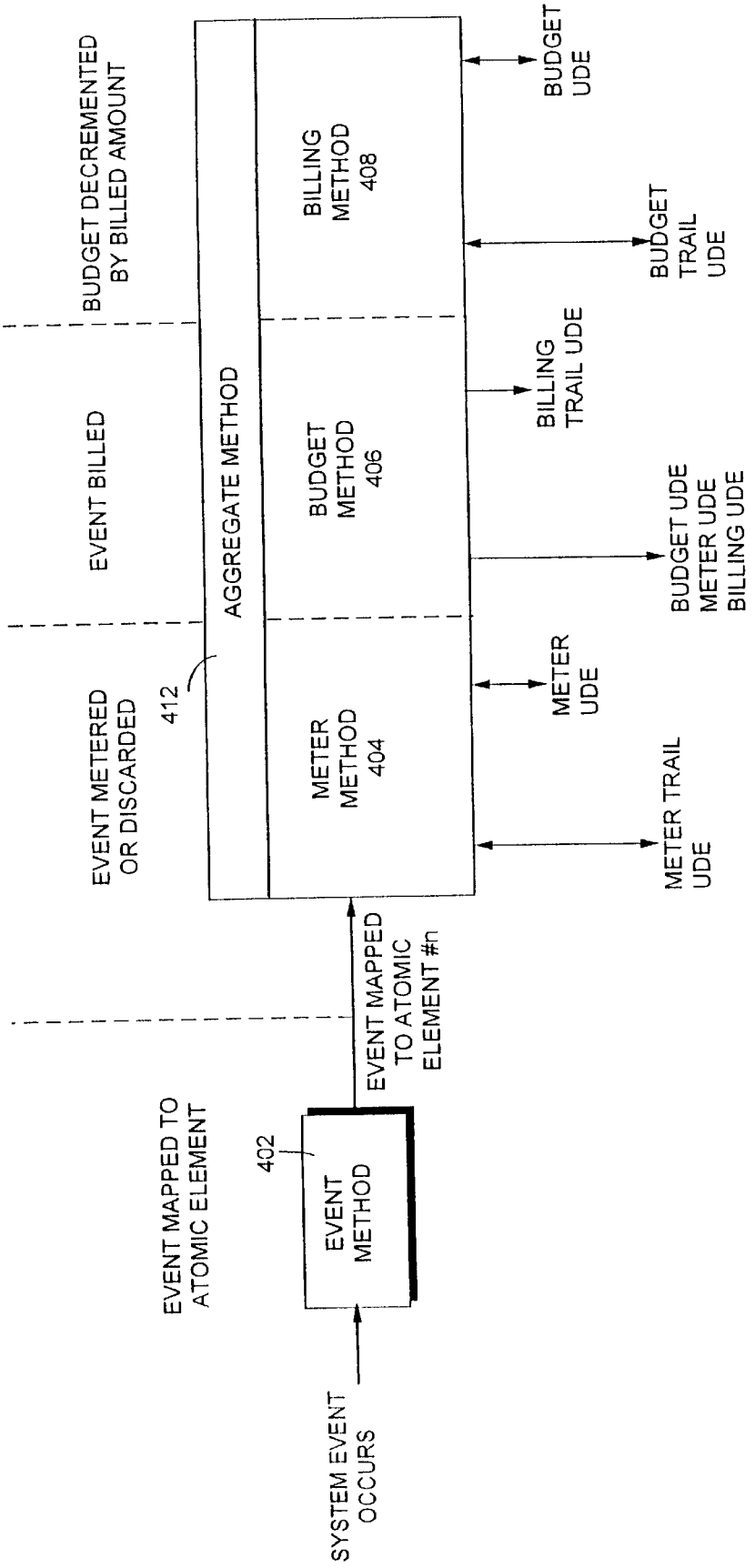

SYSTEM EVENT OCCURS
CONTROL METHOD
410
CONTROL SET FROM PERC
EVENT MAPPED TO ATOMIC ELEMENT
EVENT MAPPED TO ATOMIC ELEMENT #n
EVENT METERED
BUDGET DECREMENTED BY BILLED AMOUNT
EVENT METERED OR DISCARDED
EVENT BILLED
BILLING PROCESSED
EVENT METHOD
METER METHOD
BILLING METHOD
BUDGET METHOD
402
404
406
408
METER UDE
METER TRAIL UDE
BUDGET UDE METER UDE BILLING UDE
BILLING TRAIL
BUDGET UDE
BUDGET TRAIL UDE

FIG. 47


SYSTEM EVENT OCCURS
EVENT MAPPED TO ATOMIC ELEMENT
402
EVENT METHOD
EVENT MAPPED TO ATOMIC ELEMENT #n
412
EVENT METERED OR DISCARDED
EVENT BILLED
BUDGET DECREMENTED BY BILLED AMOUNT
AGGREGATE METHOD
METER METHOD 404
BUDGET METHOD 406
BILLING METHOD 408
METER UDE
METER TRAIL UDE
BILLING TRAIL UDE
BUDGET UDE METER UDE BILLING UDE
BUDGET TRAIL UDE
BUDGET UDE

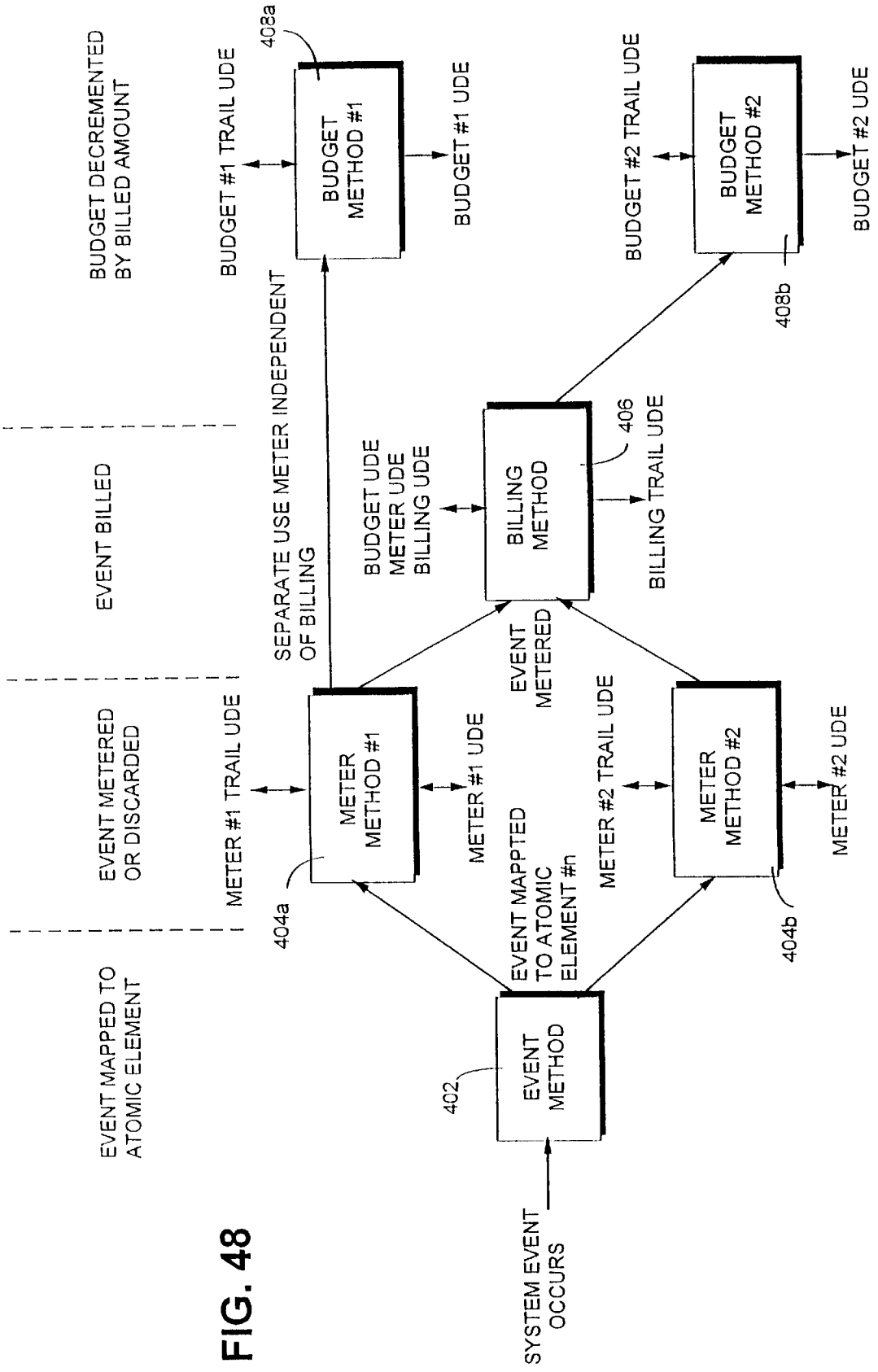

EVENT MAPPED TO ATOMIC ELEMENT
EVENT METERED OR DISCARDED
EVENT BILLED
BUDGET DECREMENTED BY BILLED AMOUNT
SYSTEM EVENT OCCURS
EVENT METHOD
402
EVENT MAPPTED TO ATOMIC ELEMENT #n
METER METHOD #1
404a
METER #1 TRAIL UDE
METER #1 UDE
METER METHOD #2
404b
METER #2 TRAIL UDE
METER #2 UDE
EVENT METERED
SEPARATE USE METER INDEPENDENT OF BILLING
BILLING METHOD
406
BUDGET UDE METER UDE BILLING UDE
BILLING TRAIL UDE
BUDGET METHOD #1
408a
BUDGET #1 TRAIL UDE
BUDGET #1 UDE
BUDGET METHOD #2
408b
BUDGET #2 TRAIL UDE
BUDGET #2 UDE

FIG. 48

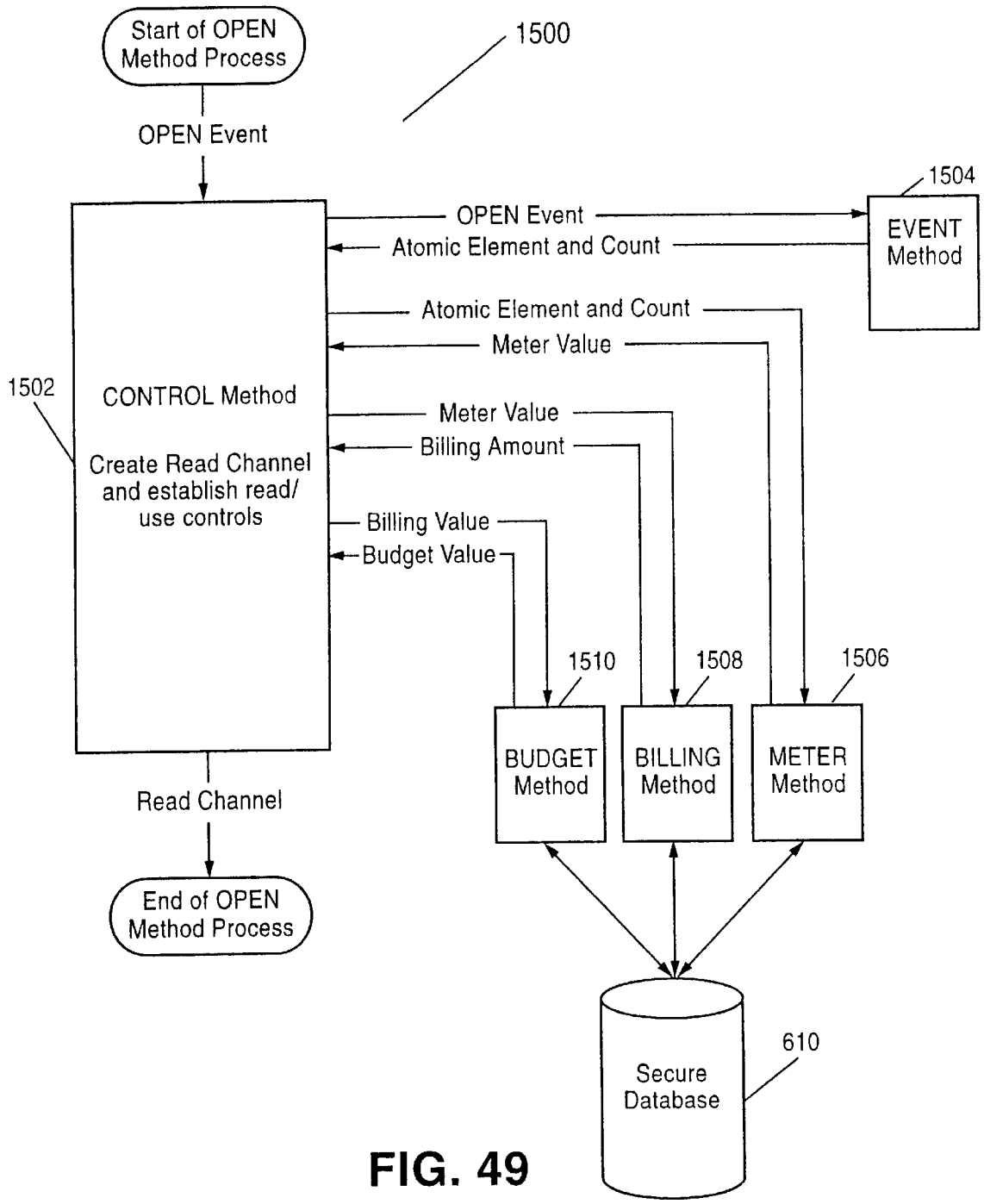

Start of OPEN Method Process
1500
OPEN Event
1504
OPEN Event
Atomic Element and Count
EVENT Method
Atomic Element and Count
Meter Value
1502
CONTROL Method
Create Read Channel and establish read/ use controls
Meter Value
Billing Amount
Billing Value
Budget Value
1510
1508
1506
BUDGET Method
BILLING Method
METER Method
Read Channel
End of OPEN Method Process
610
Secure Database

FIG. 49

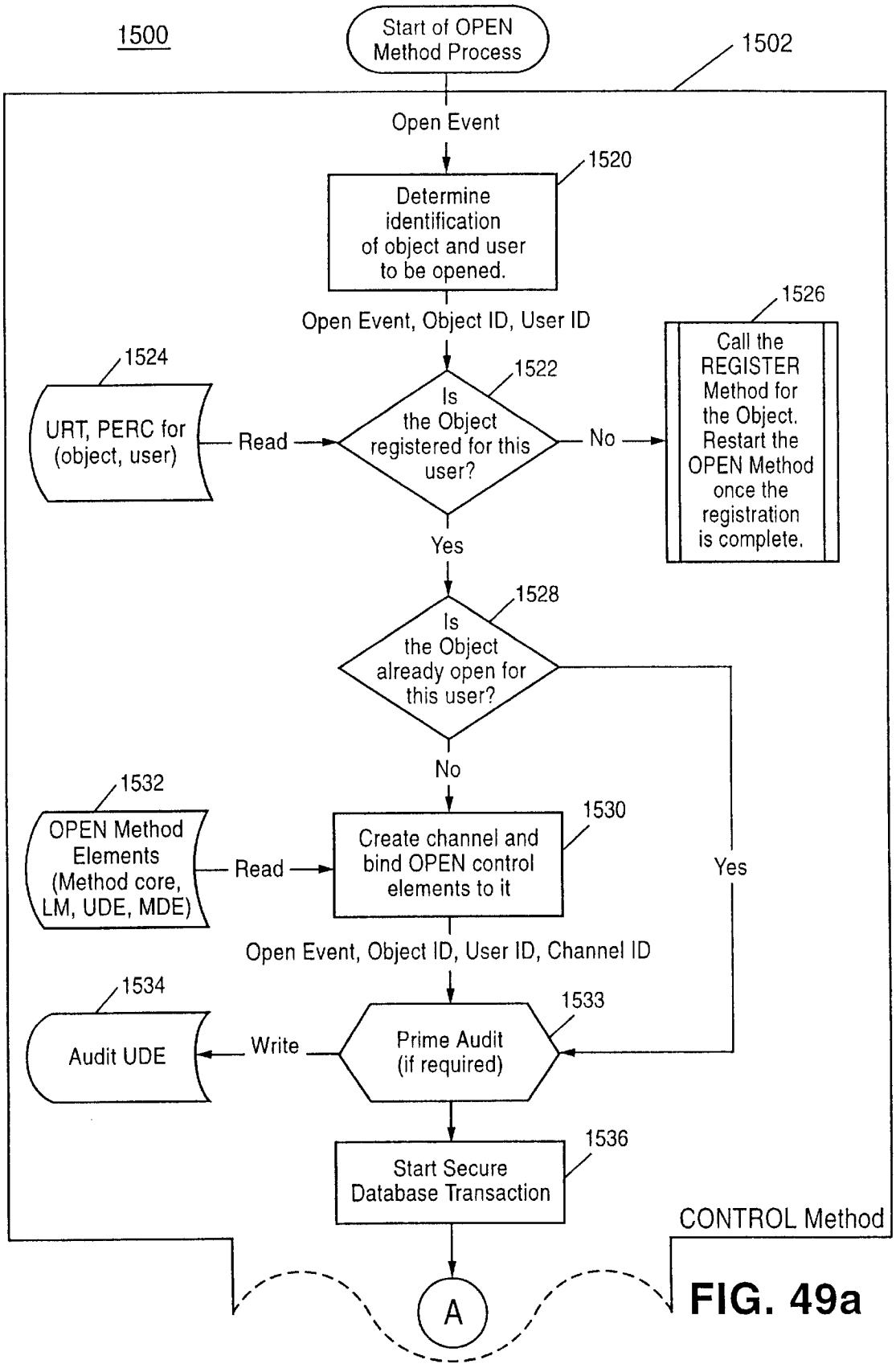

1500
Start of OPEN Method Process
1502
Open Event
1520
Determine identification of object and user to be opened.
Open Event, Object ID, User ID
1526
Call the REGISTER Method for the Object. Restart the OPEN Method once the registration is complete.
1524
URT, PERC for (object, user)
Read
1522
Is the Object registered for this user?
No
Yes
1528
Is the Object already open for this user?
No
Yes
1532
OPEN Method Elements (Method core, LM, UDE, MDE)
Read
1530
Create channel and bind OPEN control elements to it
Open Event, Object ID, User ID, Channel ID
1534
Audit UDE
Write
1533
Prime Audit (if required)
1536
Start Secure Database Transaction
CONTROL Method
A

FIG. 49a

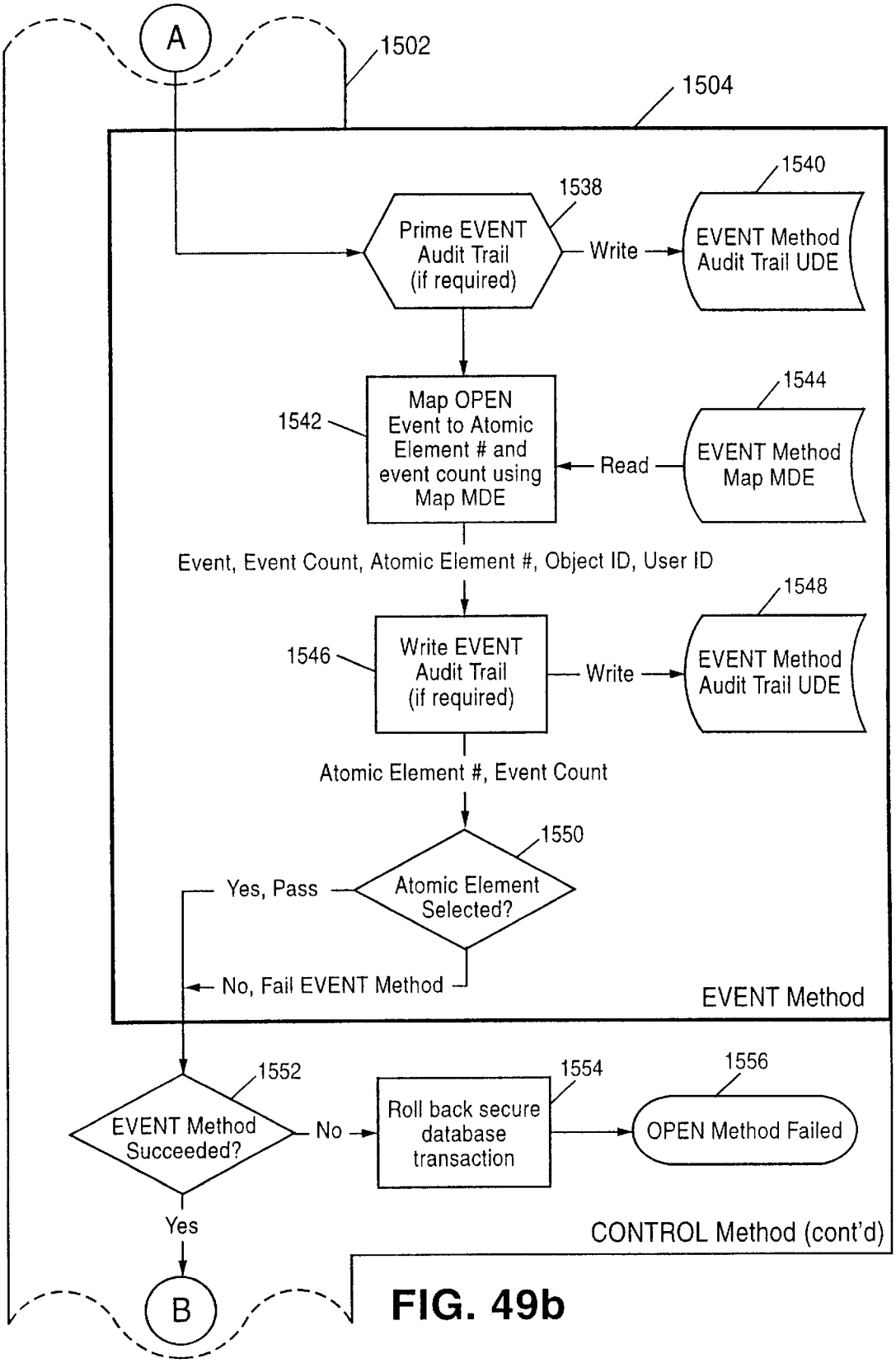

A
1502
1504
1538
Prime EVENT Audit Trail (if required)
Write
1540
EVENT Method Audit Trail UDE
1542
Map OPEN Event to Atomic Element # and event count using Map MDE
Read
1544
EVENT Method Map MDE
Event, Event Count, Atomic Element #, Object ID, User ID
1546
Write EVENT Audit Trail (if required)
Write
1548
EVENT Method Audit Trail UDE
Atomic Element #, Event Count
1550
Atomic Element Selected?
Yes, Pass
No, Fail EVENT Method
EVENT Method
1552
EVENT Method Succeeded?
No
1554
Roll back secure database transaction
1556
OPEN Method Failed
Yes
CONTROL Method (cont'd)
B

FIG. 49b

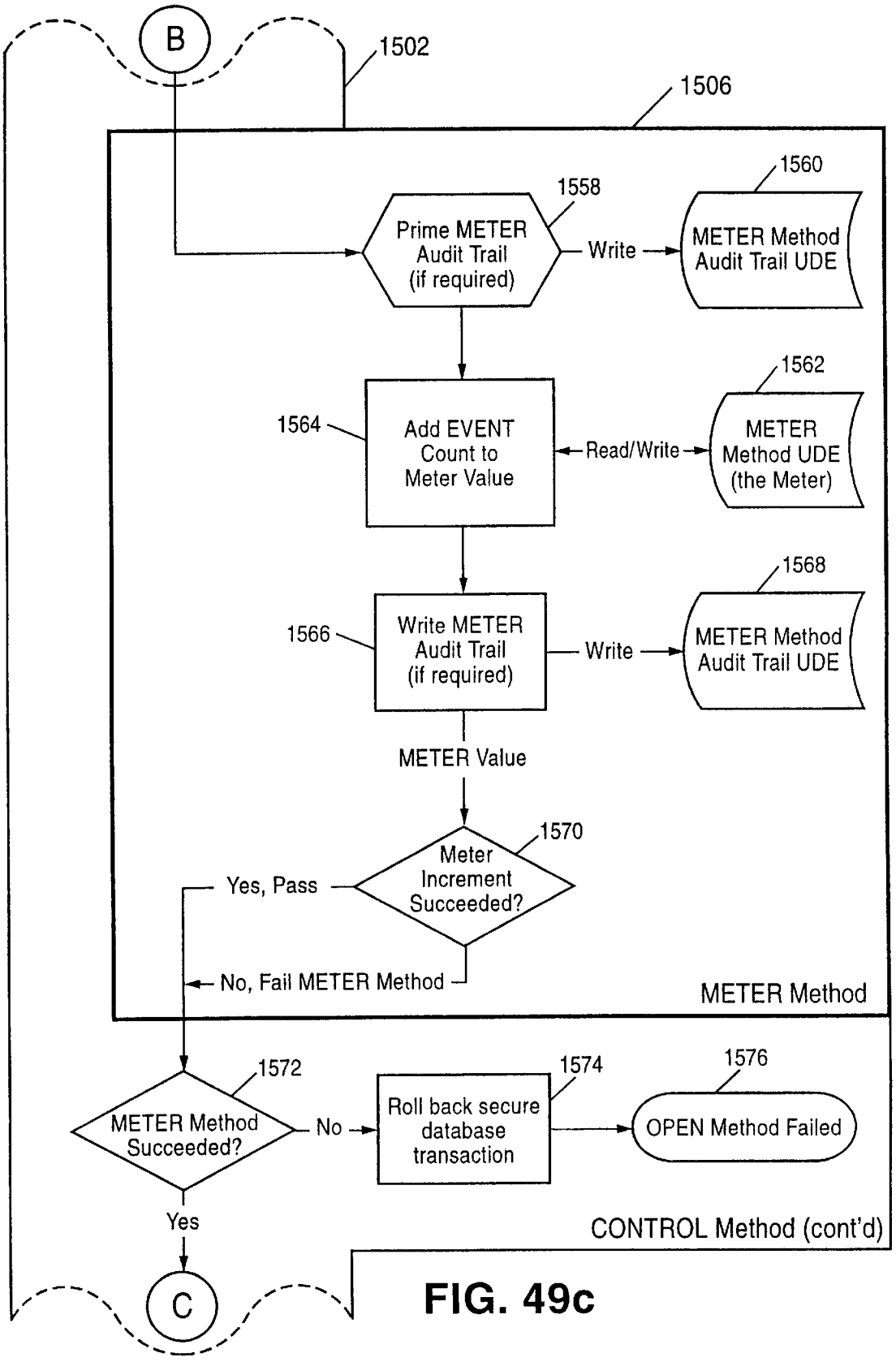

B
1502
1506
1558
Prime METER Audit Trail (if required)
Write
1560
METER Method Audit Trail UDE
1562
1564
Add EVENT Count to Meter Value
Read/Write
METER Method UDE (the Meter)
1568
1566
Write METER Audit Trail (if required)
Write
METER Method Audit Trail UDE
METER Value
1570
Meter Increment Succeeded?
Yes, Pass
No, Fail METER Method
METER Method
1572
METER Method Succeeded?
No
1574
Roll back secure database transaction
1576
OPEN Method Failed
Yes
CONTROL Method (cont'd)
C

FIG. 49c

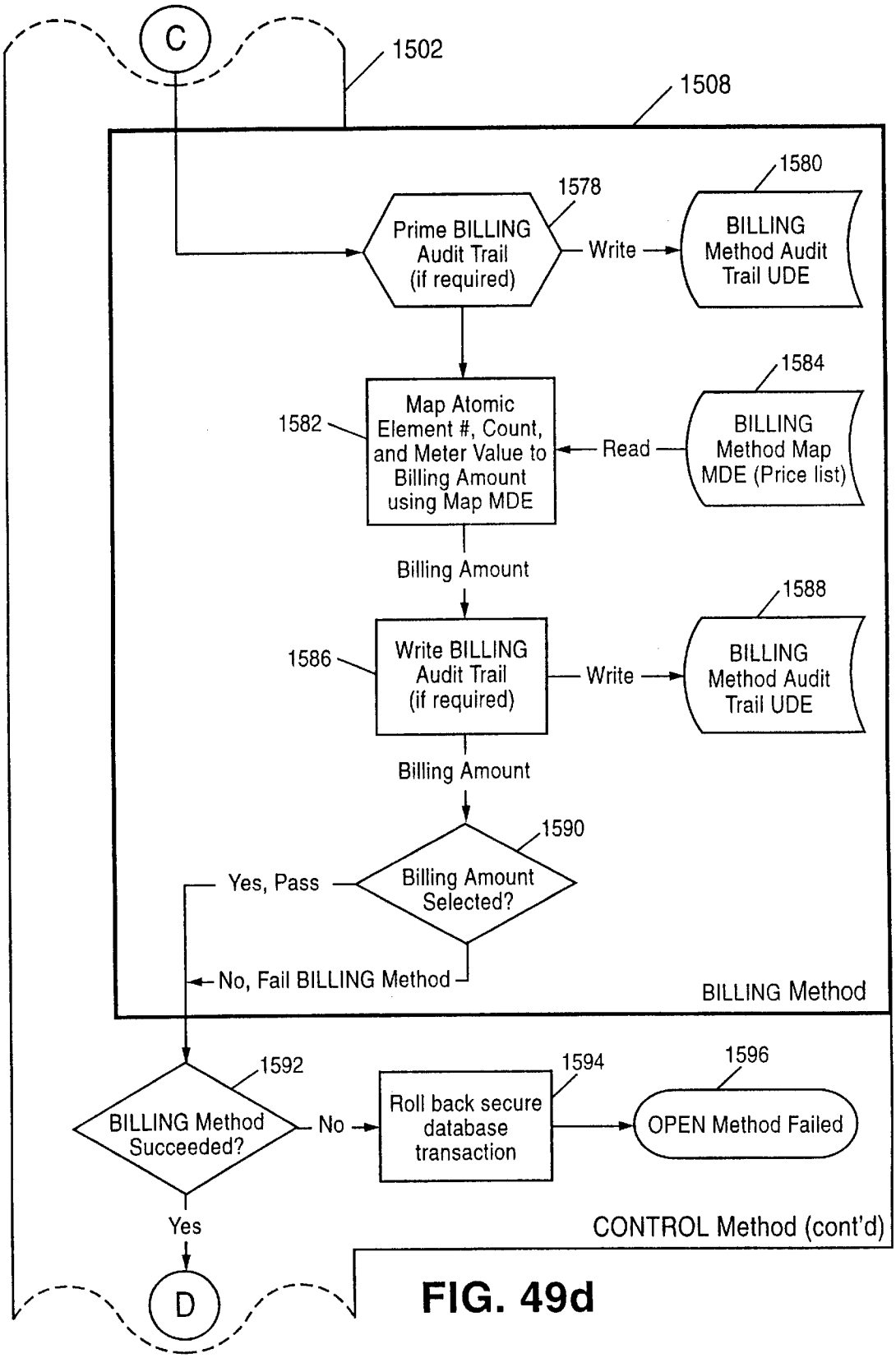

C
1502
1508
1578
Prime BILLING Audit Trail (if required)
Write
1580
BILLING Method Audit Trail UDE
1584
1582
Map Atomic Element #, Count, and Meter Value to Billing Amount using Map MDE
Read
BILLING Method Map MDE (Price list)
Billing Amount
1588
1586
Write BILLING Audit Trail (if required)
Write
BILLING Method Audit Trail UDE
Billing Amount
1590
Yes, Pass
Billing Amount Selected?
No, Fail BILLING Method
BILLING Method
1592
1594
1596
BILLING Method Succeeded?
No
Roll back secure database transaction
OPEN Method Failed
Yes
CONTROL Method (cont'd)
D

FIG. 49d

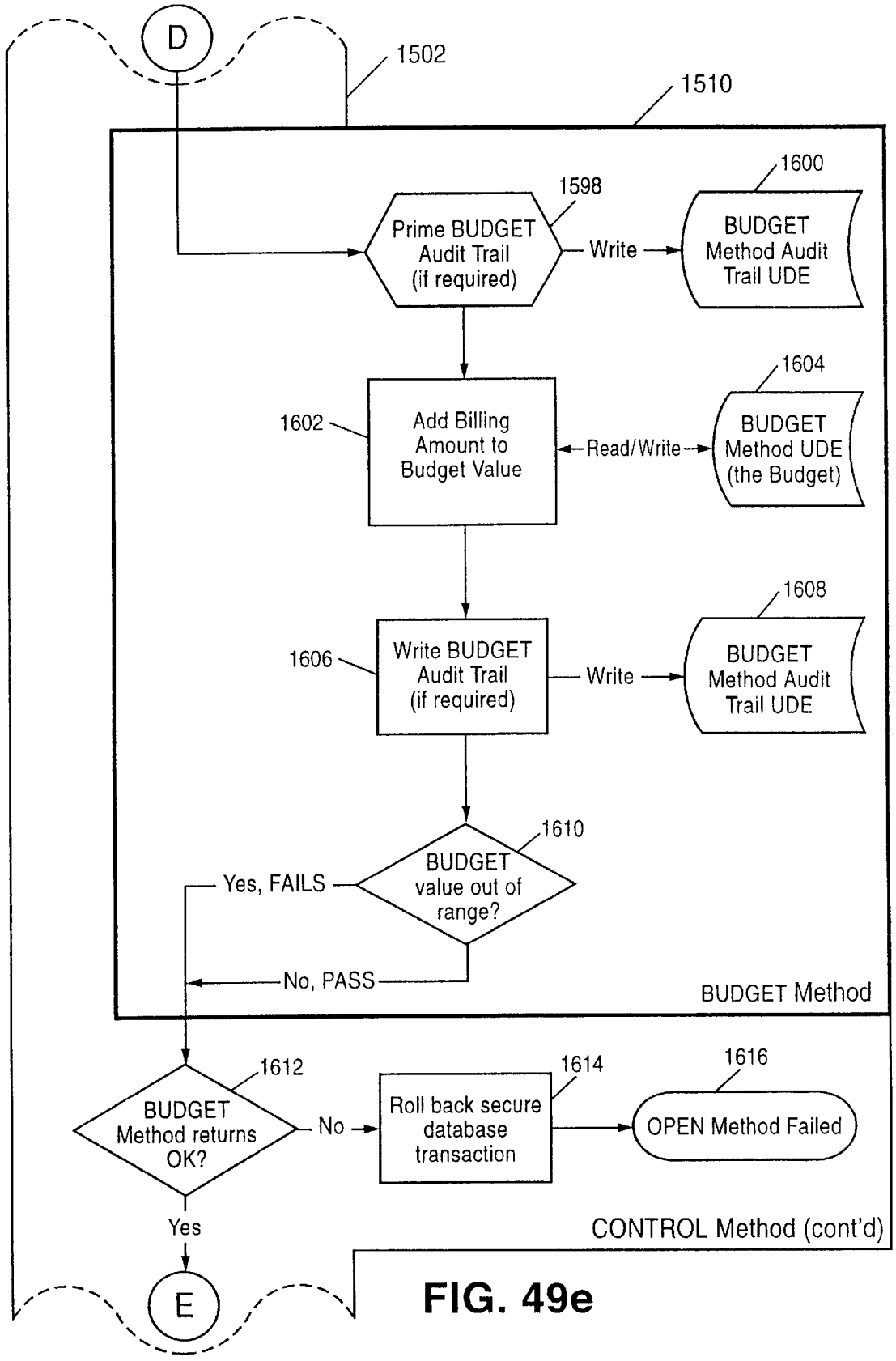

D
1502
1510
1598
Prime BUDGET Audit Trail (if required)
Write
1600
BUDGET Method Audit Trail UDE
1602
Add Billing Amount to Budget Value
Read/Write
1604
BUDGET Method UDE (the Budget)
1606
Write BUDGET Audit Trail (if required)
Write
1608
BUDGET Method Audit Trail UDE
1610
BUDGET value out of range?
Yes, FAILS
No, PASS
BUDGET Method
1612
BUDGET Method returns OK?
No
1614
Roll back secure database transaction
1616
OPEN Method Failed
Yes
CONTROL Method (cont'd)
E

FIG. 49e

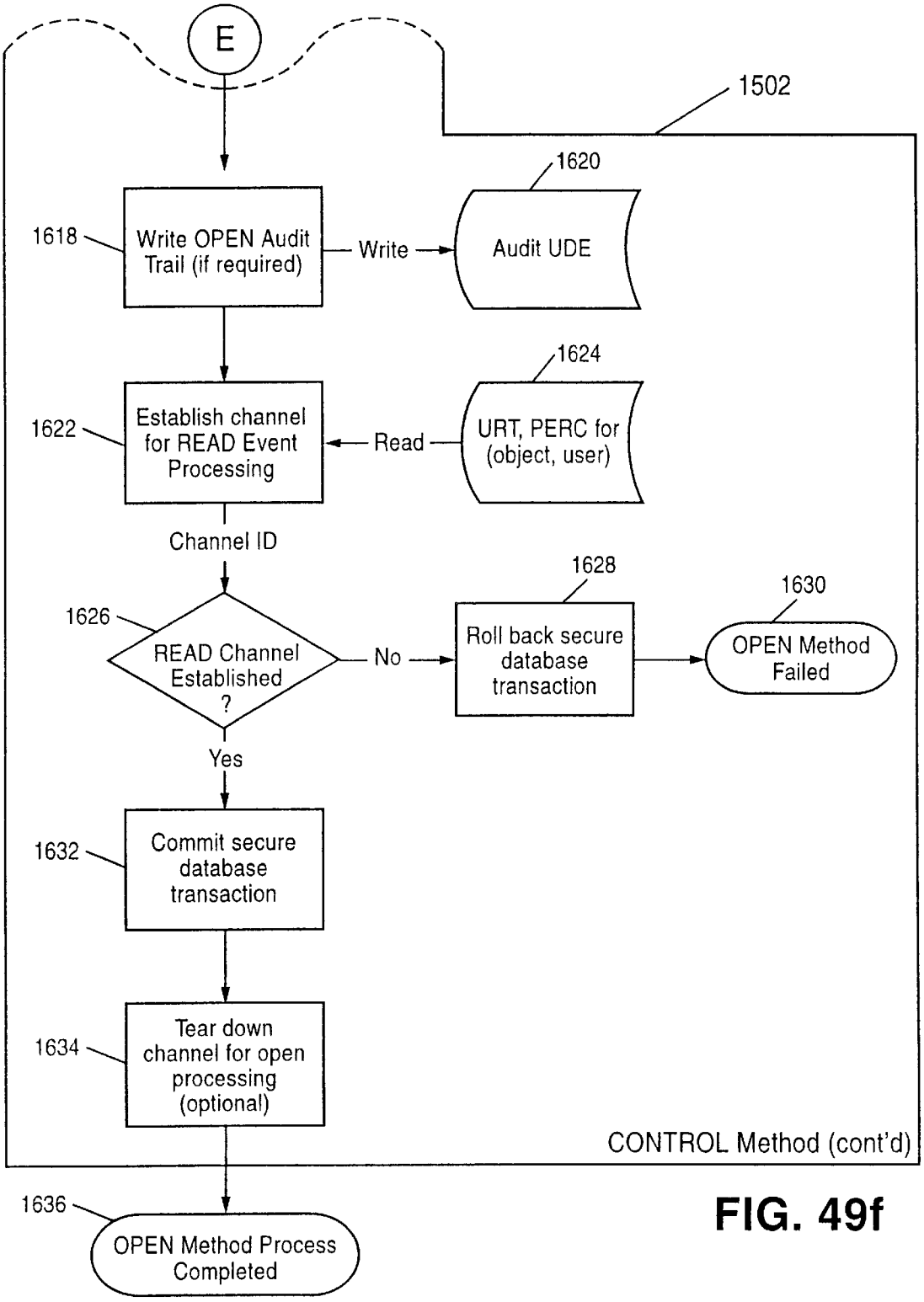

E
1502
1620
1618
Write OPEN Audit Trail (if required)
Write
Audit UDE
1624
1622
Establish channel for READ Event Processing
Read
URT, PERC for (object, user)
Channel ID
1628
1630
1626
READ Channel Established ?
No
Roll back secure database transaction
OPEN Method Failed
Yes
1632
Commit secure database transaction
1634
Tear down channel for open processing (optional)
CONTROL Method (cont'd)
1636
OPEN Method Process Completed

FIG. 49f

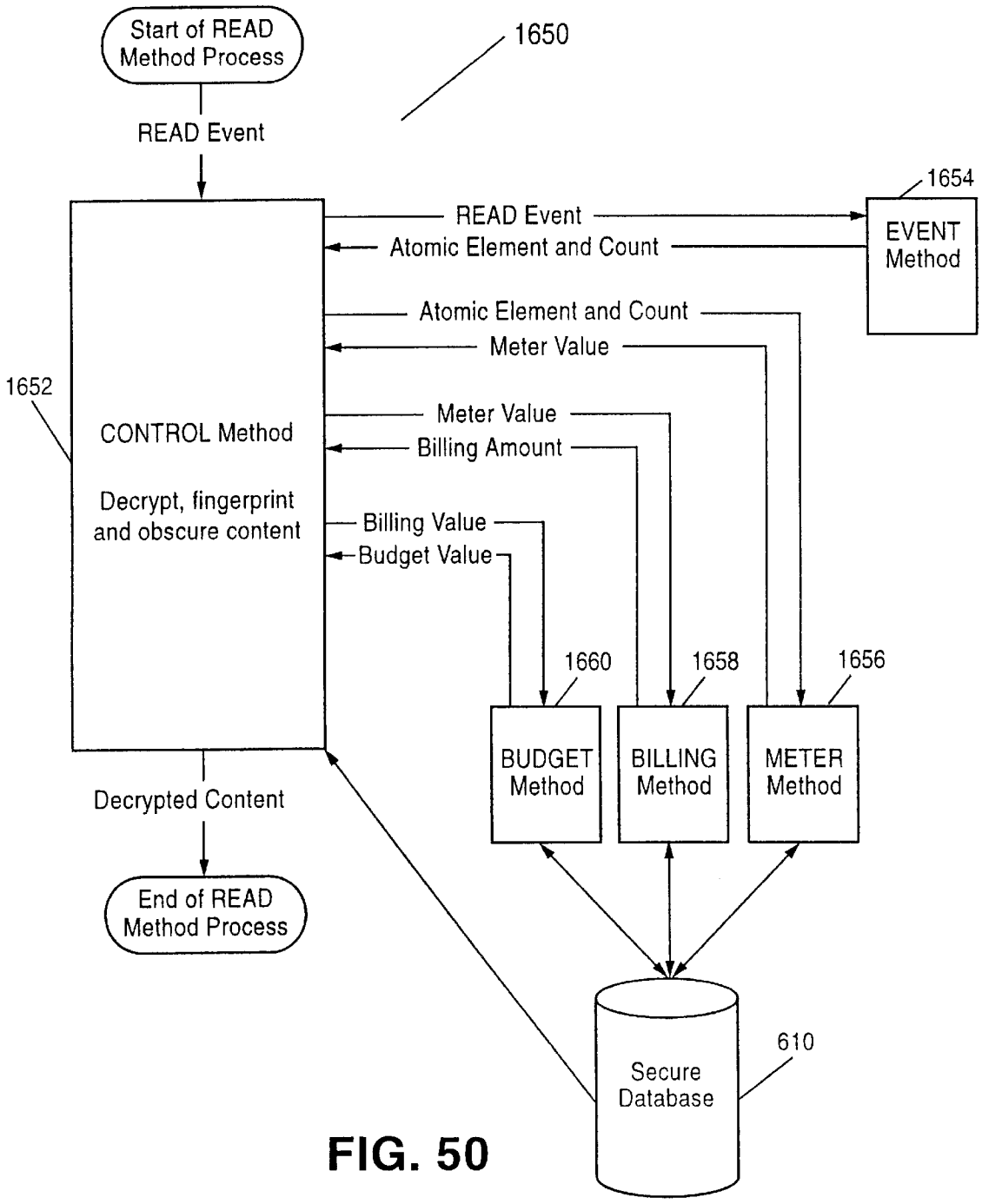

Start of READ Method Process
1650
READ Event
1654
READ Event
Atomic Element and Count
EVENT Method
Atomic Element and Count
Meter Value
1652
CONTROL Method
Decrypt, fingerprint and obscure content
Meter Value
Billing Amount
Billing Value
Budget Value
1660
1658
1656
BUDGET Method
BILLING Method
METER Method
Decrypted Content
End of READ Method Process
610
Secure Database

FIG. 50

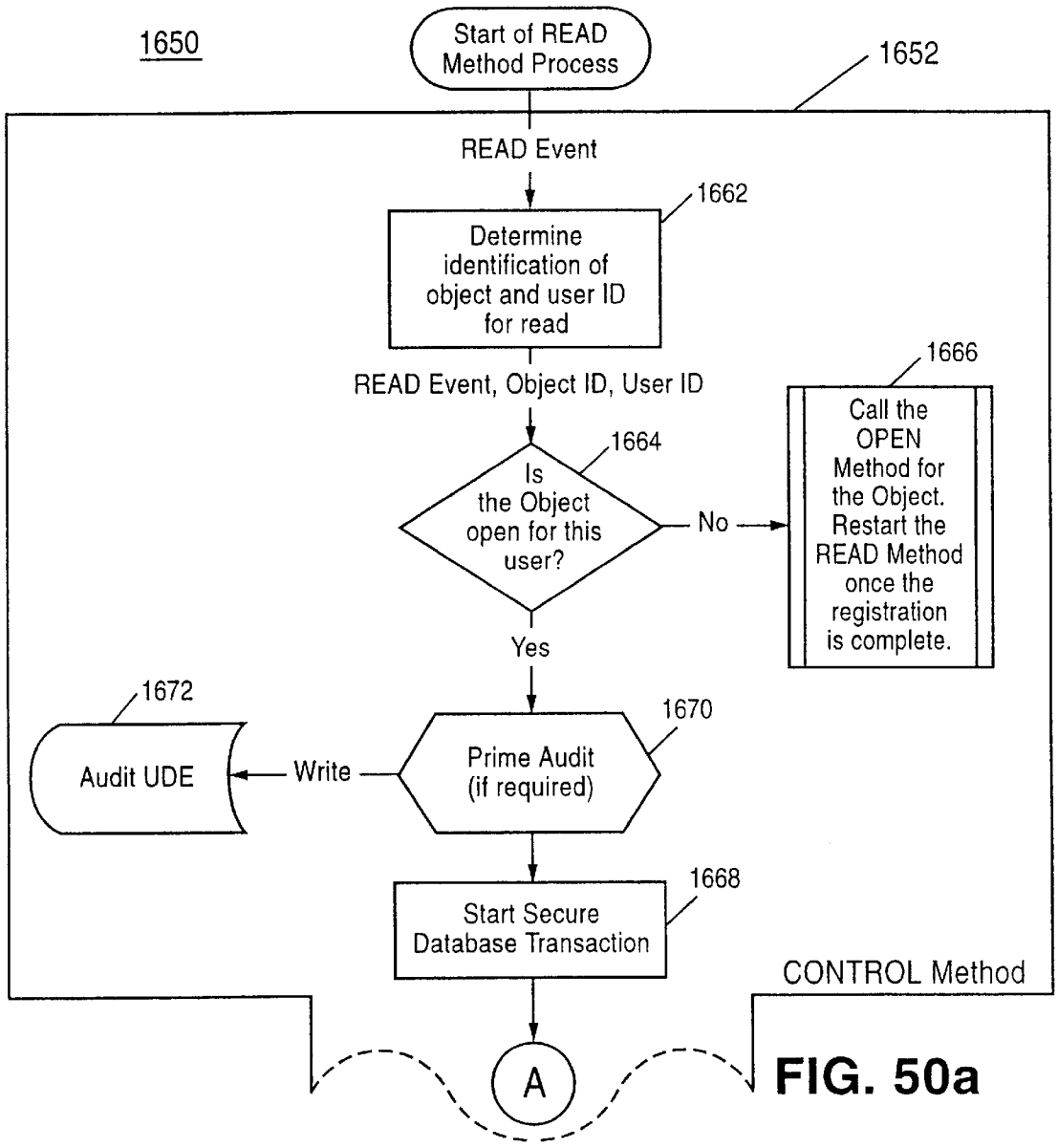

1650
Start of READ Method Process
1652
READ Event
1662
Determine identification of object and user ID for read
READ Event, Object ID, User ID
1666
Call the OPEN Method for the Object. Restart the READ Method once the registration is complete.
1664
Is the Object open for this user?
No
Yes
1672
Audit UDE
Write
1670
Prime Audit (if required)
1668
Start Secure Database Transaction
CONTROL Method
A

FIG. 50a

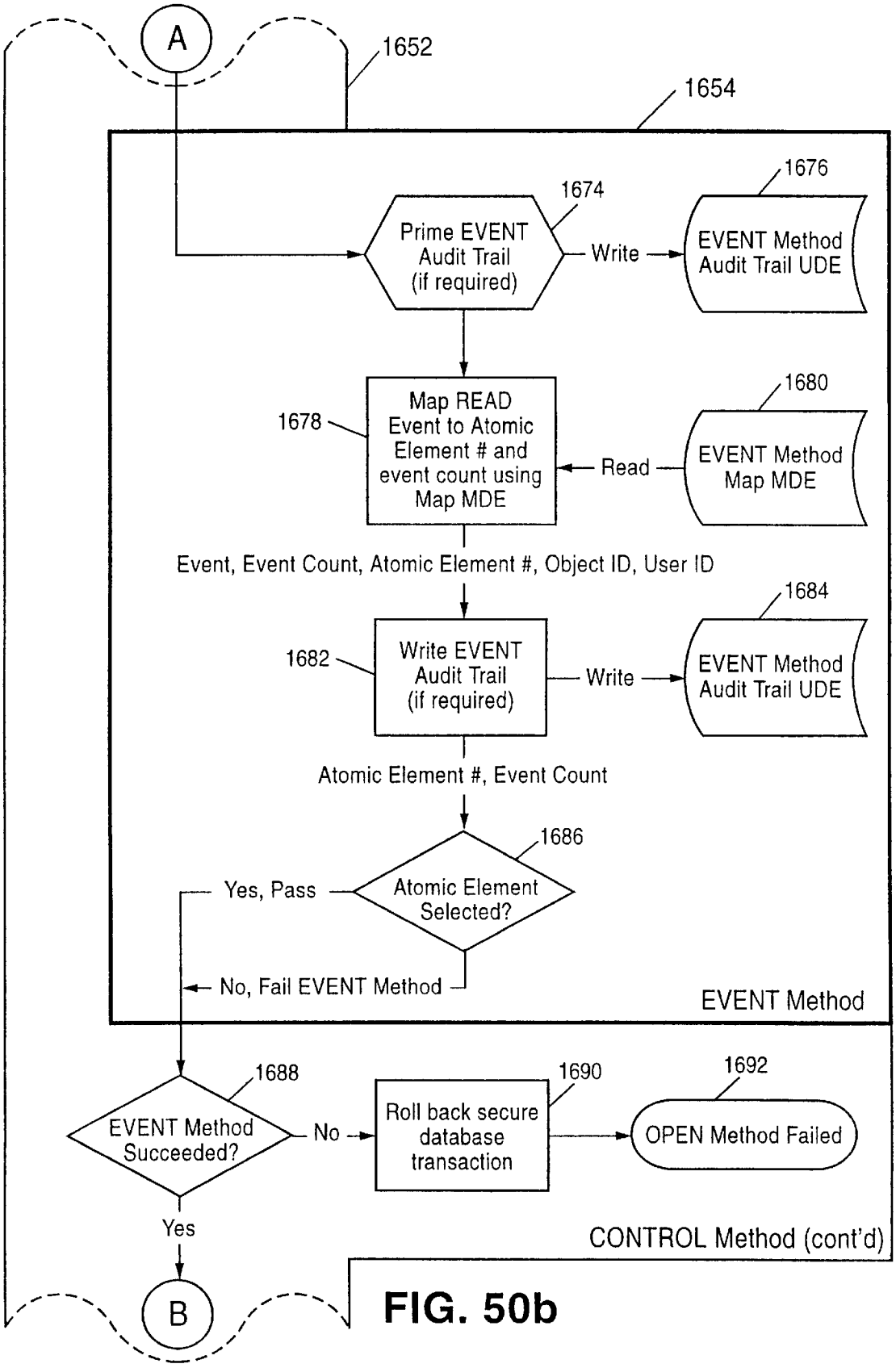

A
1652
1654
1676
1674
Prime EVENT Audit Trail (if required)
Write
EVENT Method Audit Trail UDE
1680
1678
Map READ Event to Atomic Element # and event count using Map MDE
Read
EVENT Method Map MDE
Event, Event Count, Atomic Element #, Object ID, User ID
1684
1682
Write EVENT Audit Trail (if required)
Write
EVENT Method Audit Trail UDE
Atomic Element #, Event Count
1686
Yes, Pass
Atomic Element Selected?
No, Fail EVENT Method
EVENT Method
1688
1690
1692
EVENT Method Succeeded?
No
Roll back secure database transaction
OPEN Method Failed
Yes
CONTROL Method (cont'd)
B

FIG. 50b

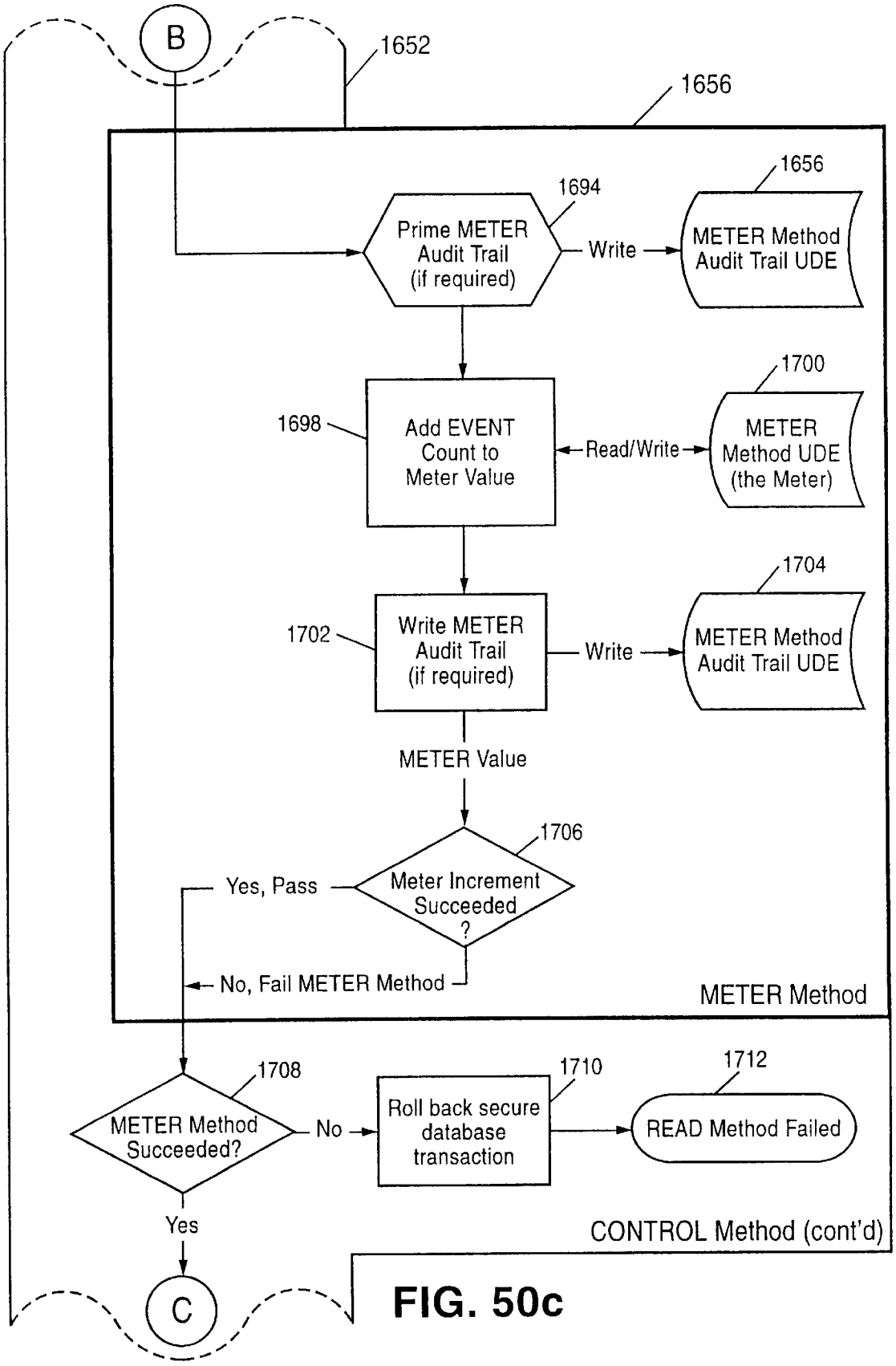

B
1652
1656
1656
1694
Prime METER Audit Trail (if required)
Write
METER Method Audit Trail UDE
1700
1698
Add EVENT Count to Meter Value
Read/Write
METER Method UDE (the Meter)
1704
1702
Write METER Audit Trail (if required)
Write
METER Method Audit Trail UDE
METER Value
1706
Meter Increment Succeeded ?
Yes, Pass
No, Fail METER Method
METER Method
1708
METER Method Succeeded?
No
1710
Roll back secure database transaction
1712
READ Method Failed
Yes
CONTROL Method (cont'd)
C

FIG. 50c

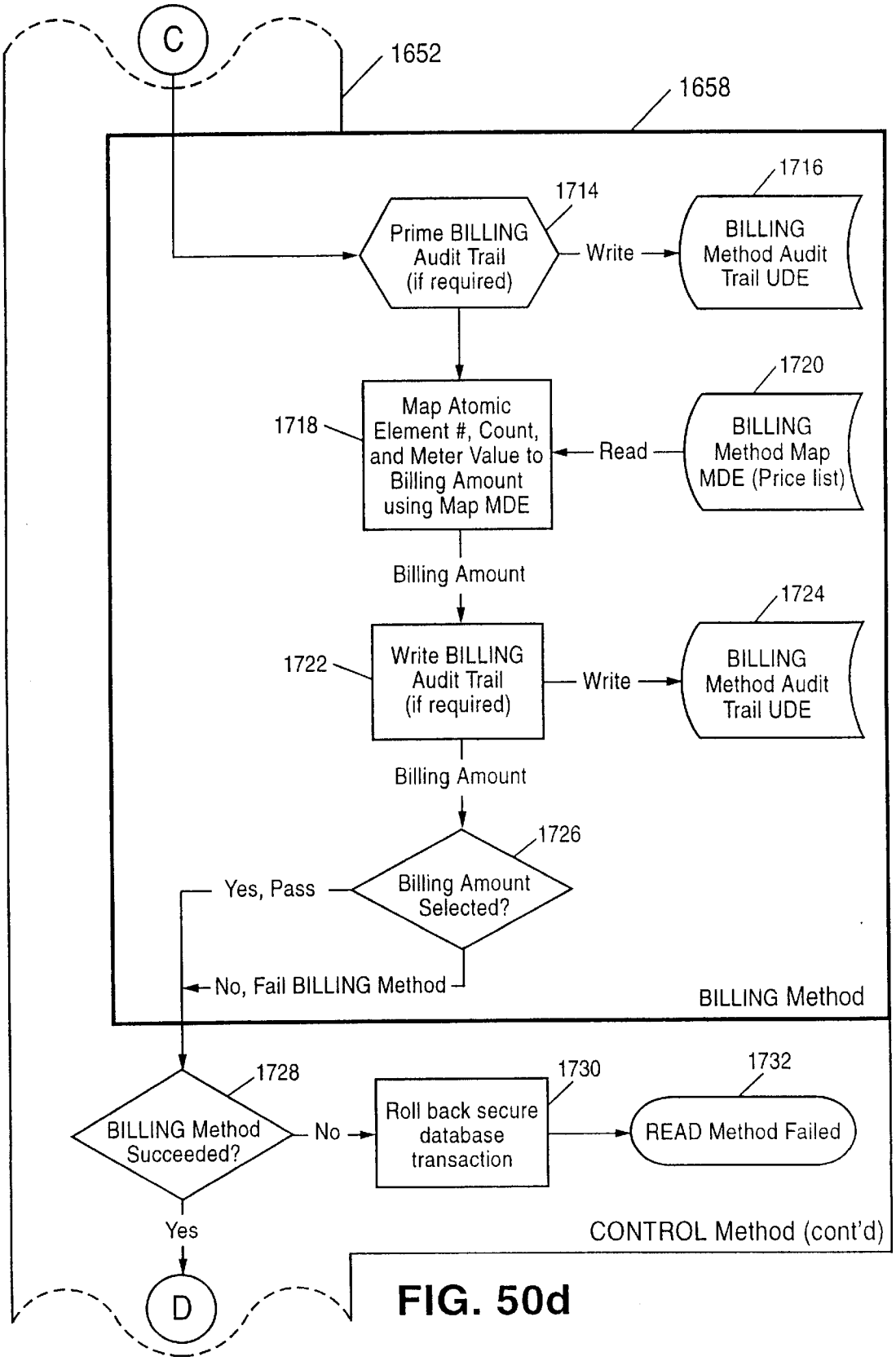

C
1652
1658
1714
Prime BILLING Audit Trail (if required)
Write
1716
BILLING Method Audit Trail UDE
1718
Map Atomic Element #, Count, and Meter Value to Billing Amount using Map MDE
Read
1720
BILLING Method Map MDE (Price list)
Billing Amount
1722
Write BILLING Audit Trail (if required)
Write
1724
BILLING Method Audit Trail UDE
Billing Amount
1726
Billing Amount Selected?
Yes, Pass
No, Fail BILLING Method
BILLING Method
1728
BILLING Method Succeeded?
No
1730
Roll back secure database transaction
1732
READ Method Failed
Yes
CONTROL Method (cont'd)
D

FIG. 50d

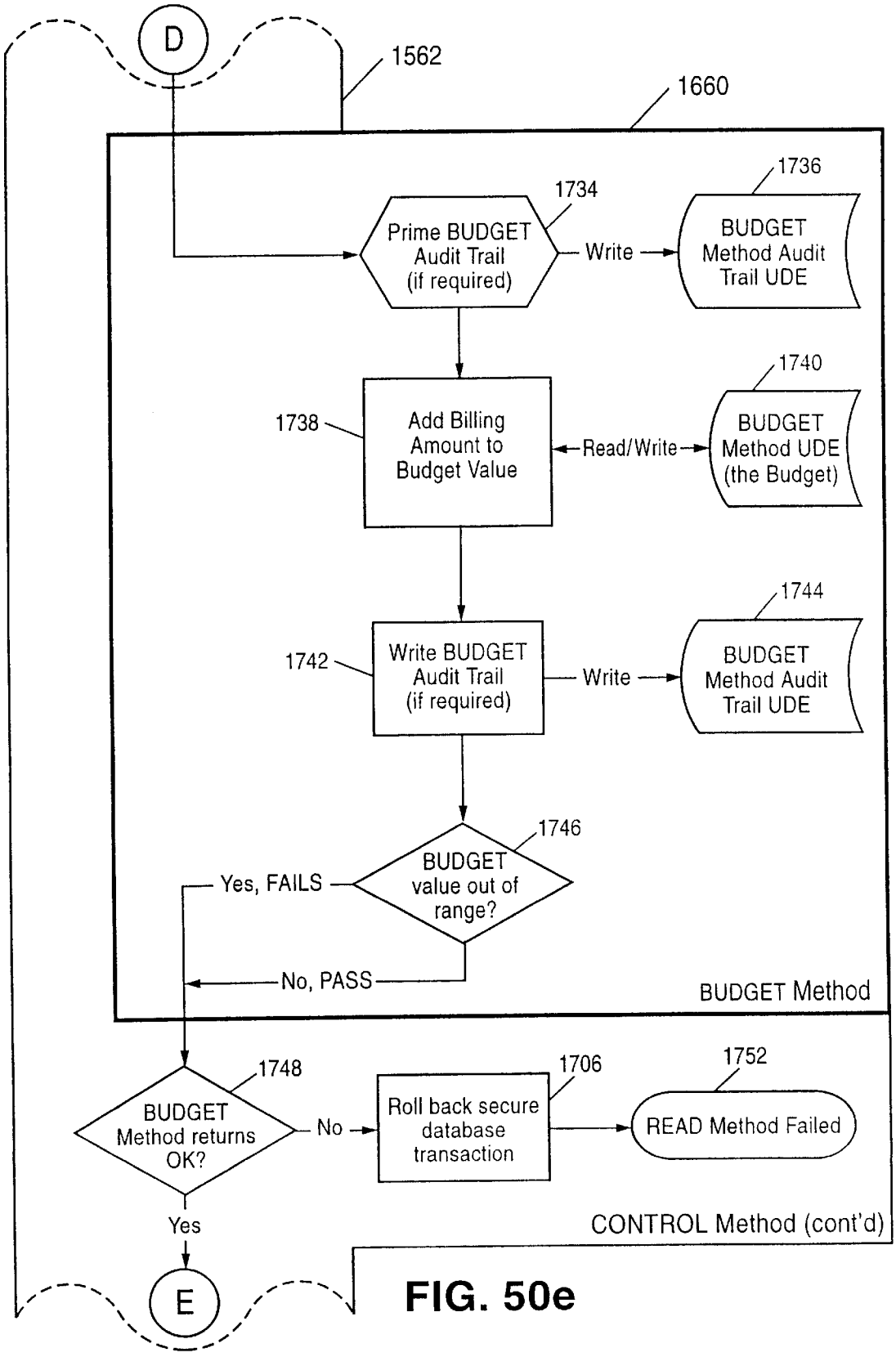
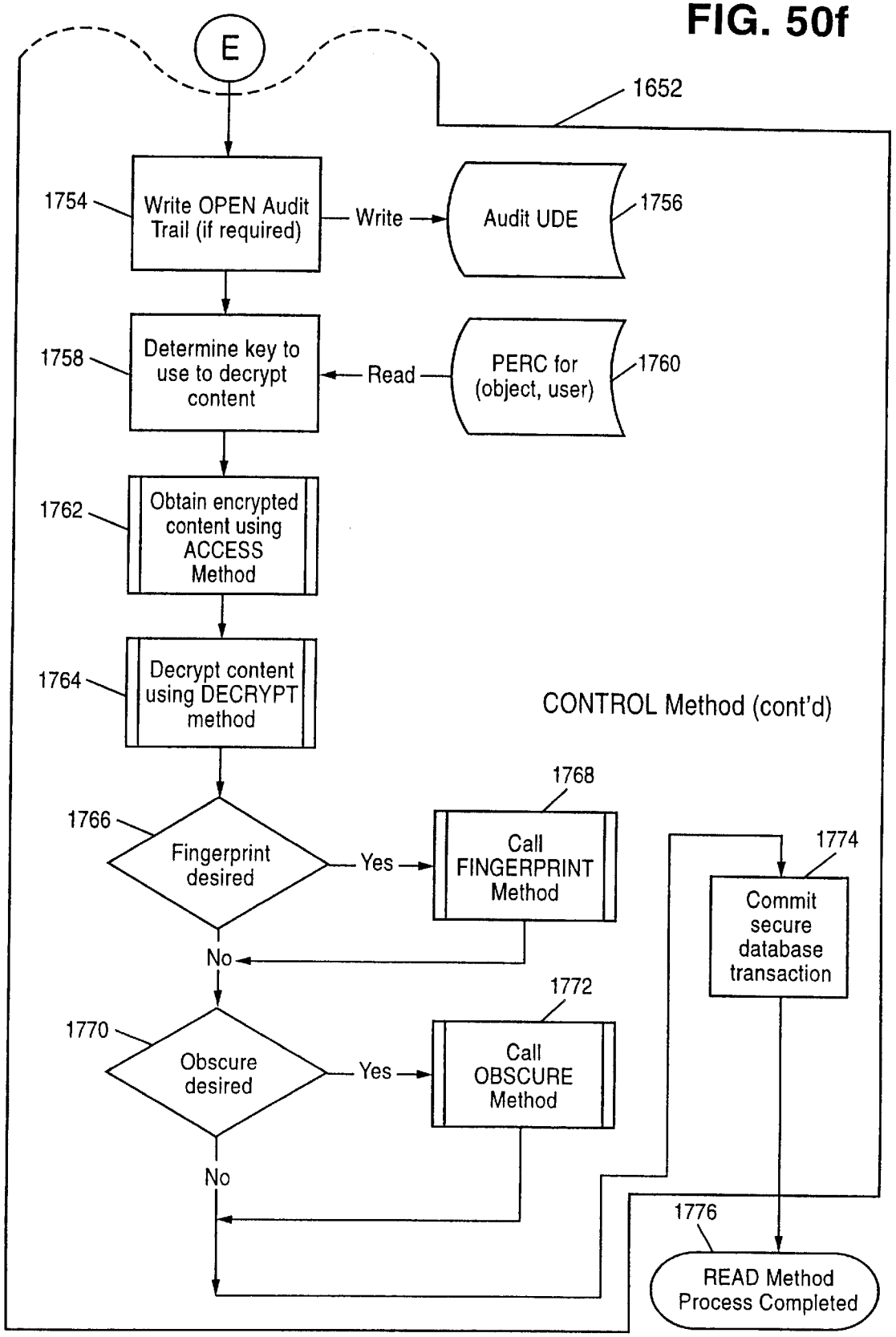

D
1562
1660
1734
Prime BUDGET Audit Trail (if required)
Write
1736
BUDGET Method Audit Trail UDE
1738
Add Billing Amount to Budget Value
Read/Write
1740
BUDGET Method UDE (the Budget)
1742
Write BUDGET Audit Trail (if required)
Write
1744
BUDGET Method Audit Trail UDE
1746
BUDGET value out of range?
Yes, FAILS
No, PASS
BUDGET Method
1748
BUDGET Method returns OK?
No
1706
Roll back secure database transaction
1752
READ Method Failed
Yes
CONTROL Method (cont'd)
E

FIG. 50e

FIG. 50f


E
1652
1754
Write OPEN Audit Trail (if required)
Write
Audit UDE
1756
1758
Determine key to use to decrypt content
Read
PERC for (object, user)
1760
1762
Obtain encrypted content using ACCESS Method
1764
Decrypt content using DECRYPT method
CONTROL Method (cont'd)
1768
1766
Fingerprint desired
Yes
Call FINGERPRINT Method
1774
Commit secure database transaction
No
1772
1770
Obscure desired
Yes
Call OBSCURE Method
No
1776
READ Method Process Completed

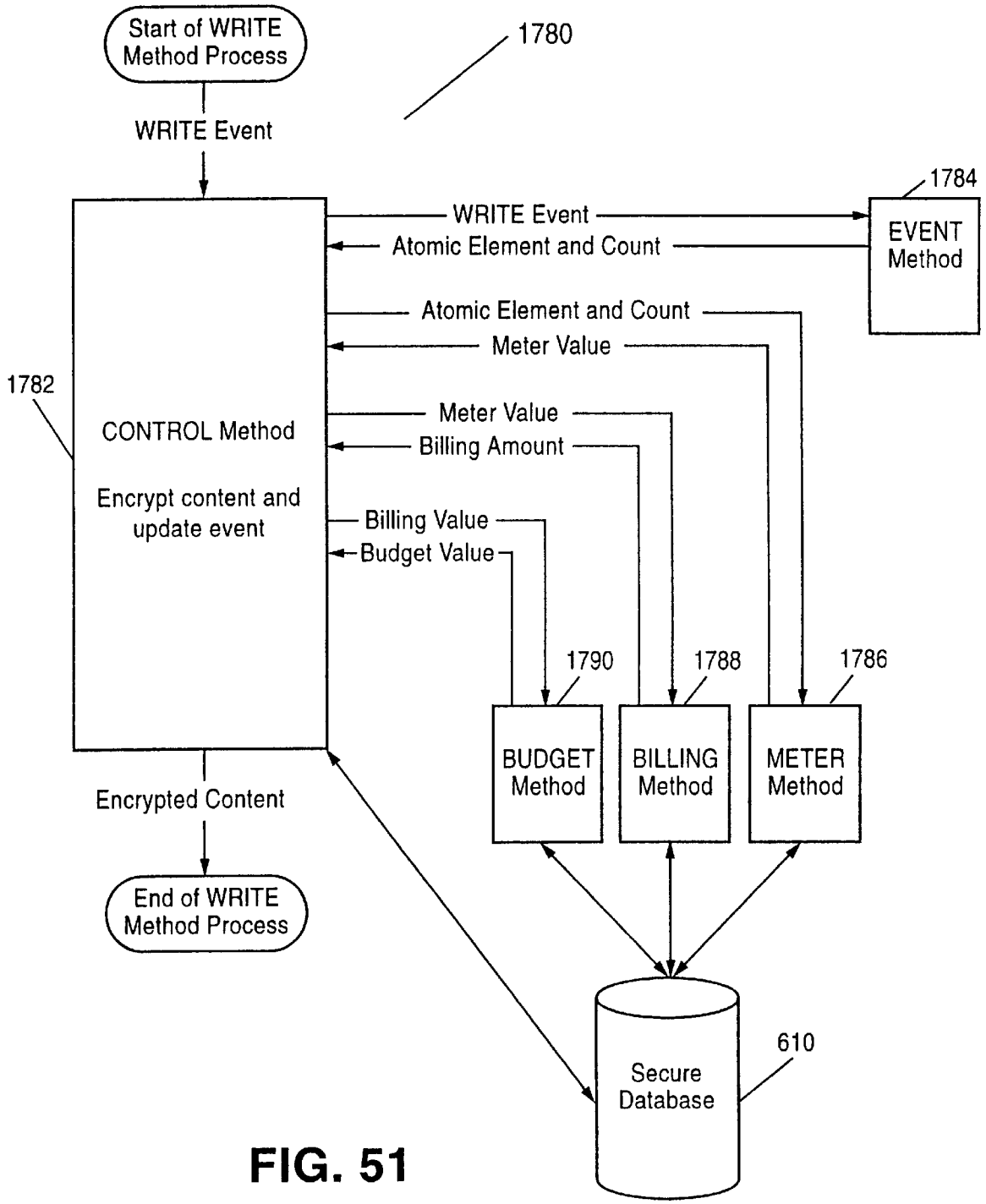

Start of WRITE Method Process
1780
WRITE Event
1784
WRITE Event
Atomic Element and Count
EVENT Method
Atomic Element and Count
Meter Value
1782
CONTROL Method
Encrypt content and update event
Meter Value
Billing Amount
Billing Value
Budget Value
1790
1788
1786
BUDGET Method
BILLING Method
METER Method
Encrypted Content
End of WRITE Method Process
610
Secure Database

FIG. 51

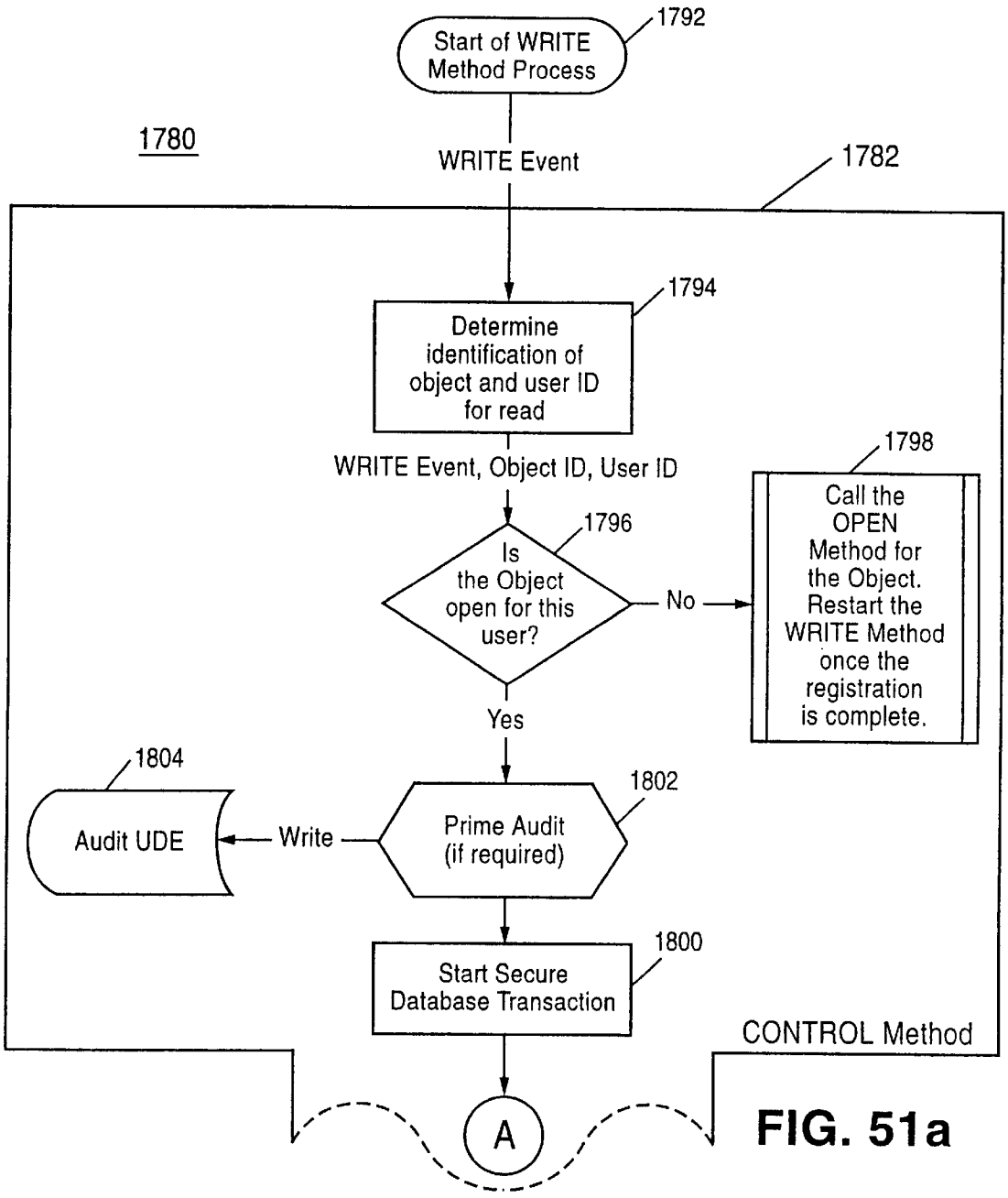

1780
Start of WRITE Method Process
1792
WRITE Event
1782
Determine identification of object and user ID for read
1794
WRITE Event, Object ID, User ID
Is the Object open for this user?
1796
No
Call the OPEN Method for the Object. Restart the WRITE Method once the registration is complete.
1798
Yes
Audit UDE
1804
Write
Prime Audit (if required)
1802
Start Secure Database Transaction
1800
CONTROL Method
A

FIG. 51a

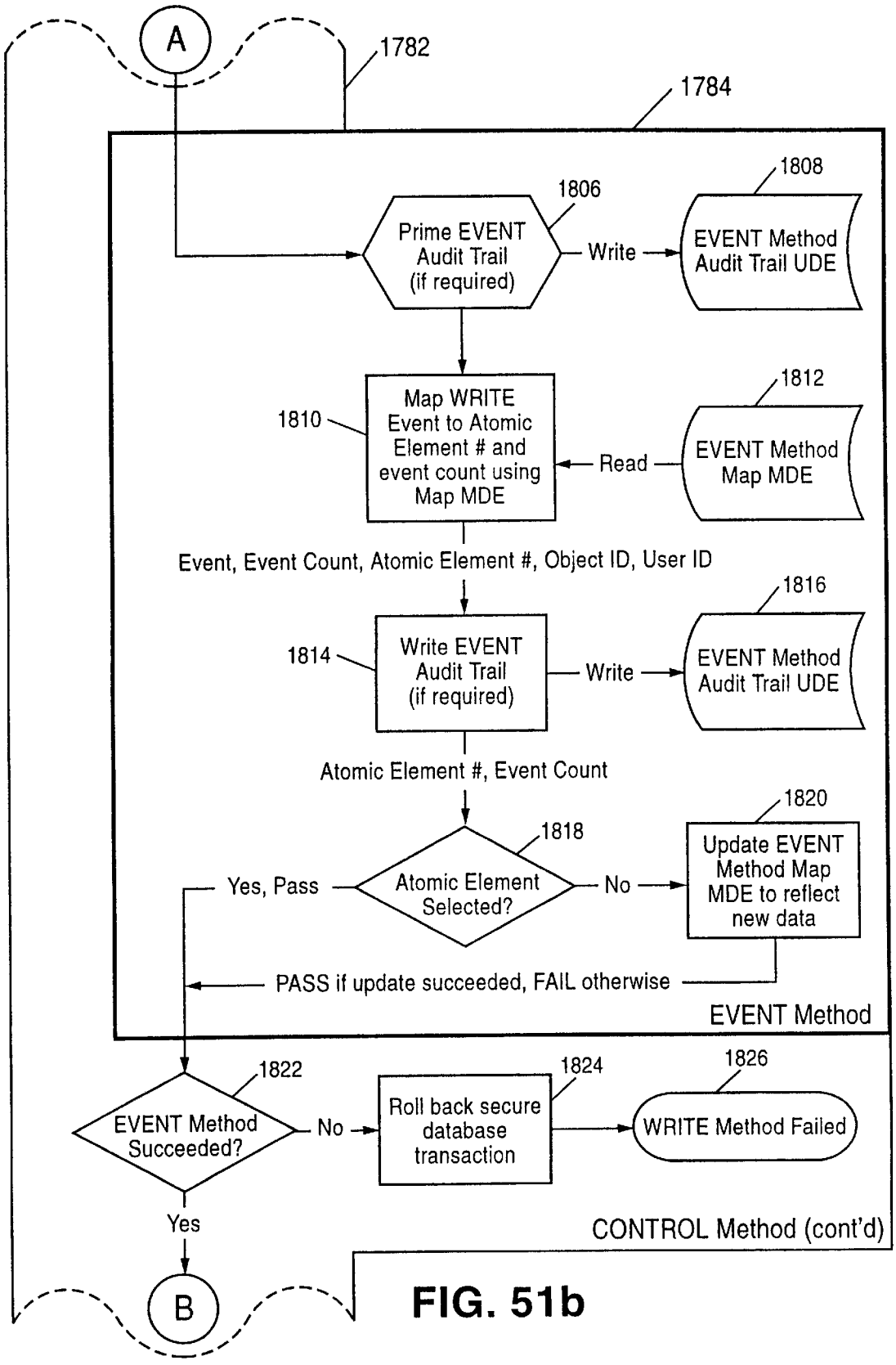

A
1782
1784
1806
Prime EVENT Audit Trail (if required)
Write
1808
EVENT Method Audit Trail UDE
1810
Map WRITE Event to Atomic Element # and event count using Map MDE
Read
1812
EVENT Method Map MDE
Event, Event Count, Atomic Element #, Object ID, User ID
1814
Write EVENT Audit Trail (if required)
Write
1816
EVENT Method Audit Trail UDE
Atomic Element #, Event Count
1818
Atomic Element Selected?
Yes, Pass
No
1820
Update EVENT Method Map MDE to reflect new data
PASS if update succeeded, FAIL otherwise
EVENT Method
1822
EVENT Method Succeeded?
No
1824
Roll back secure database transaction
1826
WRITE Method Failed
Yes
CONTROL Method (cont'd)
B

FIG. 51b

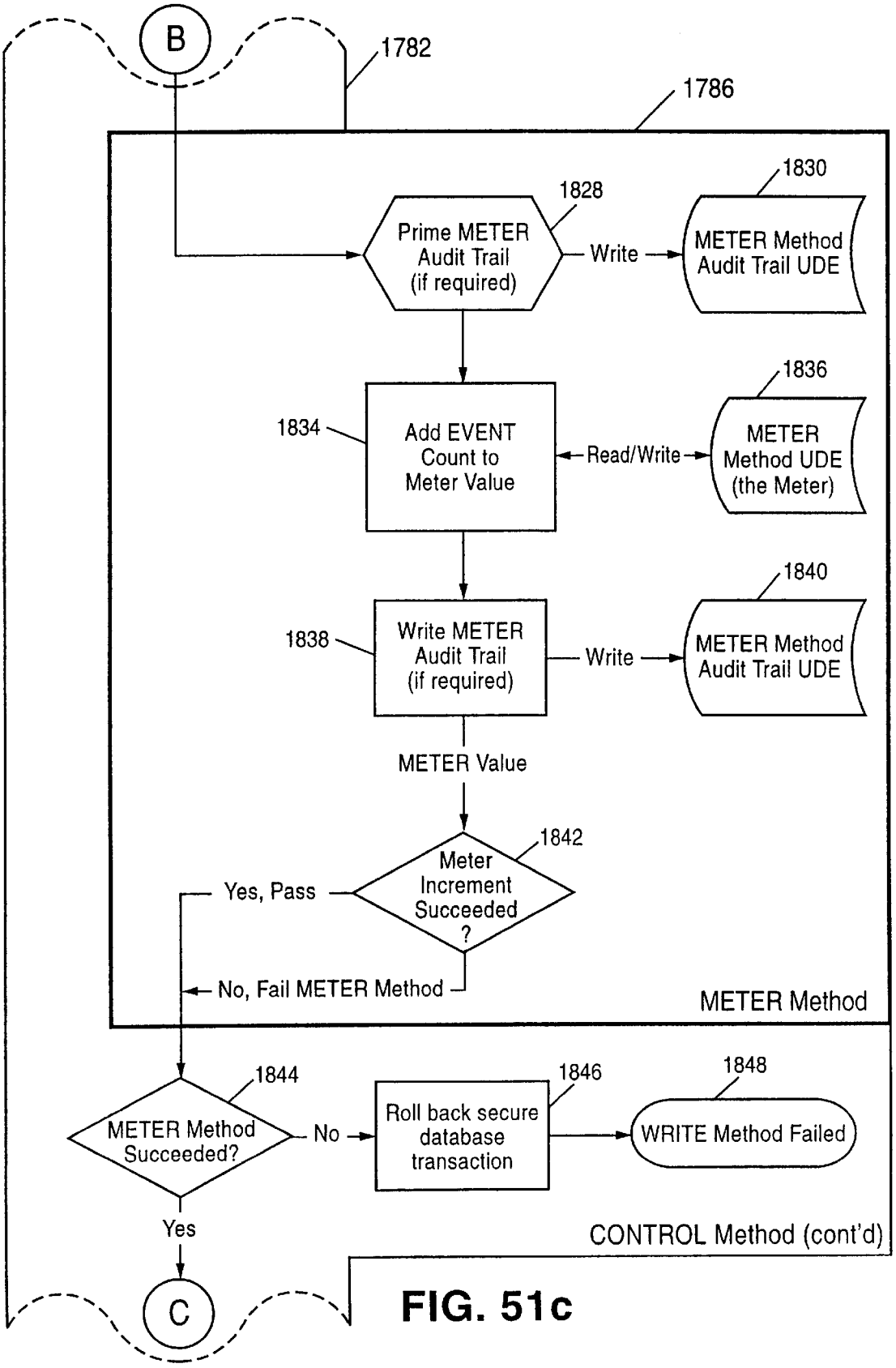

B
1782
1786
1828
Prime METER Audit Trail (if required)
Write
1830
METER Method Audit Trail UDE
1834
Add EVENT Count to Meter Value
Read/Write
1836
METER Method UDE (the Meter)
1838
Write METER Audit Trail (if required)
Write
1840
METER Method Audit Trail UDE
METER Value
1842
Meter Increment Succeeded ?
Yes, Pass
No, Fail METER Method
METER Method
1844
METER Method Succeeded?
No
1846
Roll back secure database transaction
1848
WRITE Method Failed
Yes
CONTROL Method (cont'd)
C

FIG. 51c

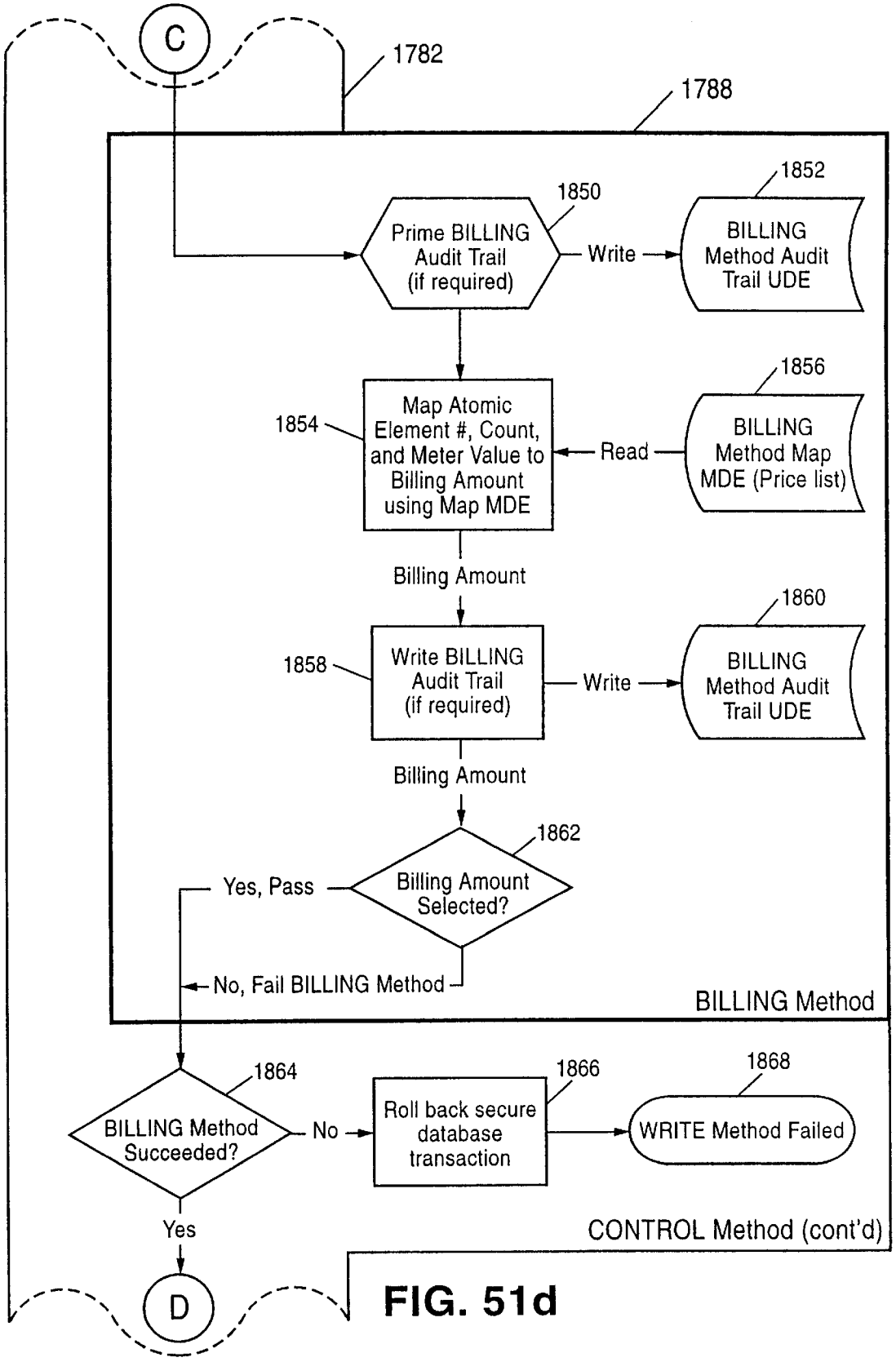

C
1782
1788
1850
Prime BILLING Audit Trail (if required)
Write
1852
BILLING Method Audit Trail UDE
1854
Map Atomic Element #, Count, and Meter Value to Billing Amount using Map MDE
Read
1856
BILLING Method Map MDE (Price list)
Billing Amount
1858
Write BILLING Audit Trail (if required)
Write
1860
BILLING Method Audit Trail UDE
Billing Amount
1862
Billing Amount Selected?
Yes, Pass
No, Fail BILLING Method
BILLING Method
1864
BILLING Method Succeeded?
No
1866
Roll back secure database transaction
1868
WRITE Method Failed
Yes
CONTROL Method (cont'd)
D

FIG. 51d

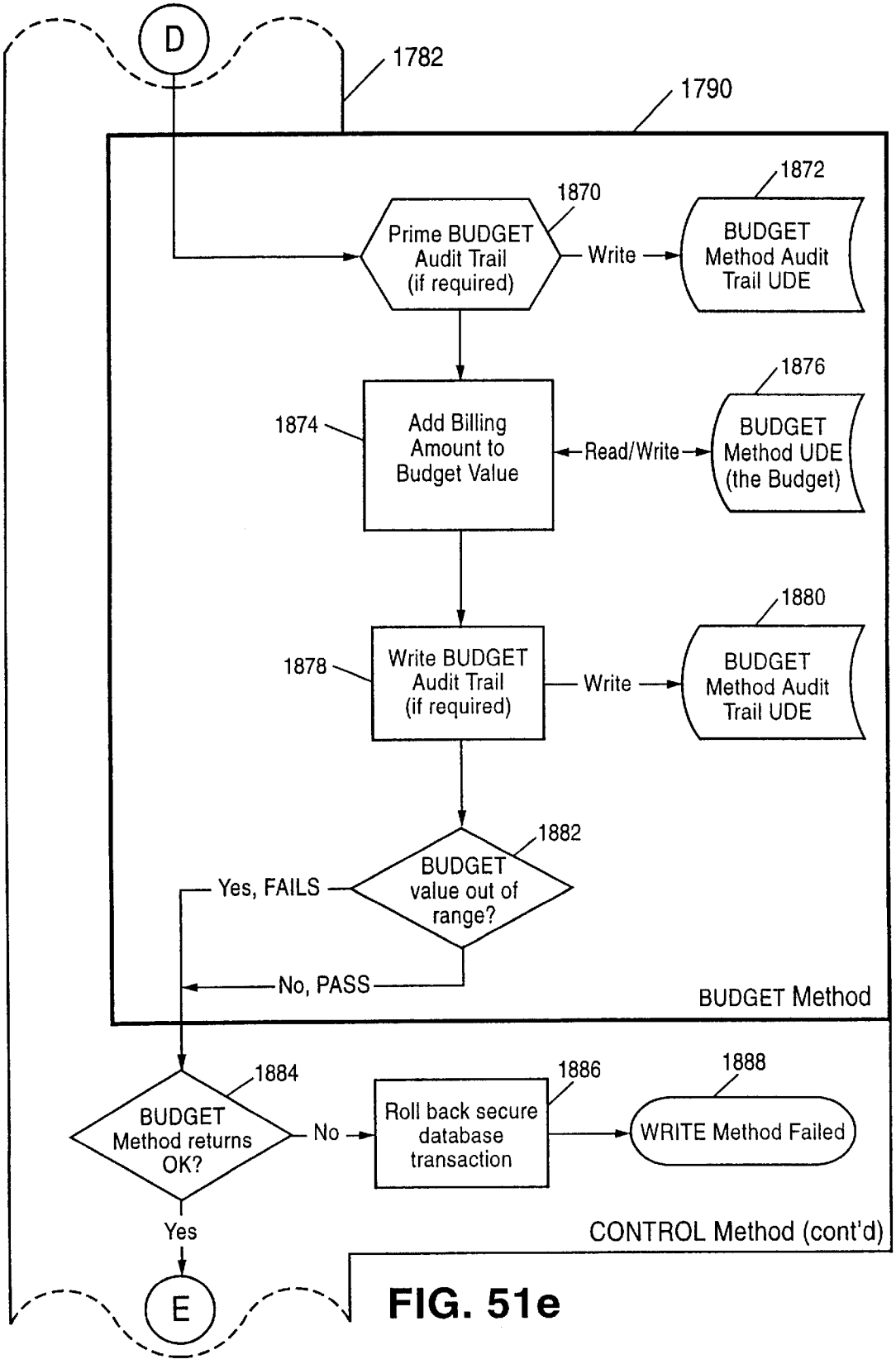

D
1782
1790
1870
Prime BUDGET Audit Trail (if required)
Write
1872
BUDGET Method Audit Trail UDE
1874
Add Billing Amount to Budget Value
Read/Write
1876
BUDGET Method UDE (the Budget)
1878
Write BUDGET Audit Trail (if required)
Write
1880
BUDGET Method Audit Trail UDE
1882
BUDGET value out of range?
Yes, FAILS
No, PASS
BUDGET Method
1884
BUDGET Method returns OK?
No
1886
Roll back secure database transaction
1888
WRITE Method Failed
Yes
CONTROL Method (cont'd)
E

FIG. 51e

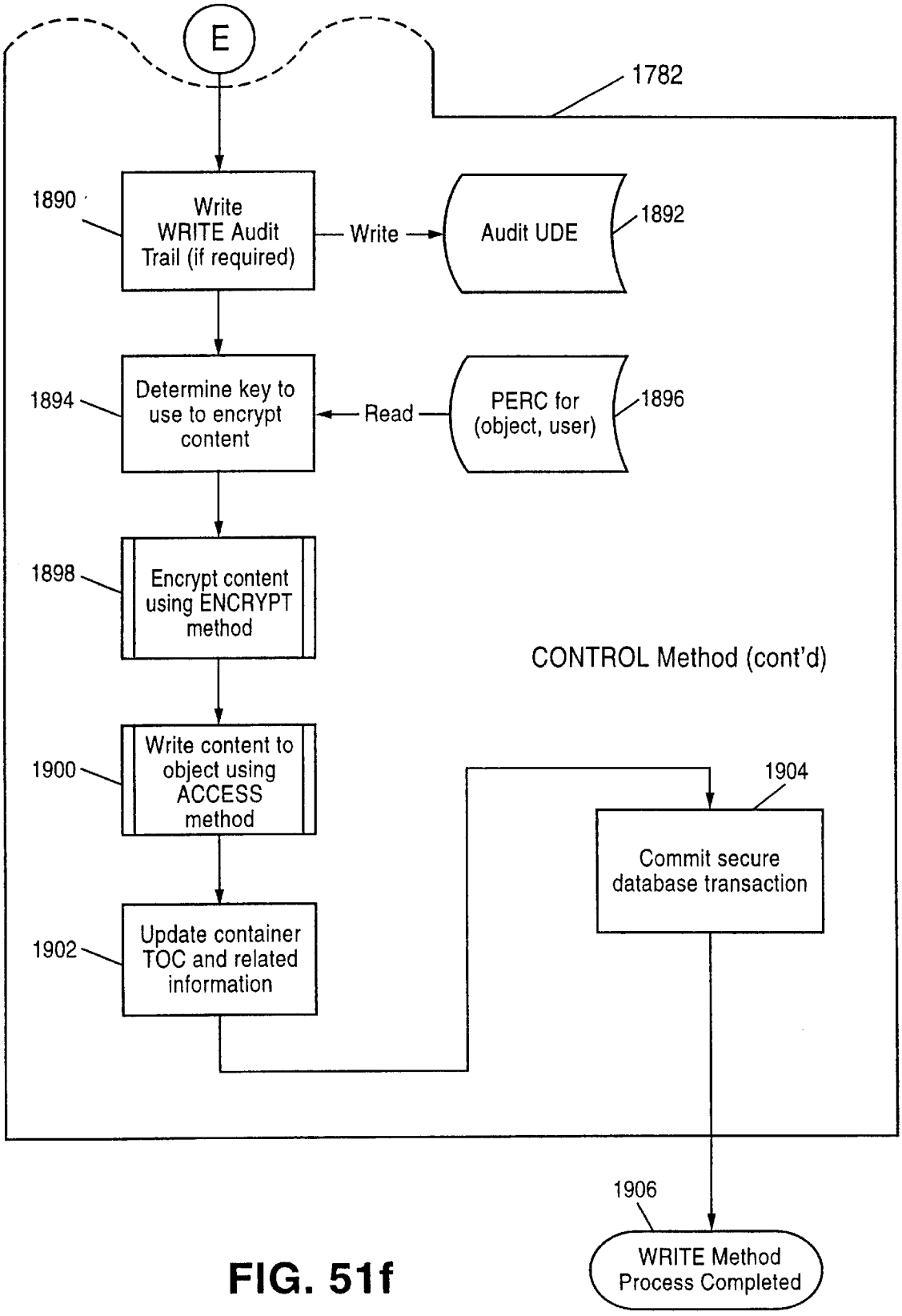

E
1782
1890
Write WRITE Audit Trail (if required)
Write
Audit UDE
1892
1894
Determine key to use to encrypt content
Read
PERC for (object, user)
1896
1898
Encrypt content using ENCRYPT method
CONTROL Method (cont'd)
1900
Write content to object using ACCESS method
1904
Commit secure database transaction
1902
Update container TOC and related information
1906
WRITE Method Process Completed

FIG. 51f

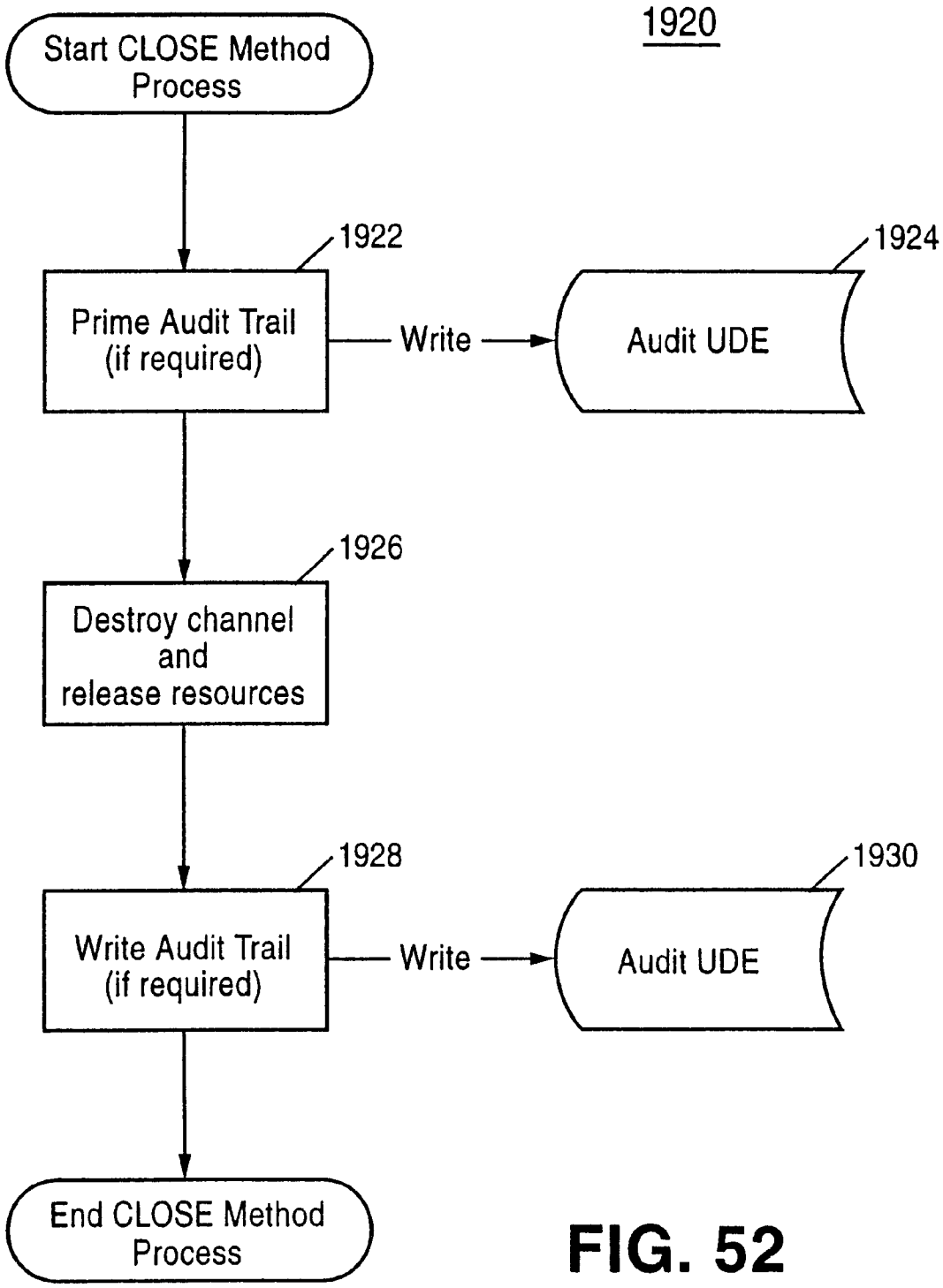

1920
Start CLOSE Method Process
1922
Prime Audit Trail (if required)
Write
1924
Audit UDE
1926
Destroy channel and release resources
1928
Write Audit Trail (if required)
Write
1930
Audit UDE
End CLOSE Method Process

FIG. 52

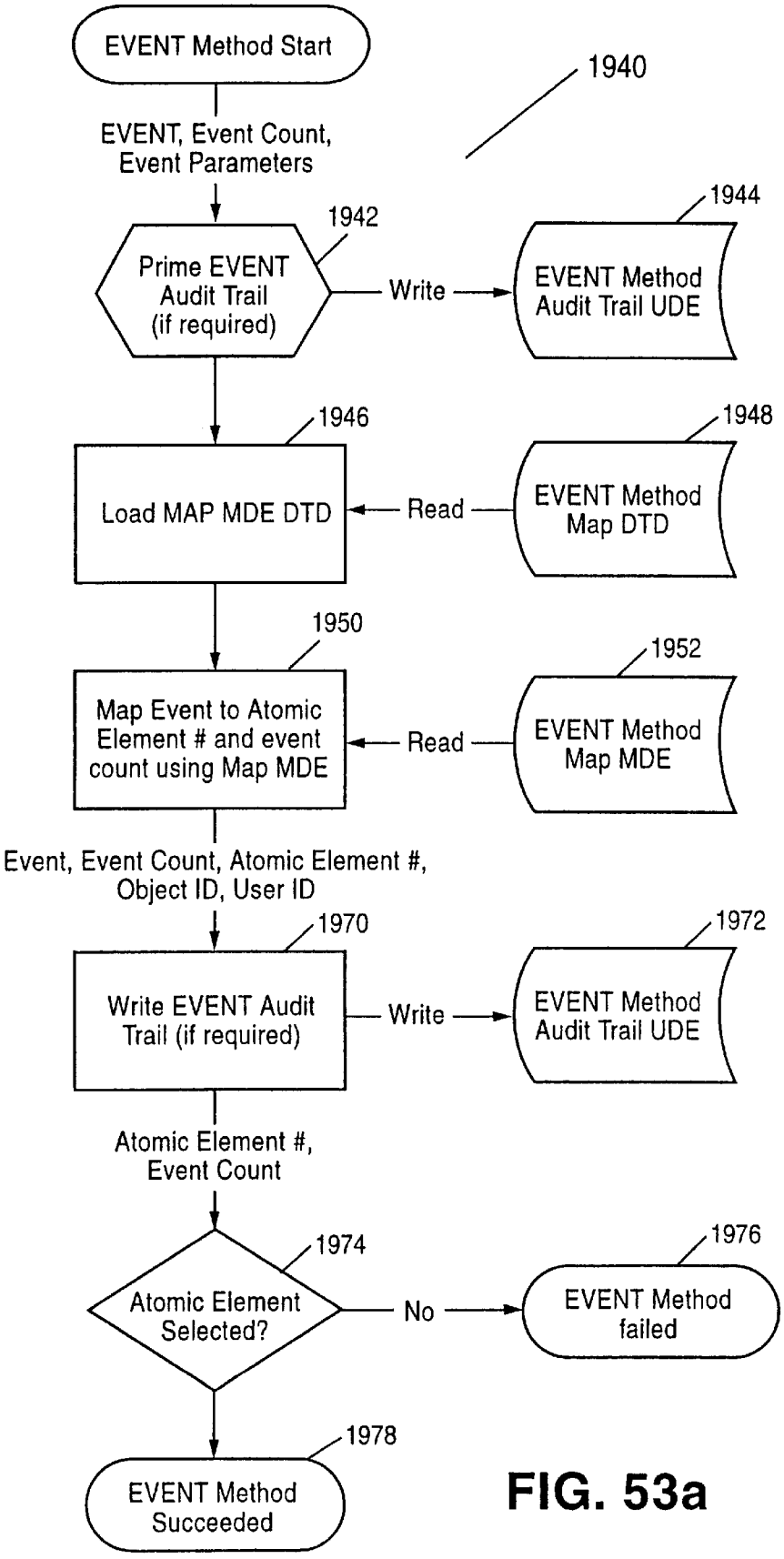

EVENT Method Start
1940
EVENT, Event Count, Event Parameters
1942
Prime EVENT Audit Trail (if required)
Write
1944
EVENT Method Audit Trail UDE
1946
Load MAP MDE DTD
Read
1948
EVENT Method Map DTD
1950
Map Event to Atomic Element # and event count using Map MDE
Read
1952
EVENT Method Map MDE
Event, Event Count, Atomic Element #, Object ID, User ID
1970
Write EVENT Audit Trail (if required)
Write
1972
EVENT Method Audit Trail UDE
Atomic Element #, Event Count
1974
Atomic Element Selected?
No
1976
EVENT Method failed
1978
EVENT Method Succeeded

FIG. 53a

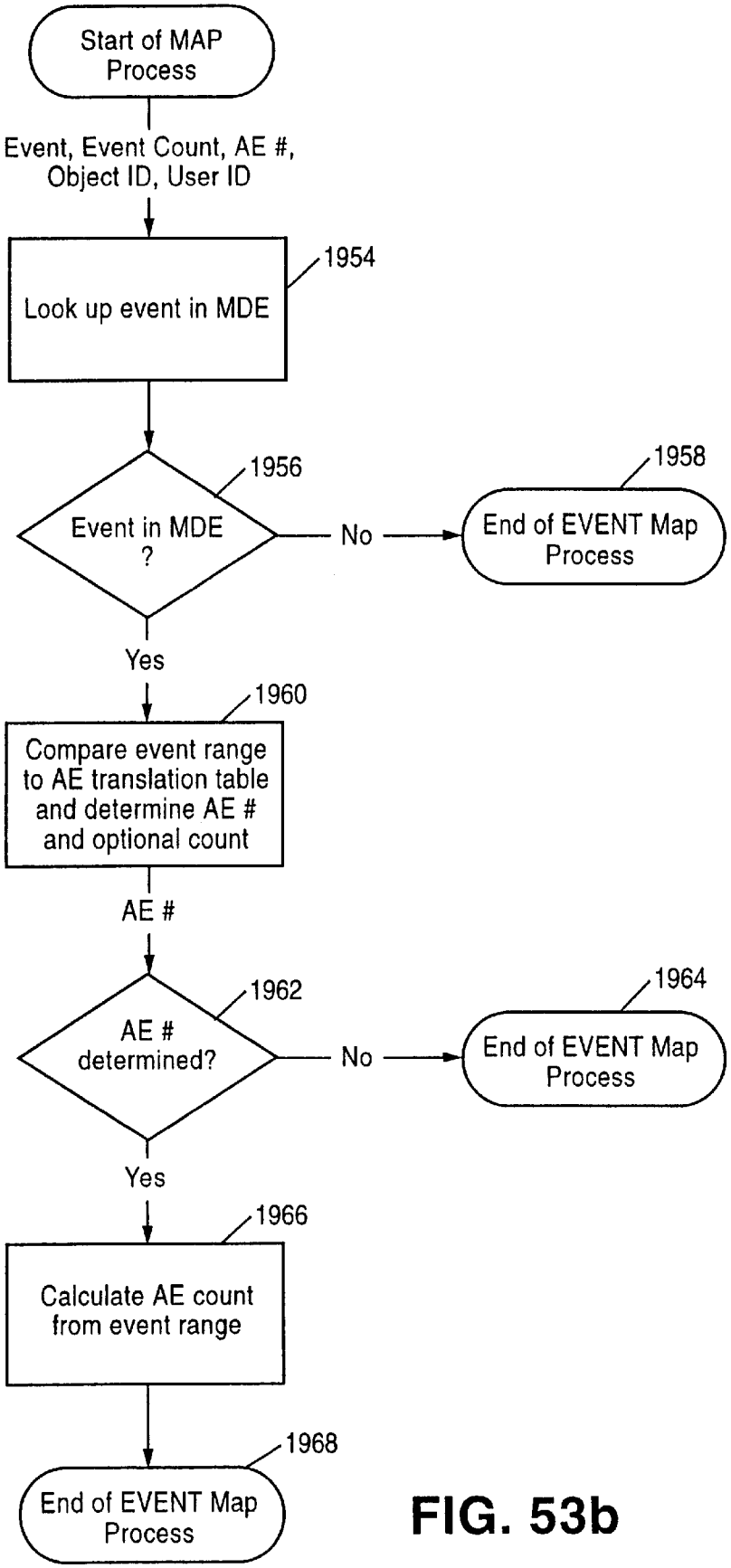

Start of MAP Process
Event, Event Count, AE #, Object ID, User ID
Look up event in MDE
1954
Event in MDE ?
1956
No
End of EVENT Map Process
1958
Yes
Compare event range to AE translation table and determine AE # and optional count
1960
AE #
AE # determined?
1962
No
End of EVENT Map Process
1964
Yes
Calculate AE count from event range
1966
End of EVENT Map Process
1968

FIG. 53b

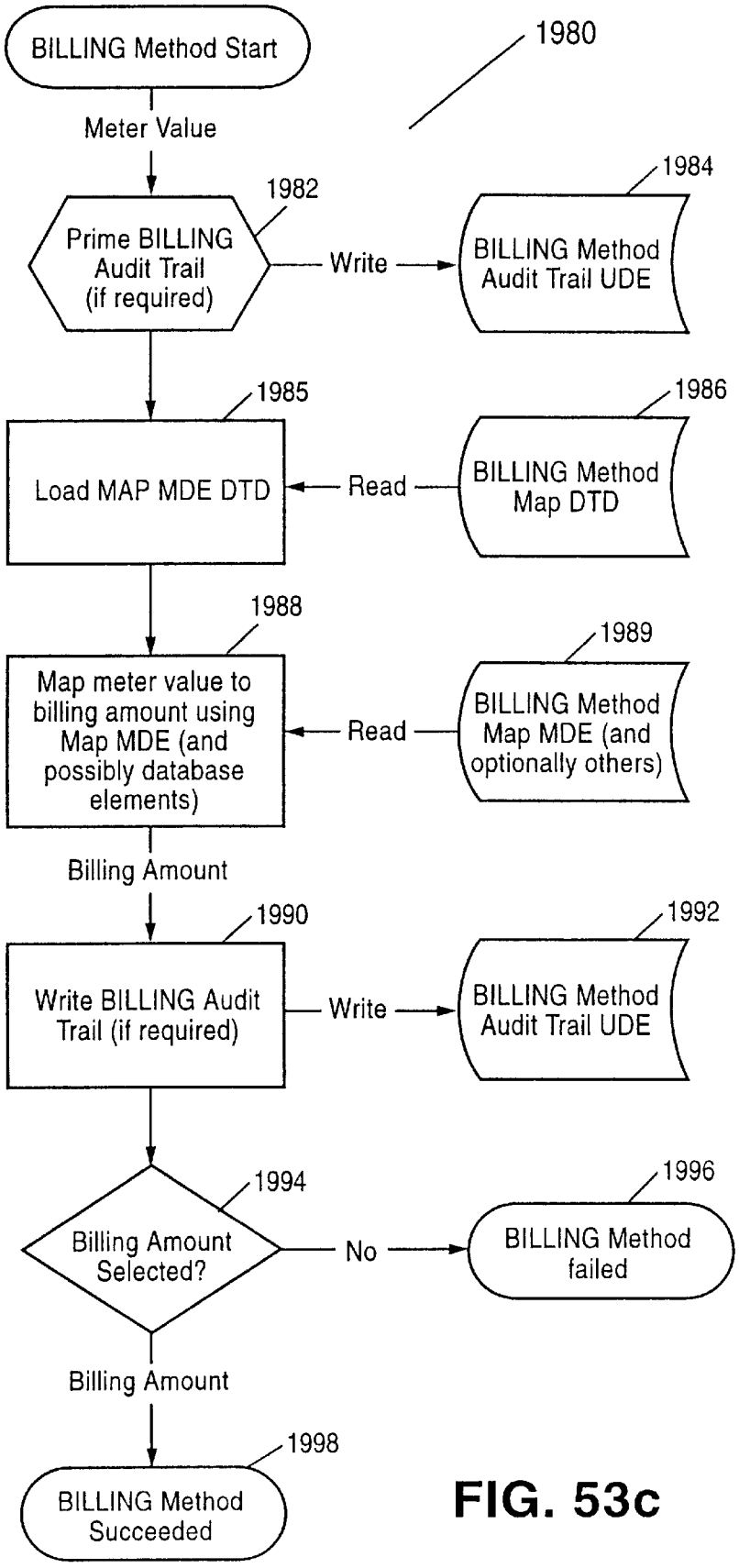

BILLING Method Start
1980
Meter Value
1982
Prime BILLING Audit Trail (if required)
Write
1984
BILLING Method Audit Trail UDE
1985
Load MAP MDE DTD
Read
1986
BILLING Method Map DTD
1988
Map meter value to billing amount using Map MDE (and possibly database elements)
Read
1989
BILLING Method Map MDE (and optionally others)
Billing Amount
1990
Write BILLING Audit Trail (if required)
Write
1992
BILLING Method Audit Trail UDE
1994
Billing Amount Selected?
No
1996
BILLING Method failed
Billing Amount
1998
BILLING Method Succeeded

FIG. 53c

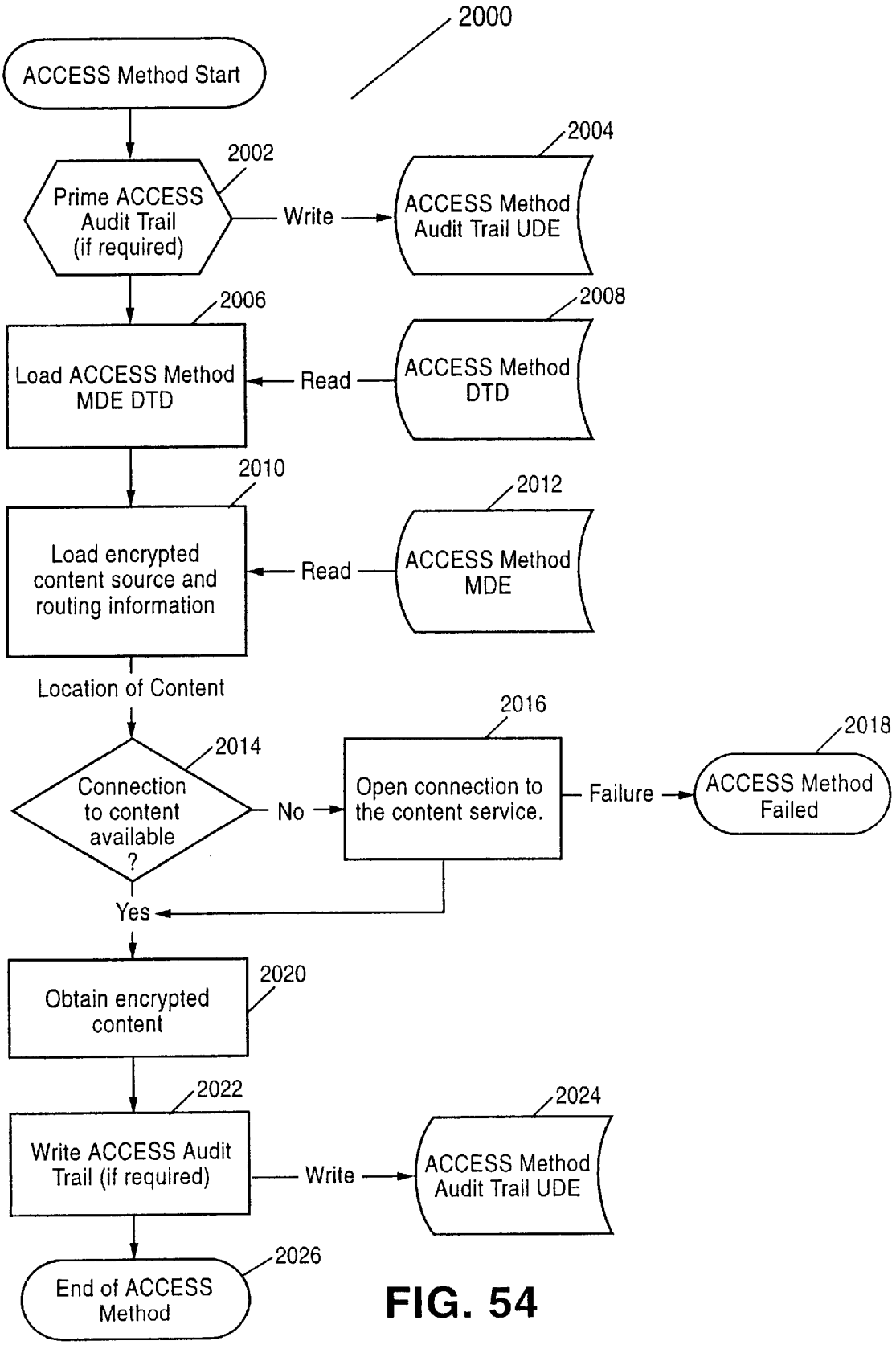

2000
ACCESS Method Start
2002
Prime ACCESS Audit Trail (if required)
Write
2004
ACCESS Method Audit Trail UDE
2006
Load ACCESS Method MDE DTD
Read
2008
ACCESS Method DTD
2010
Load encrypted content source and routing information
Read
2012
ACCESS Method MDE
Location of Content
2014
Connection to content available ?
No
2016
Open connection to the content service.
Failure
2018
ACCESS Method Failed
Yes
2020
Obtain encrypted content
2022
Write ACCESS Audit Trail (if required)
Write
2024
ACCESS Method Audit Trail UDE
2026
End of ACCESS Method

FIG. 54

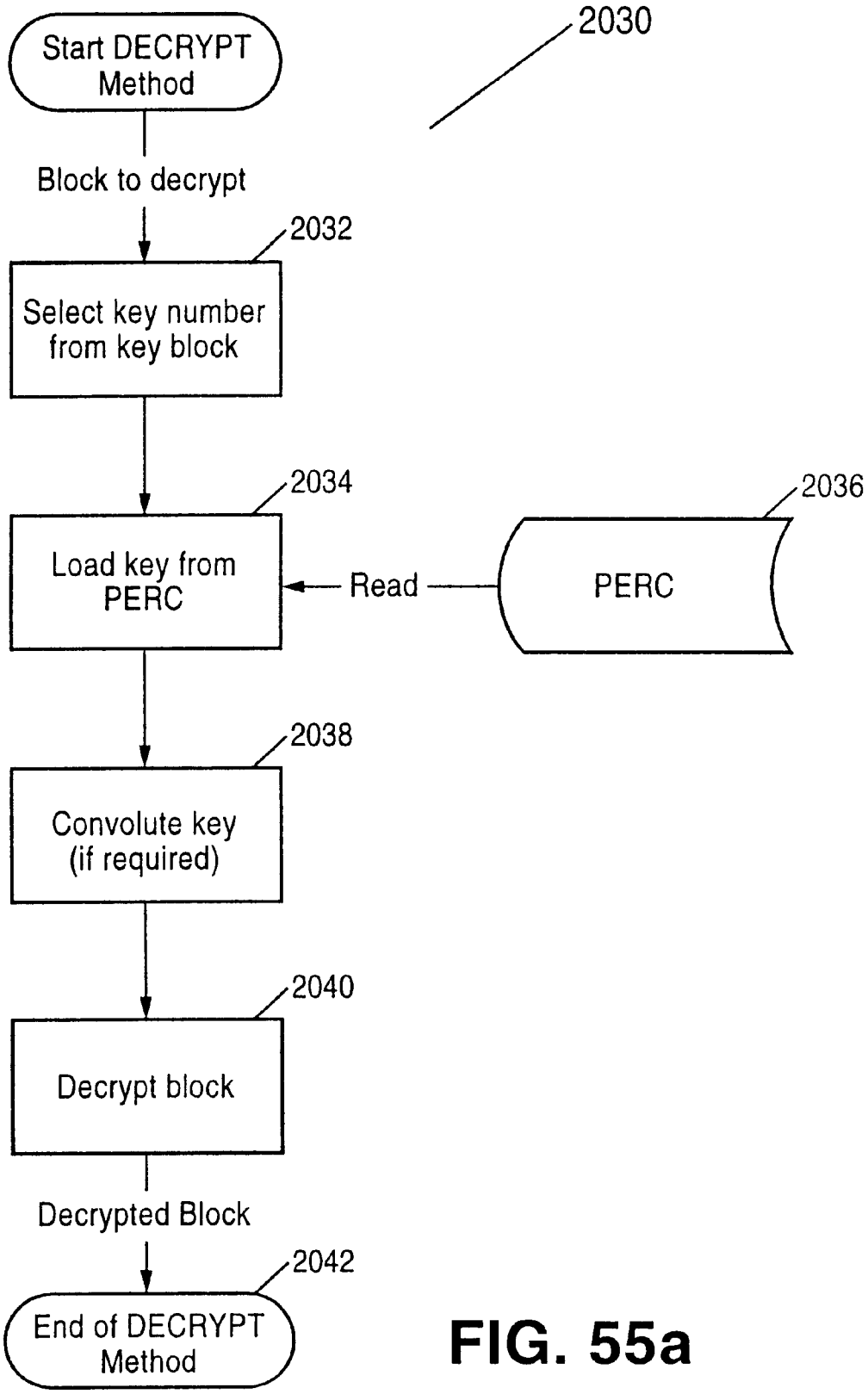

2030
Start DECRYPT Method
Block to decrypt
2032
Select key number from key block
2034
Load key from PERC
Read
2036
PERC
2038
Convolute key (if required)
2040
Decrypt block
Decrypted Block
2042
End of DECRYPT Method

FIG. 55a

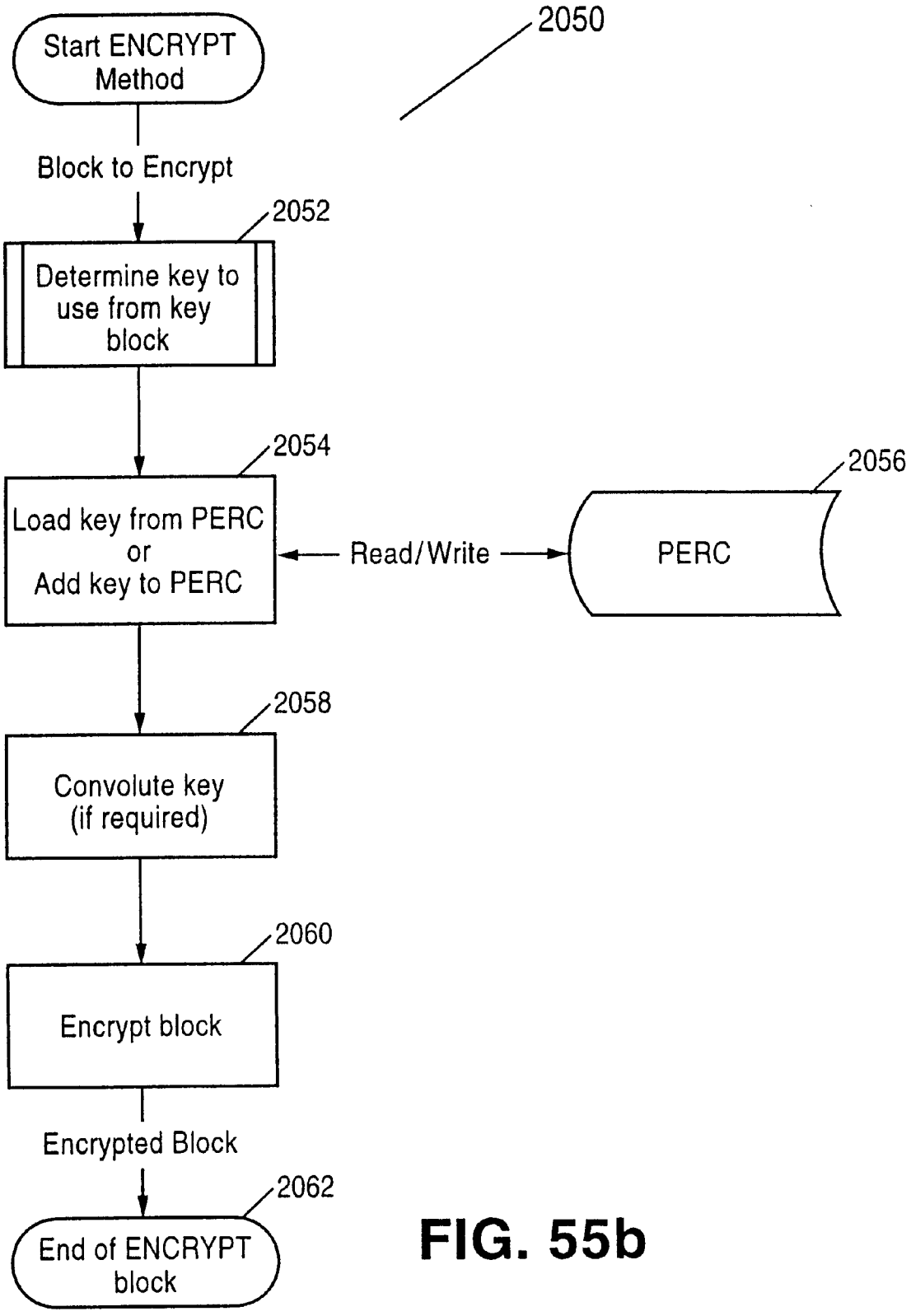

2050
Start ENCRYPT Method
Block to Encrypt
2052
Determine key to use from key block
2054
Load key from PERC or Add key to PERC
Read/Write
2056
PERC
2058
Convolute key (if required)
2060
Encrypt block
Encrypted Block
2062
End of ENCRYPT block

FIG. 55b

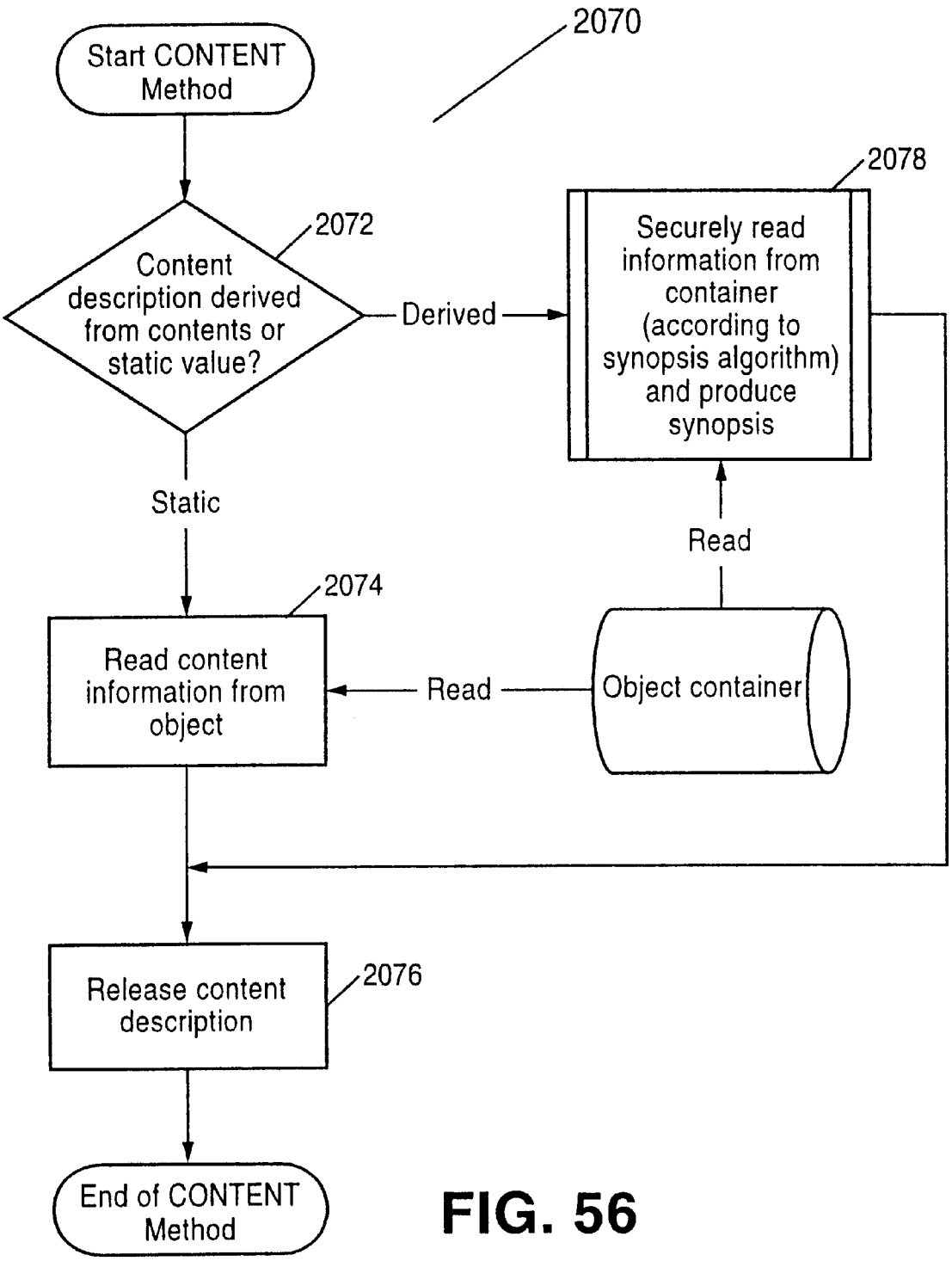

2070
Start CONTENT Method
2072
Content description derived from contents or static value?
Derived
Static
2078
Securely read information from container (according to synopsis algorithm) and produce synopsis
Read
2074
Read content information from object
Read
Object container
2076
Release content description
End of CONTENT Method

FIG. 56

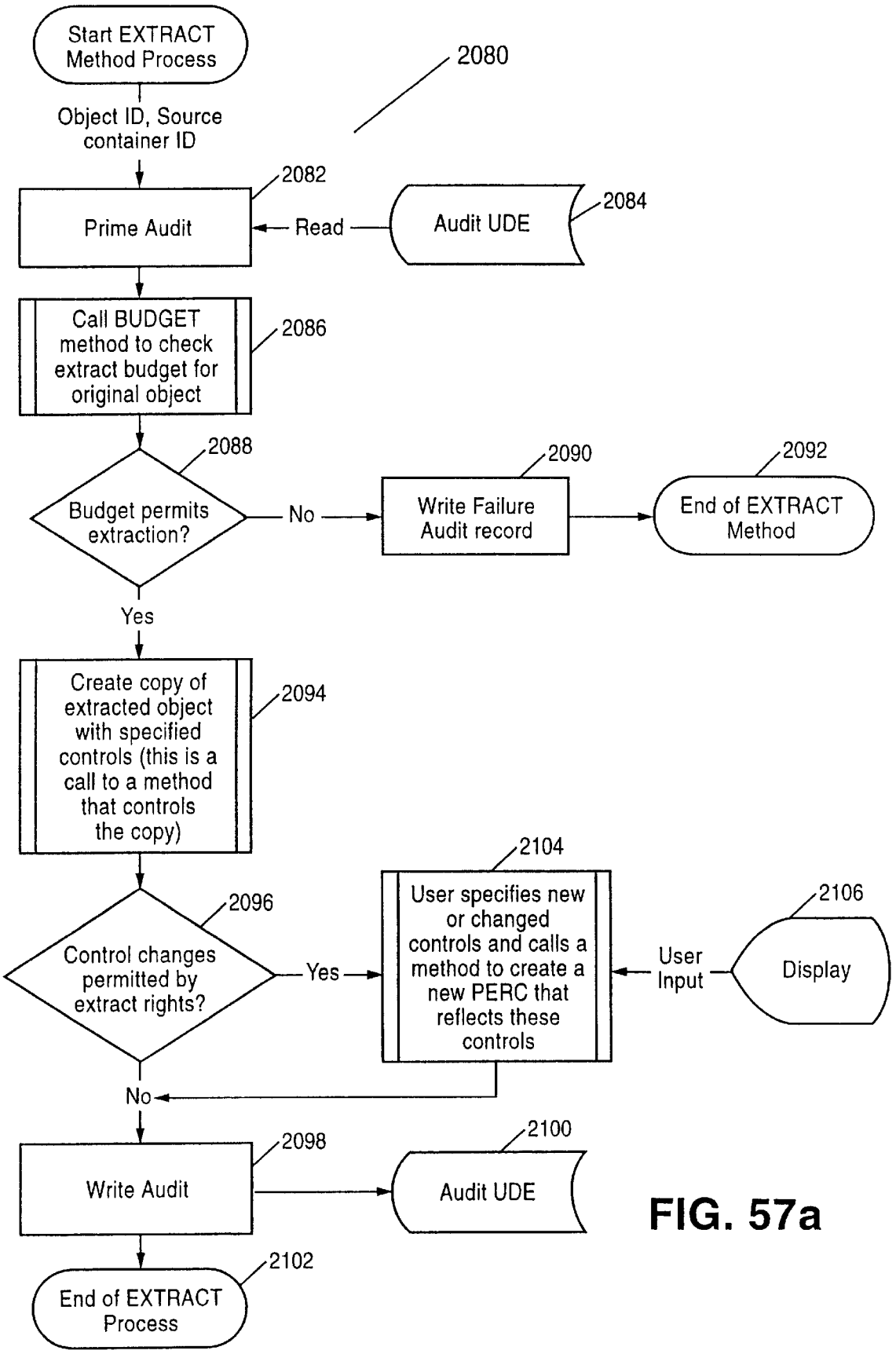

Start EXTRACT Method Process
2080
Object ID, Source container ID
2082
Prime Audit
Read
Audit UDE
2084
Call BUDGET method to check extract budget for original object
2086
2088
Budget permits extraction?
No
2090
Write Failure Audit record
2092
End of EXTRACT Method
Yes
Create copy of extracted object with specified controls (this is a call to a method that controls the copy)
2094
2104
2096
Control changes permitted by extract rights?
Yes
User specifies new or changed controls and calls a method to create a new PERC that reflects these controls
2106
User Input
Display
No
2098
Write Audit
2100
Audit UDE
2102
End of EXTRACT Process

FIG. 57a

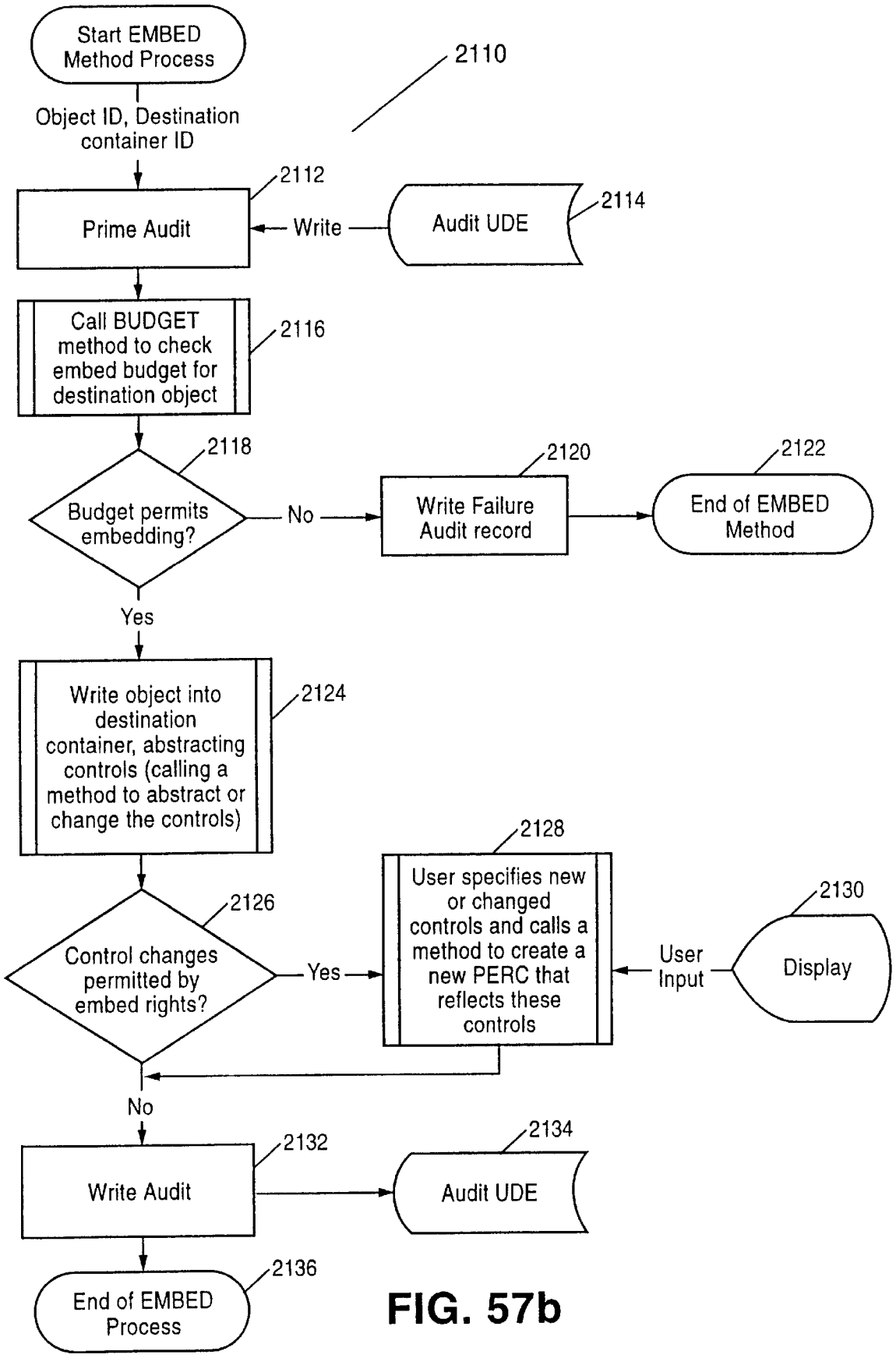

Start EMBED Method Process
2110
Object ID, Destination container ID
2112
Prime Audit
Write
Audit UDE
2114
Call BUDGET method to check embed budget for destination object
2116
2118
Budget permits embedding?
No
2120
Write Failure Audit record
2122
End of EMBED Method
Yes
Write object into destination container, abstracting controls (calling a method to abstract or change the controls)
2124
2128
User specifies new or changed controls and calls a method to create a new PERC that reflects these controls
2130
User Input
Display
2126
Control changes permitted by embed rights?
Yes
No
2132
Write Audit
2134
Audit UDE
2136
End of EMBED Process

FIG. 57b

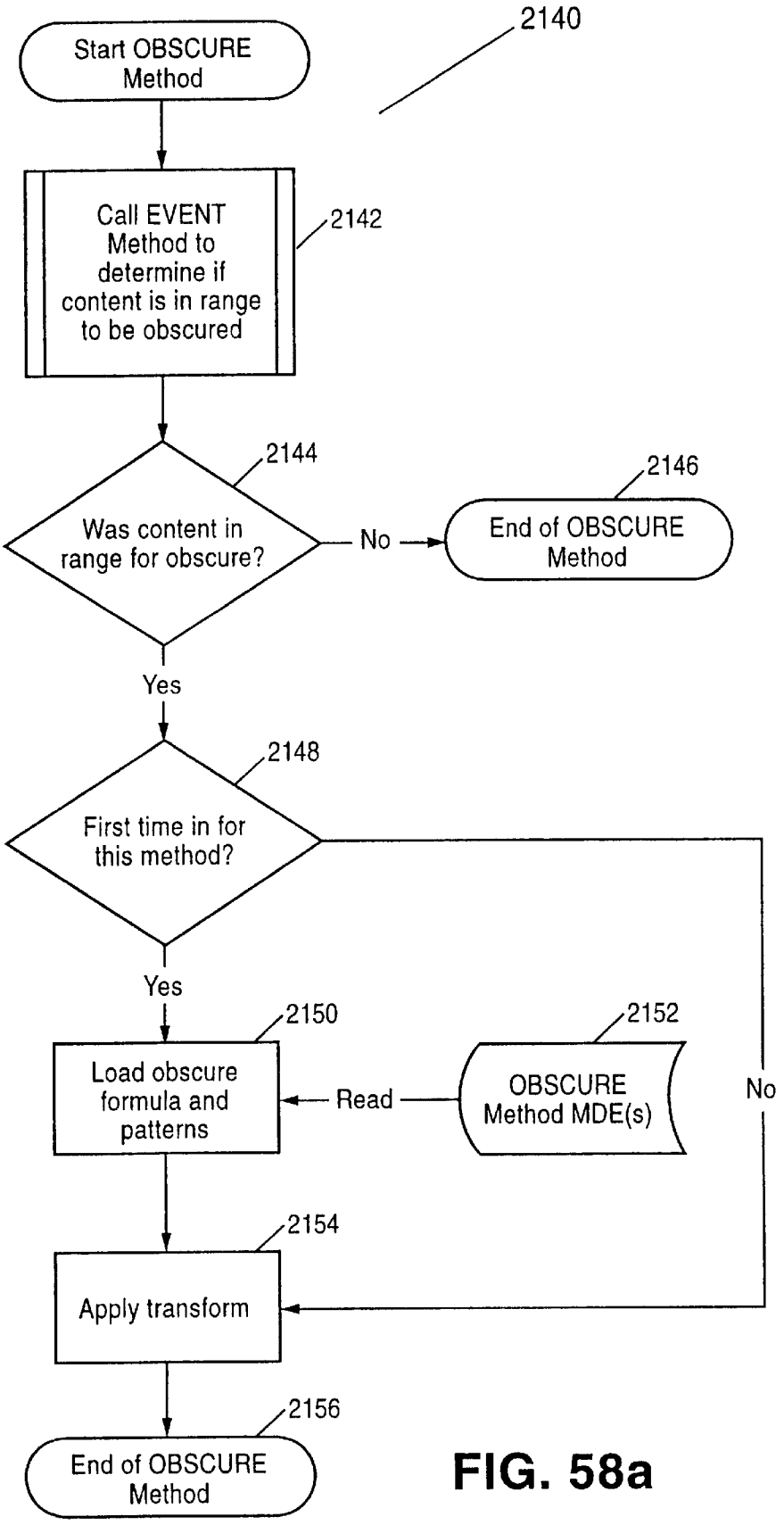

2140
Start OBSCURE Method
Call EVENT Method to determine if content is in range to be obscured
2142
2144
Was content in range for obscure?
No
2146
End of OBSCURE Method
Yes
2148
First time in for this method?
Yes
2150
Load obscure formula and patterns
Read
2152
OBSCURE Method MDE(s)
No
2154
Apply transform
2156
End of OBSCURE Method

FIG. 58a

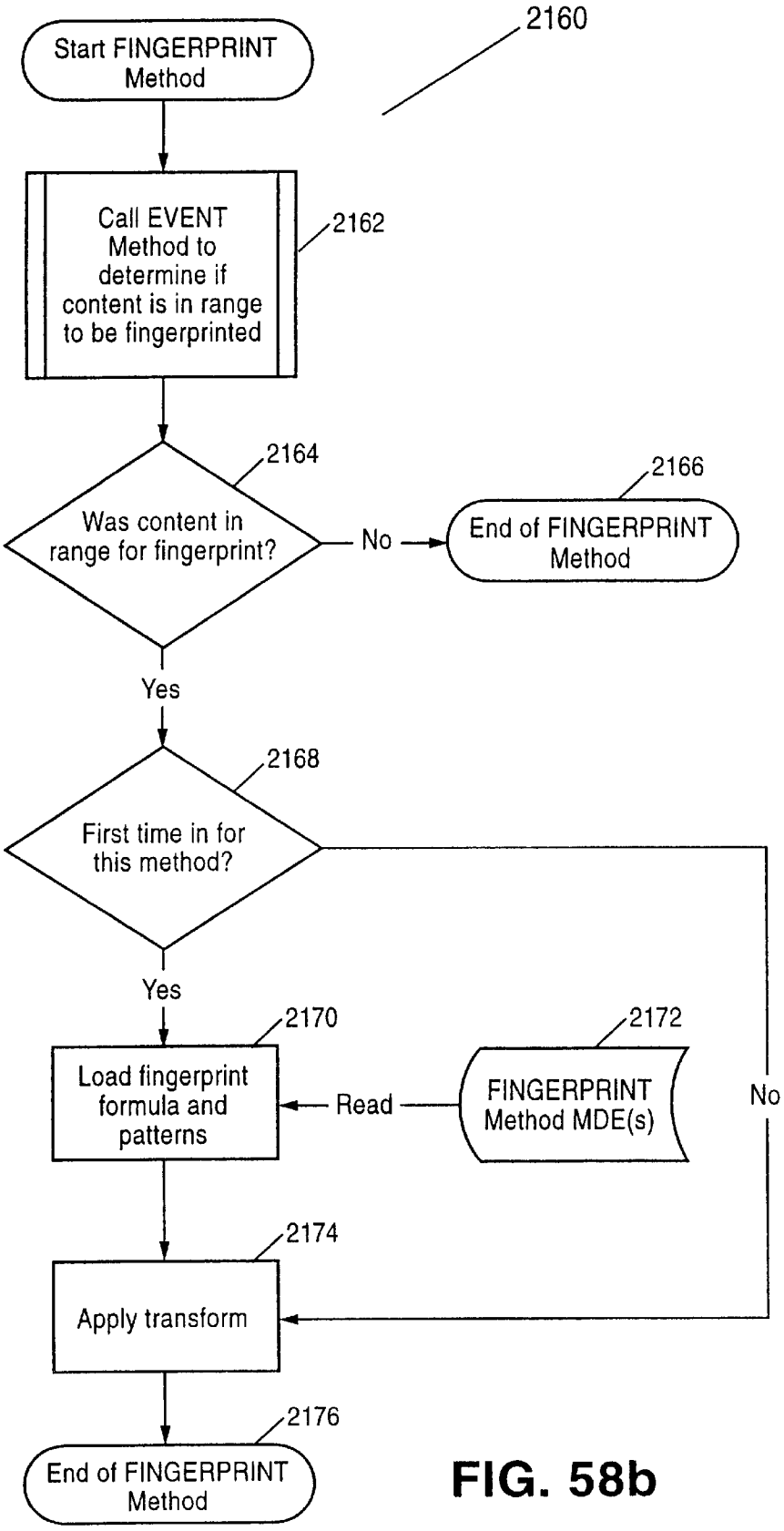

2160
Start FINGERPRINT Method
Call EVENT Method to determine if content is in range to be fingerprinted
2162
2164
Was content in range for fingerprint?
No
2166
End of FINGERPRINT Method
Yes
2168
First time in for this method?
Yes
No
2170
Load fingerprint formula and patterns
Read
2172
FINGERPRINT Method MDE(s)
2174
Apply transform
2176
End of FINGERPRINT Method

FIG. 58b

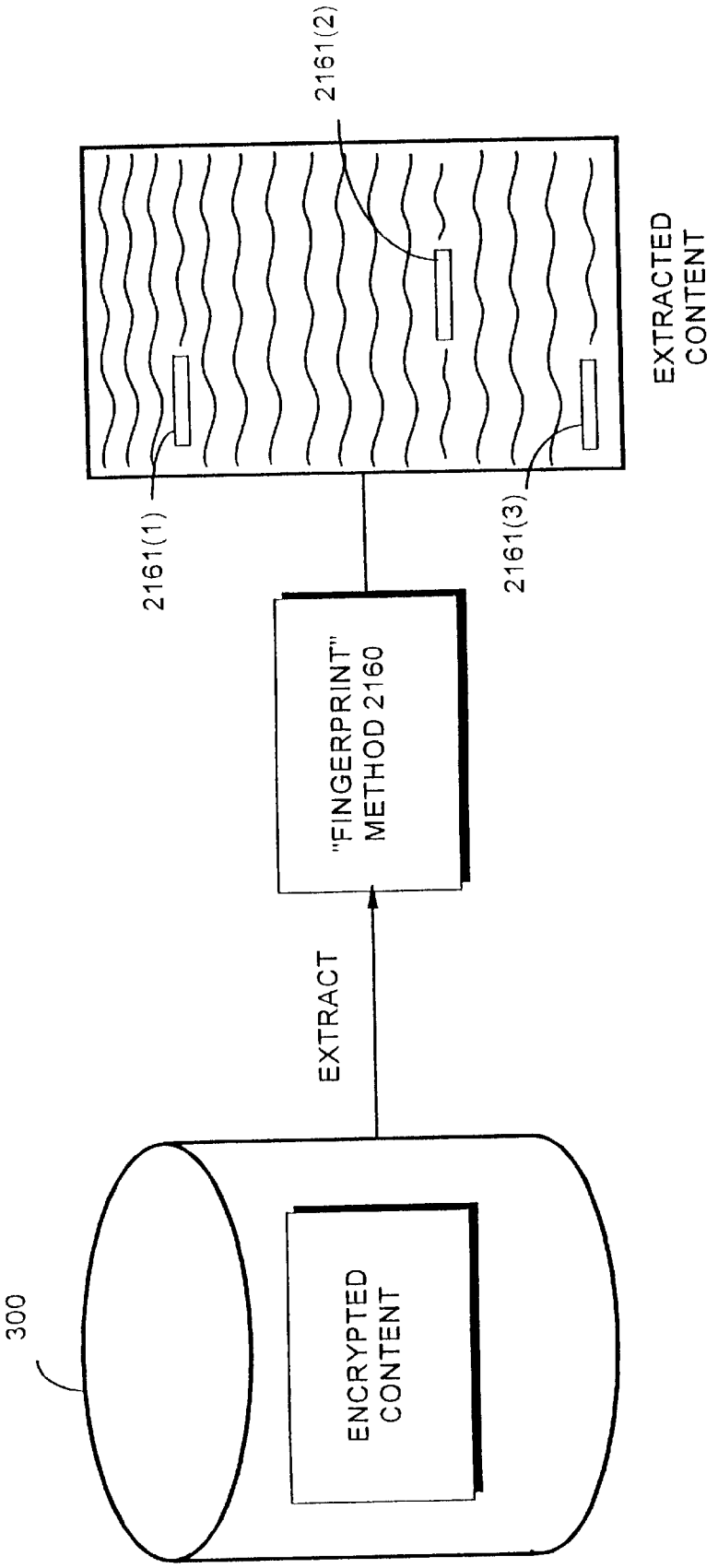

2161(2)
EXTRACTED CONTENT
2161(1)
2161(3)
"FINGERPRINT" METHOD 2160
EXTRACT
300
ENCRYPTED CONTENT

FIG. 58C

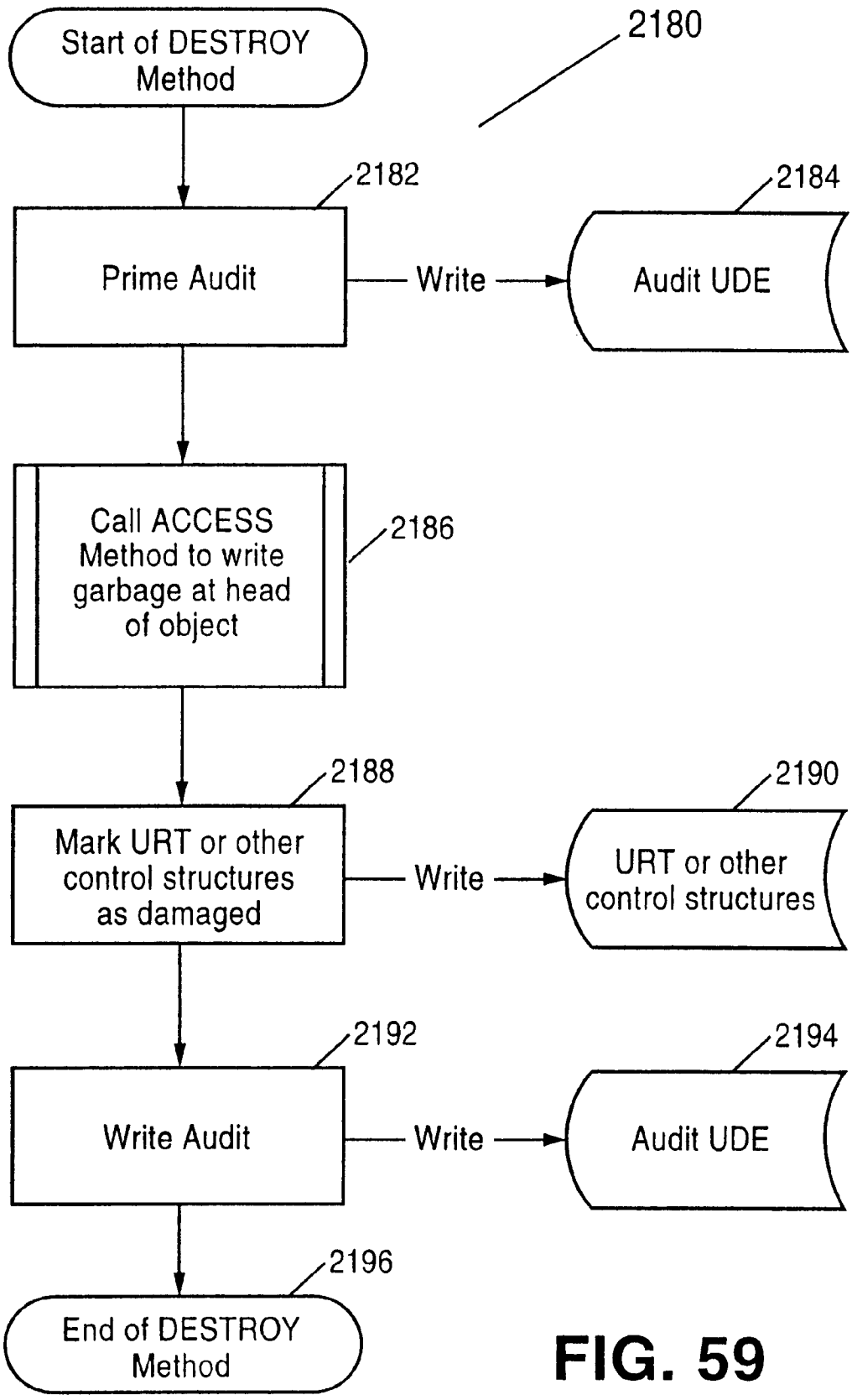

Start of DESTROY Method
2180
2182
Prime Audit
Write
2184
Audit UDE
Call ACCESS Method to write garbage at head of object
2186
2188
Mark URT or other control structures as damaged
Write
2190
URT or other control structures
2192
Write Audit
Write
2194
Audit UDE
2196
End of DESTROY Method

FIG. 59

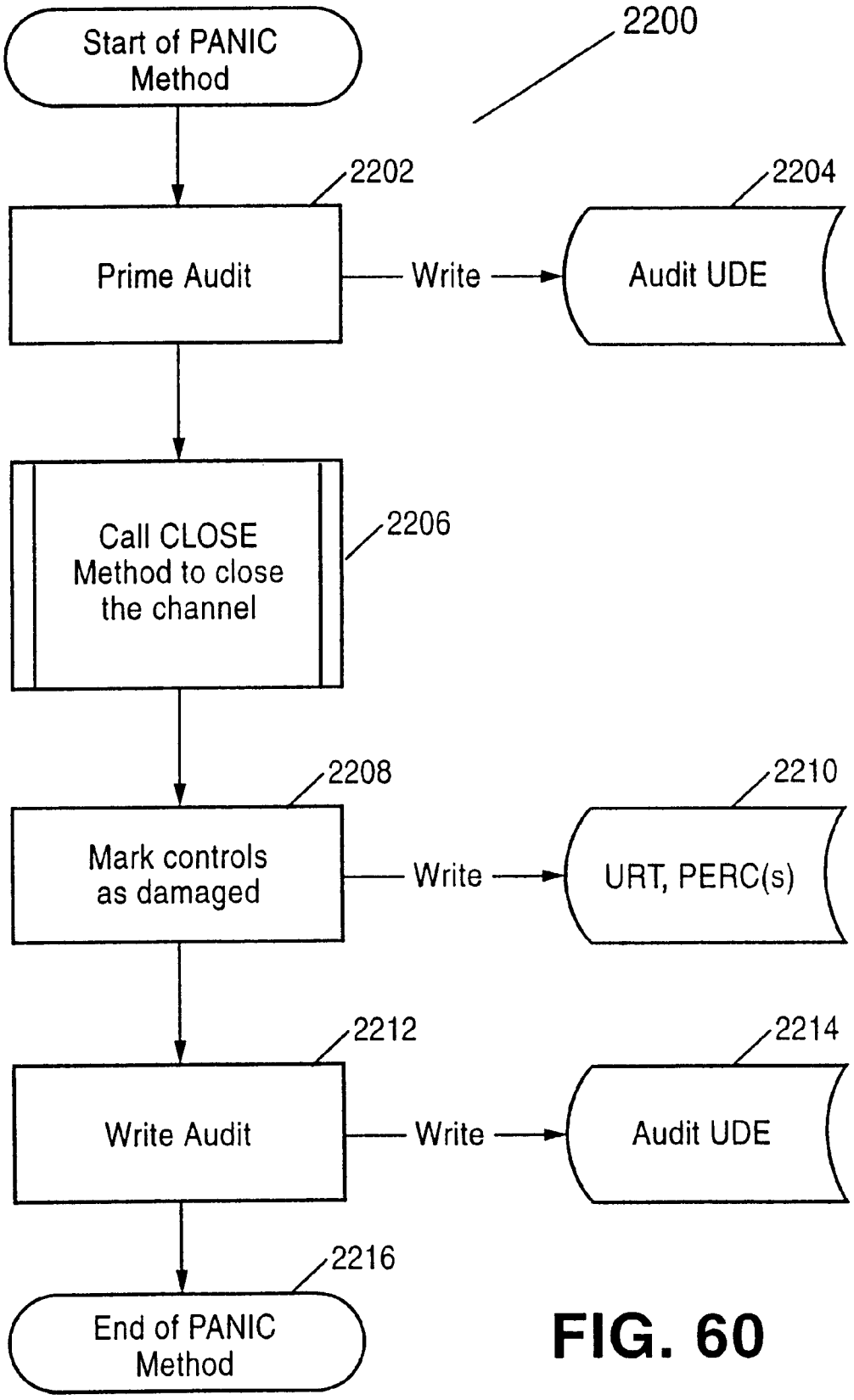

Start of PANIC Method
2200
2202
Prime Audit
Write
2204
Audit UDE
Call CLOSE Method to close the channel
2206
2208
Mark controls as damaged
Write
2210
URT, PERC(s)
2212
Write Audit
Write
2214
Audit UDE
2216
End of PANIC Method

FIG. 60

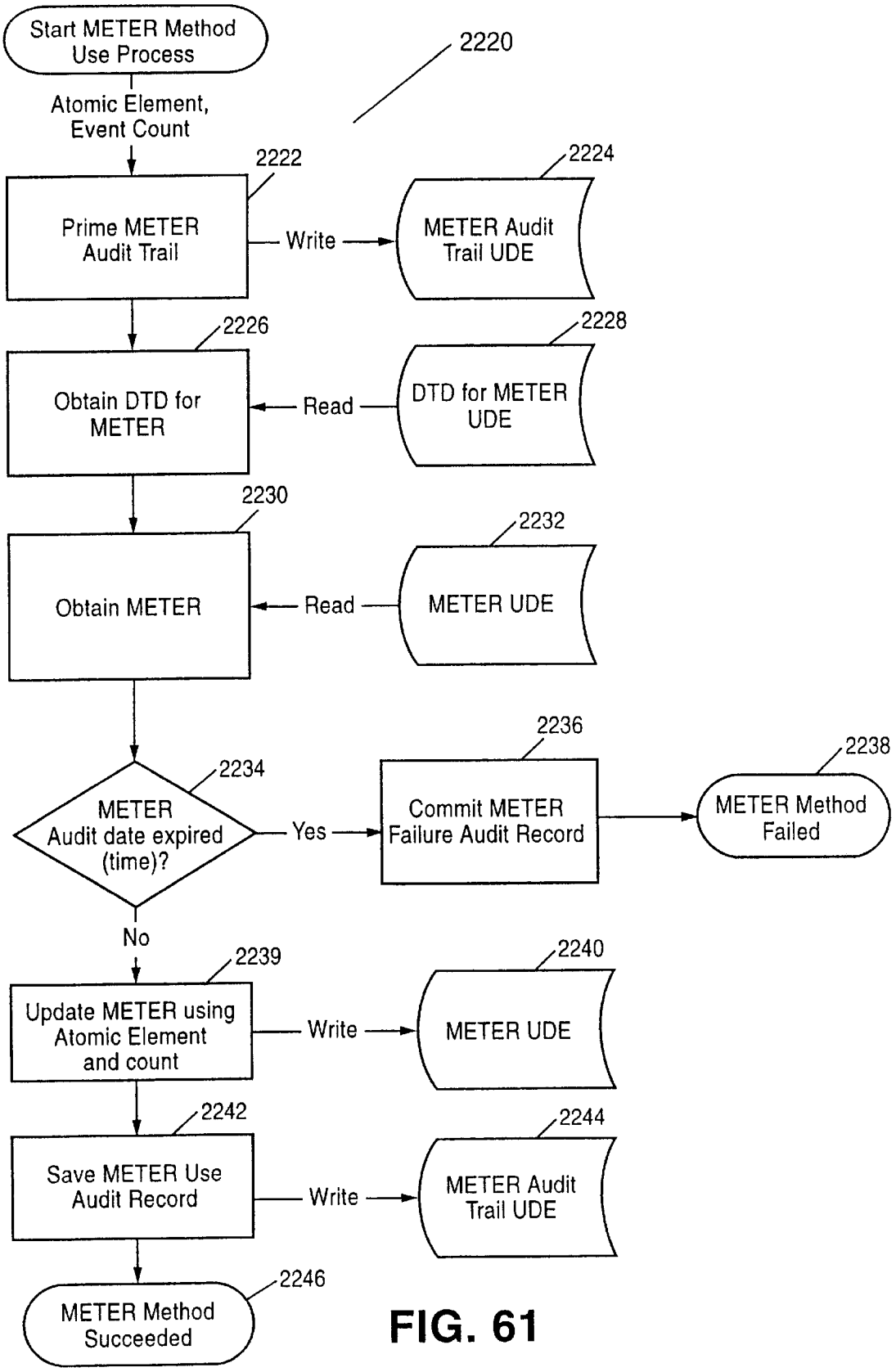
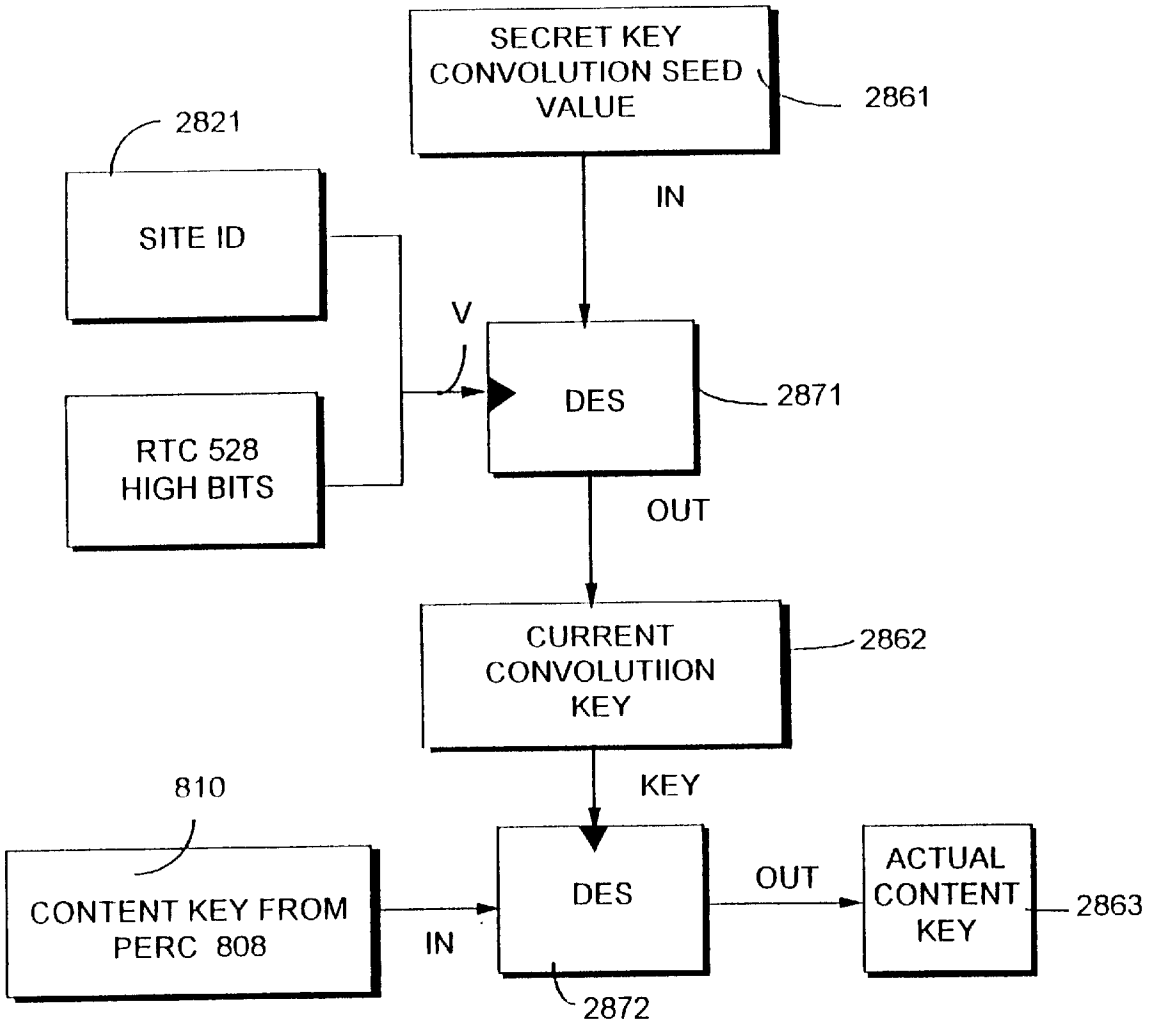

Start METER Method Use Process
2220
Atomic Element, Event Count
2222
Prime METER Audit Trail
Write
2224
METER Audit Trail UDE
2226
Obtain DTD for METER
Read
2228
DTD for METER UDE
2230
Obtain METER
Read
2232
METER UDE
2234
METER Audit date expired (time)?
Yes
2236
Commit METER Failure Audit Record
2238
METER Method Failed
No
2239
Update METER using Atomic Element and count
Write
2240
METER UDE
2242
Save METER Use Audit Record
Write
2244
METER Audit Trail UDE
2246
METER Method Succeeded

FIG. 61

FIG. 62


SECRET KEY CONVOLUTION SEED VALUE
2861
2821
SITE ID
IN
V
DES
2871
RTC 528 HIGH BITS
OUT
CURRENT CONVOLUTIION KEY
2862
KEY
810
CONTENT KEY FROM PERC 808
IN
DES
OUT
ACTUAL CONTENT KEY
2863
2872

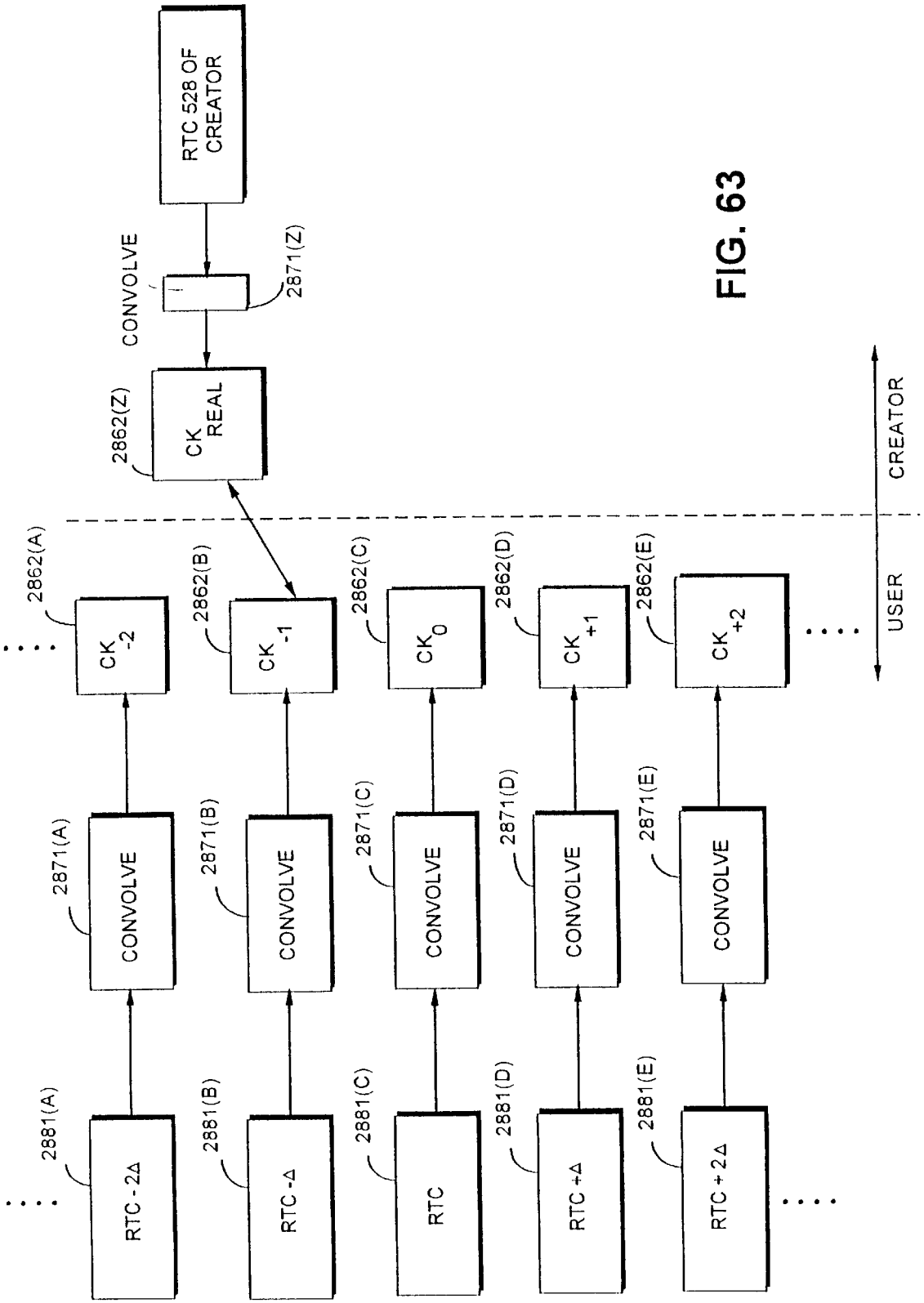

RTC 528 OF CREATOR
CONVOLVE
2871(Z)
2862(Z)
CK REAL
CREATOR
USER
2862(A)
2862(B)
2862(C)
2862(D)
2862(E)
CK -2
CK -1
CK 0
CK +1
CK +2
2871(A)
2871(B)
2871(C)
2871(D)
2871(E)
CONVOLVE
2881(A)
2881(B)
2881(C)
2881(D)
2881(E)
RTC - 2Δ
RTC -Δ
RTC
RTC +Δ
RTC + 2Δ

FIG. 63

FIG. 64

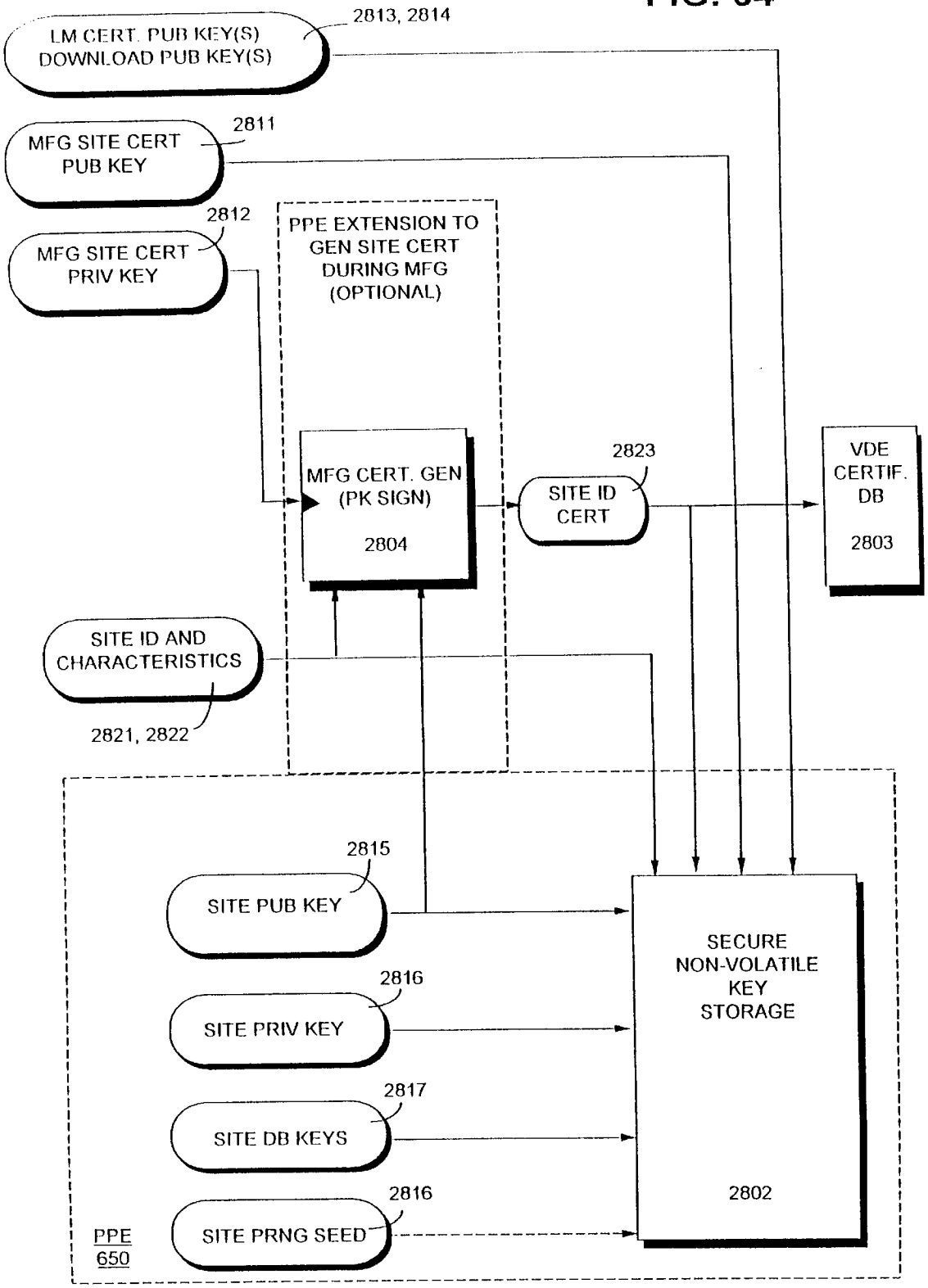

LM CERT. PUB KEY(S)
DOWNLOAD PUB KEY(S)
2813, 2814
MFG SITE CERT
PUB KEY
2811
MFG SITE CERT
PRIV KEY
2812
PPE EXTENSION TO
GEN SITE CERT
DURING MFG
(OPTIONAL)
MFG CERT. GEN
(PK SIGN)
2804
SITE ID
CERT
2823
VDE
CERTIF.
DB
2803
SITE ID AND
CHARACTERISTICS
2821, 2822
2815
SITE PUB KEY
2816
SITE PRIV KEY
2817
SITE DB KEYS
2816
SITE PRNG SEED
SECURE
NON-VOLATILE
KEY
STORAGE
2802
PPE
650

FIG. 65

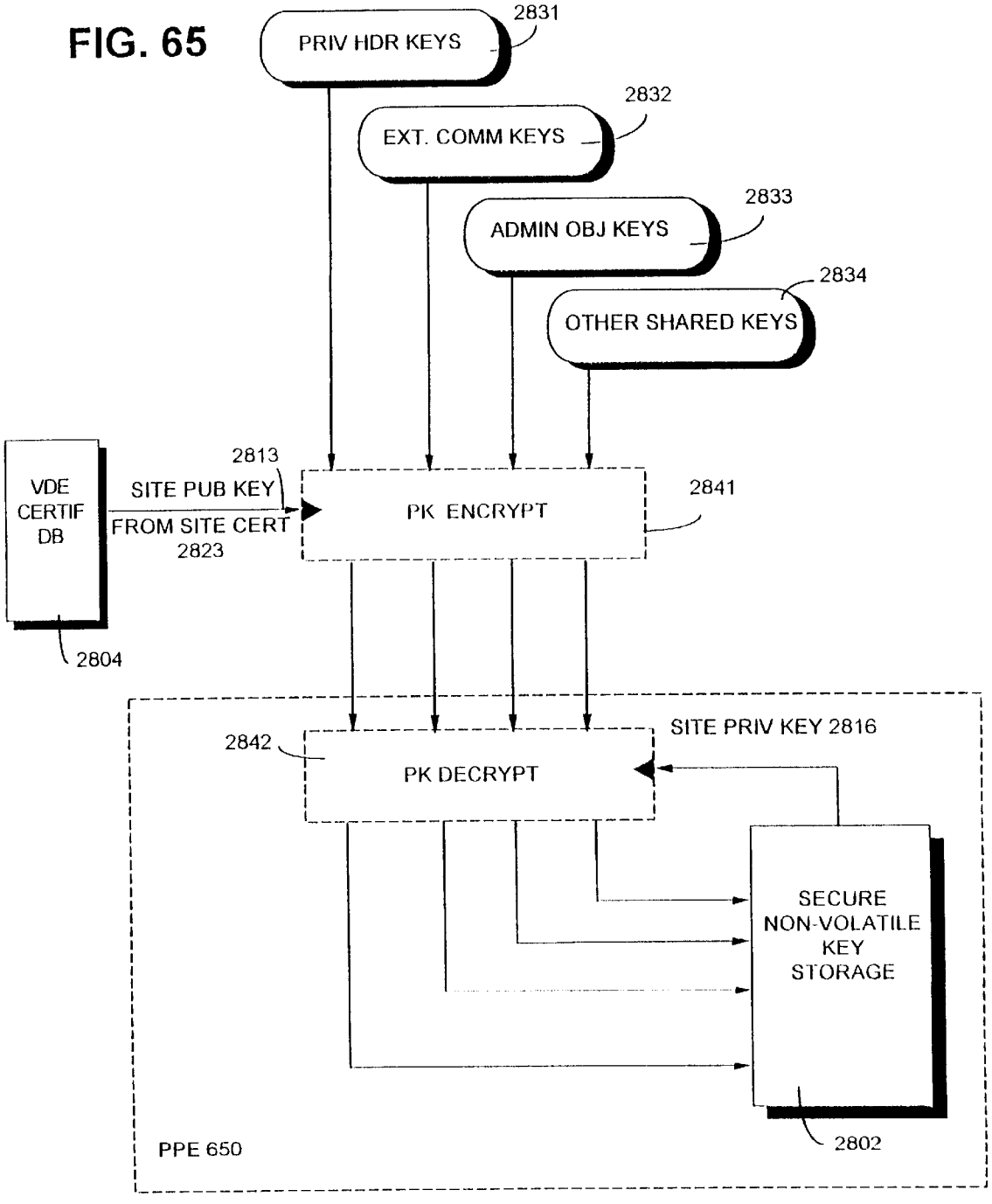
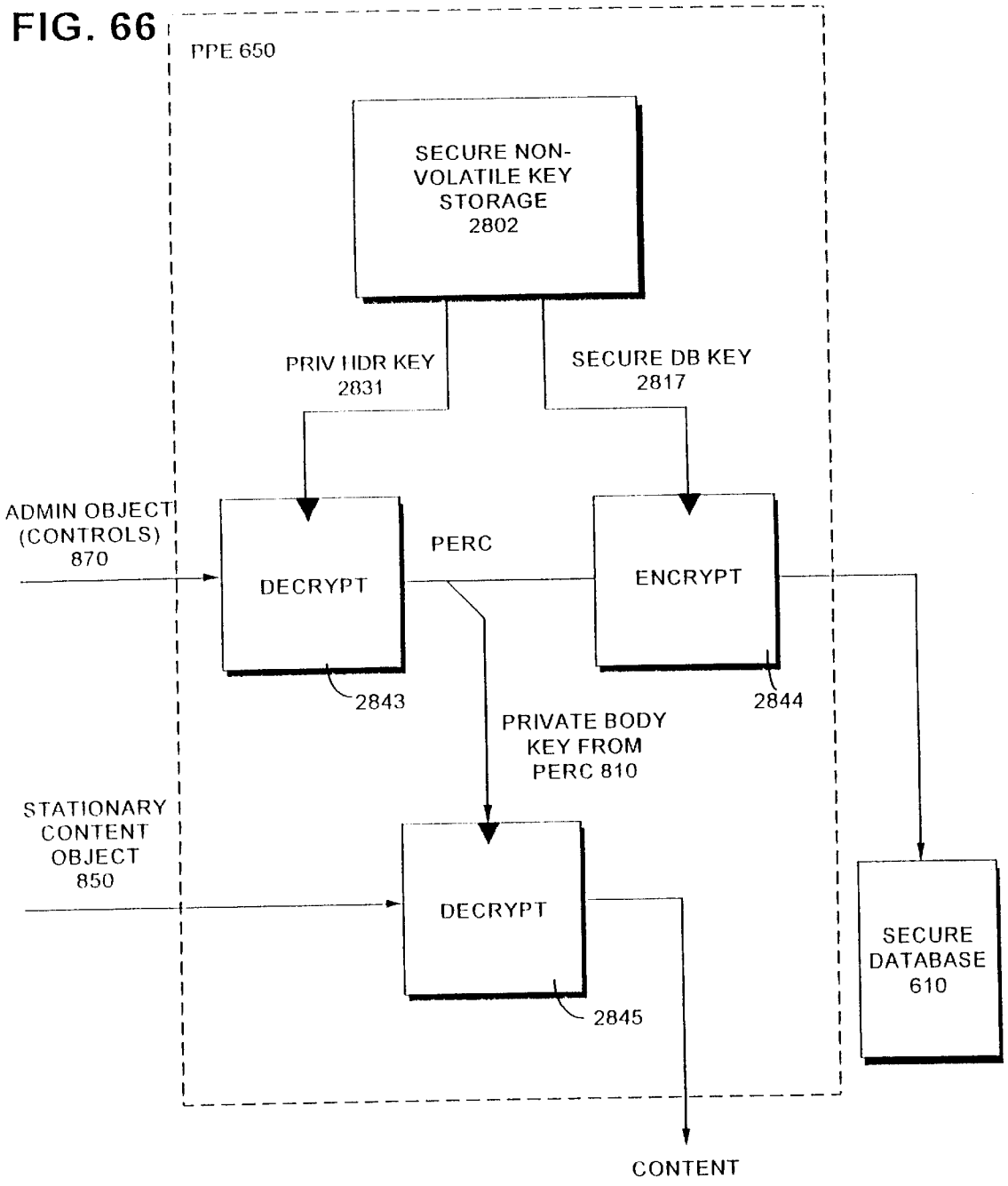

PRIV HDR KEYS
2831
EXT. COMM KEYS
2832
ADMIN OBJ KEYS
2833
OTHER SHARED KEYS
2834
VDE CERTIF DB
2804
SITE PUB KEY FROM SITE CERT 2823
2813
PK ENCRYPT
2841
SITE PRIV KEY 2816
2842
PK DECRYPT
SECURE NON-VOLATILE KEY STORAGE
2802
PPE 650

FIG. 66


PPE 650
SECURE NON-VOLATILE KEY STORAGE 2802
PRIV HDR KEY 2831
SECURE DB KEY 2817
ADMIN OBJECT (CONTROLS) 870
DECRYPT
PERC
ENCRYPT
2843
2844
PRIVATE BODY KEY FROM PERC 810
STATIONARY CONTENT OBJECT 850
DECRYPT
2845
SECURE DATABASE 610
CONTENT

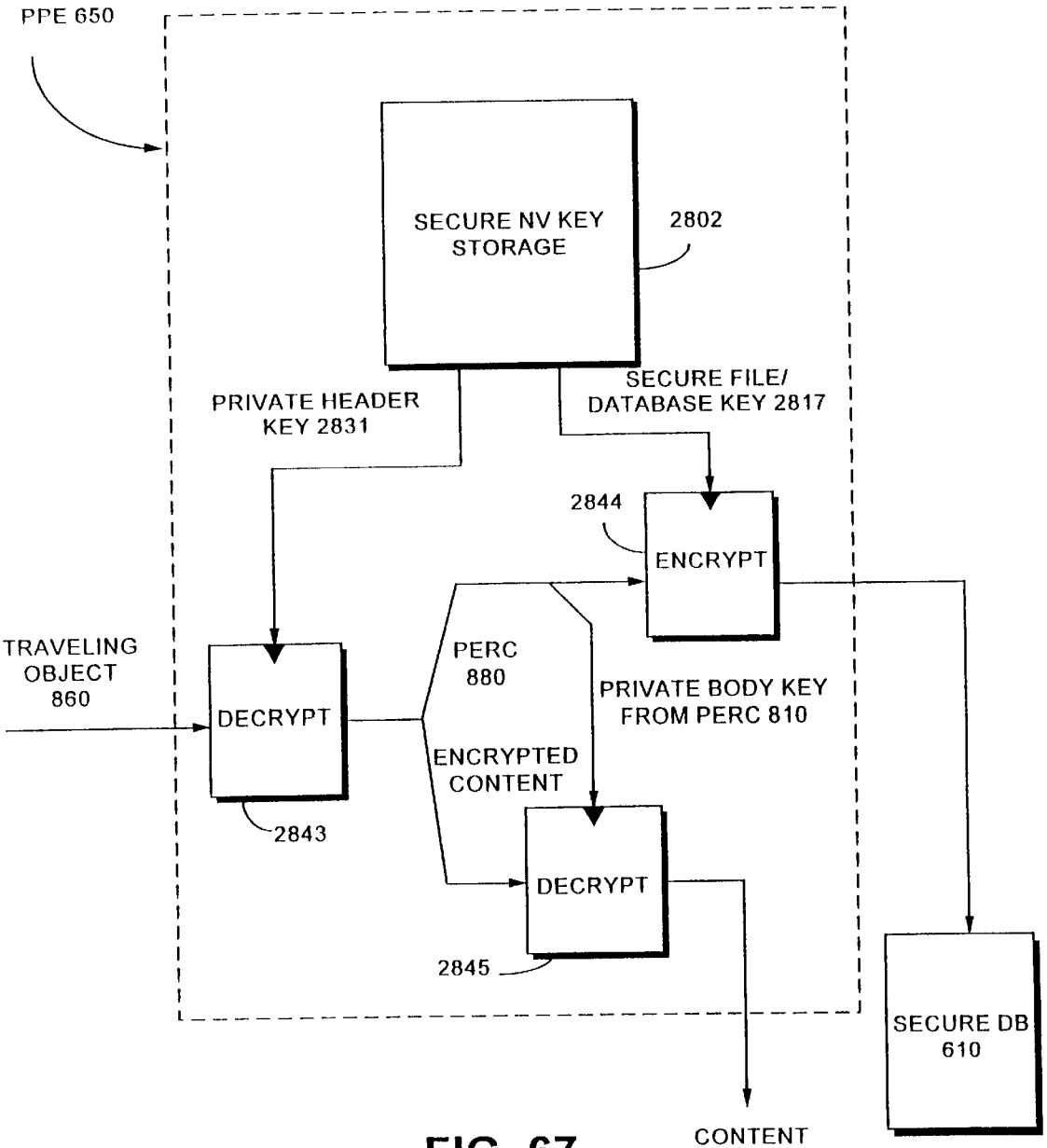

PPE 650
SECURE NV KEY STORAGE
2802
PRIVATE HEADER KEY 2831
SECURE FILE/ DATABASE KEY 2817
2844
ENCRYPT
TRAVELING OBJECT 860
DECRYPT
2843
PERC 880
PRIVATE BODY KEY FROM PERC 810
ENCRYPTED CONTENT
DECRYPT
2845
SECURE DB 610
CONTENT

FIG. 67

FIG. 68

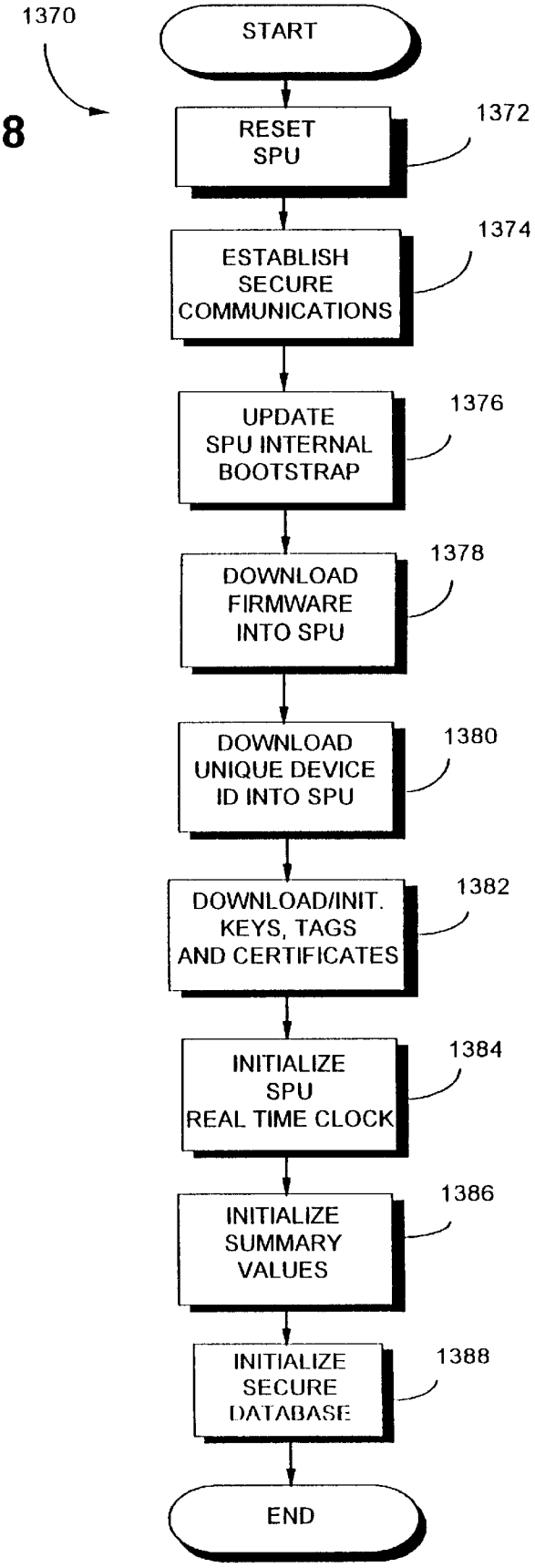
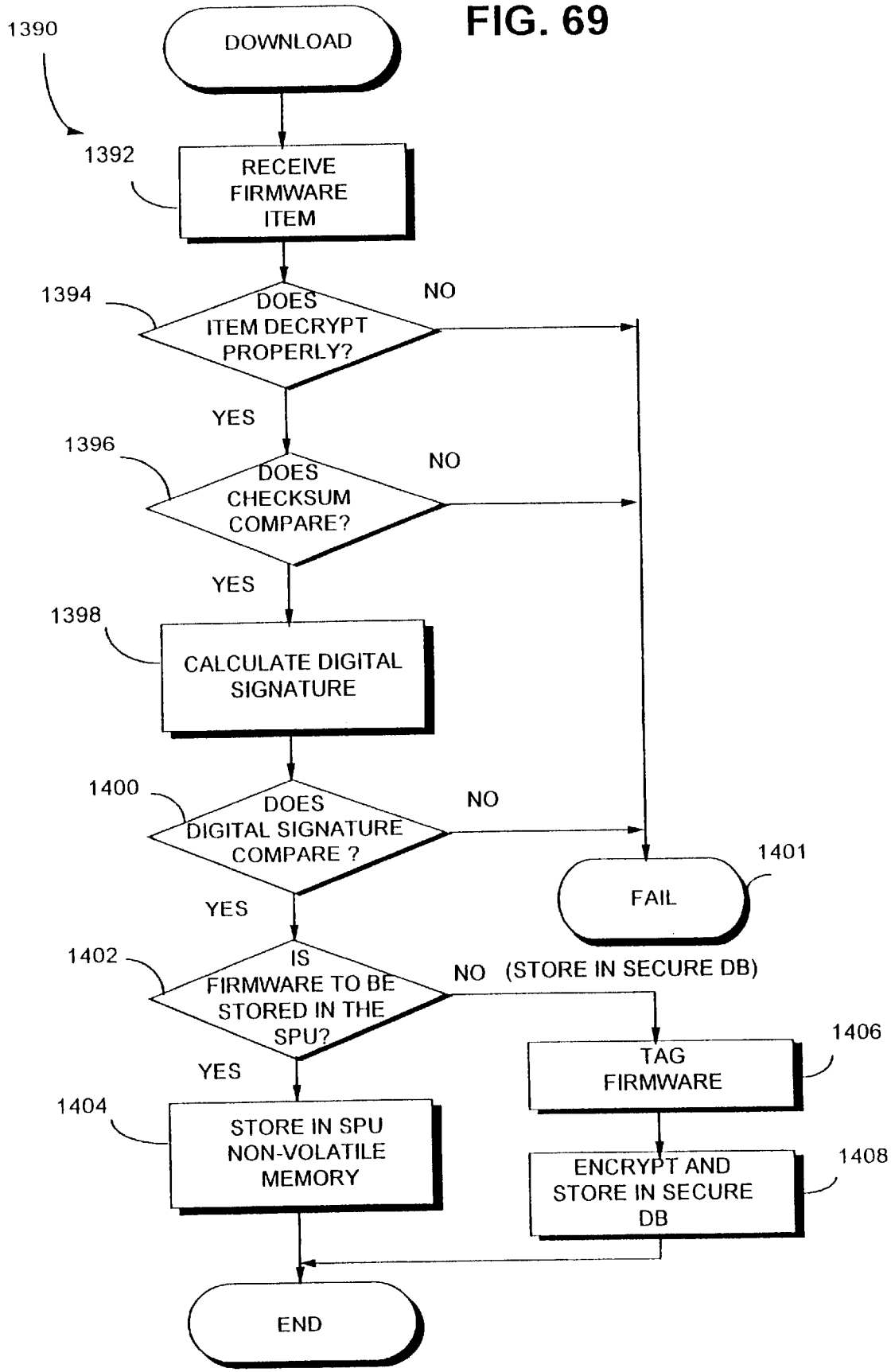

1370
START
RESET SPU
1372
ESTABLISH SECURE COMMUNICATIONS
1374
UPDATE SPU INTERNAL BOOTSTRAP
1376
DOWNLOAD FIRMWARE INTO SPU
1378
DOWNLOAD UNIQUE DEVICE ID INTO SPU
1380
DOWNLOAD/INIT. KEYS, TAGS AND CERTIFICATES
1382
INITIALIZE SPU REAL TIME CLOCK
1384
INITIALIZE SUMMARY VALUES
1386
INITIALIZE SECURE DATABASE
1388
END

FIG. 69


1390
DOWNLOAD
1392
RECEIVE FIRMWARE ITEM
1394
DOES ITEM DECRYPT PROPERLY?
NO
YES
1396
DOES CHECKSUM COMPARE?
NO
YES
1398
CALCULATE DIGITAL SIGNATURE
1400
DOES DIGITAL SIGNATURE COMPARE ?
NO
YES
1401
FAIL
1402
IS FIRMWARE TO BE STORED IN THE SPU?
NO (STORE IN SECURE DB)
YES
1404
STORE IN SPU NON-VOLATILE MEMORY
1406
TAG FIRMWARE
1408
ENCRYPT AND STORE IN SECURE DB
END

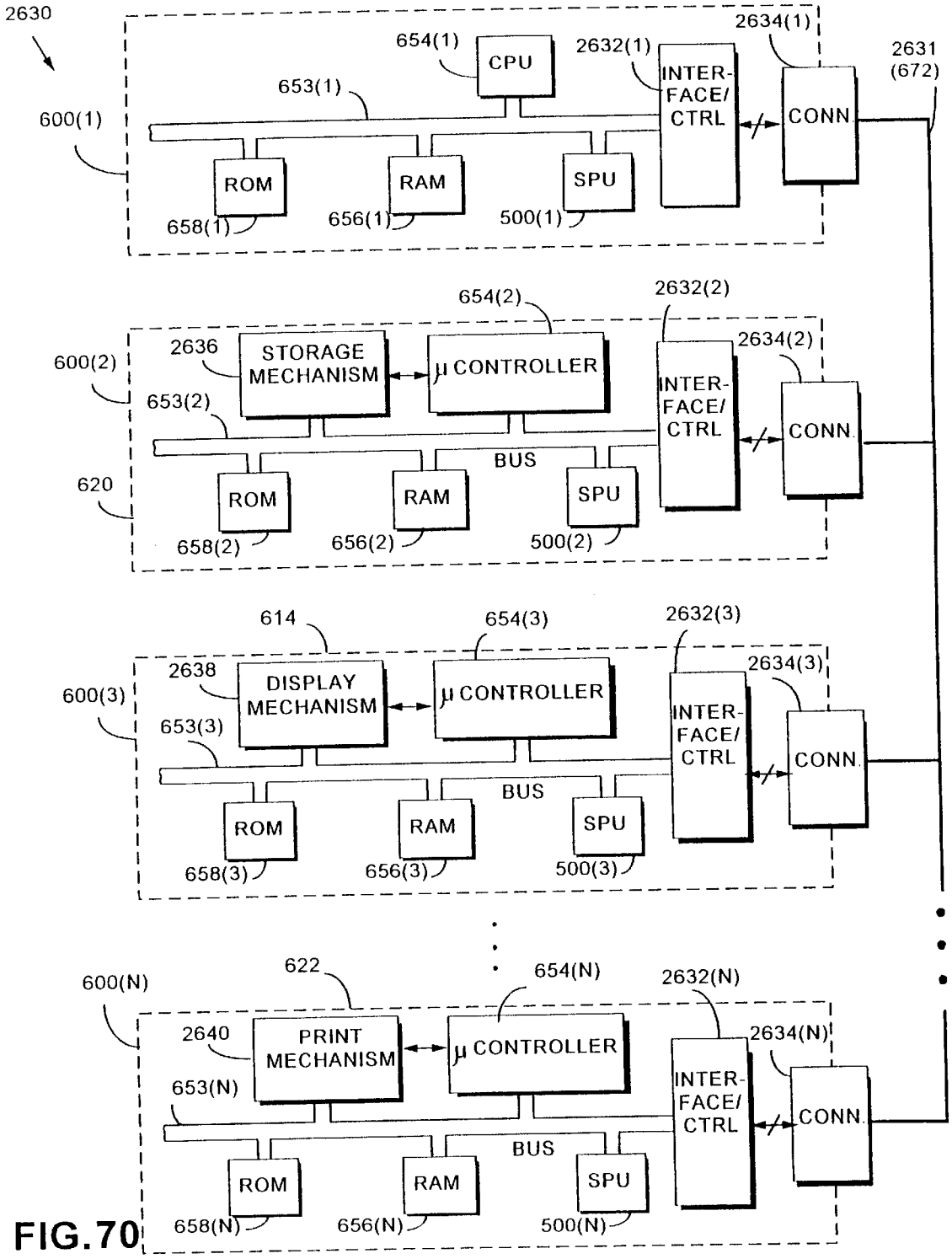

2630
600(1)
654(1)
CPU
653(1)
2632(1)
INTER-FACE/ CTRL
2634(1)
CONN.
2631 (672)
ROM
RAM
SPU
658(1)
656(1)
500(1)
654(2)
2632(2)
600(2)
2636
STORAGE MECHANISM
μ CONTROLLER
2634(2)
653(2)
BUS
620
658(2)
656(2)
500(2)
614
654(3)
2632(3)
600(3)
2638
DISPLAY MECHANISM
2634(3)
653(3)
658(3)
656(3)
500(3)
622
654(N)
2632(N)
600(N)
2640
PRINT MECHANISM
2634(N)
653(N)
658(N)
656(N)
500(N)

FIG.70

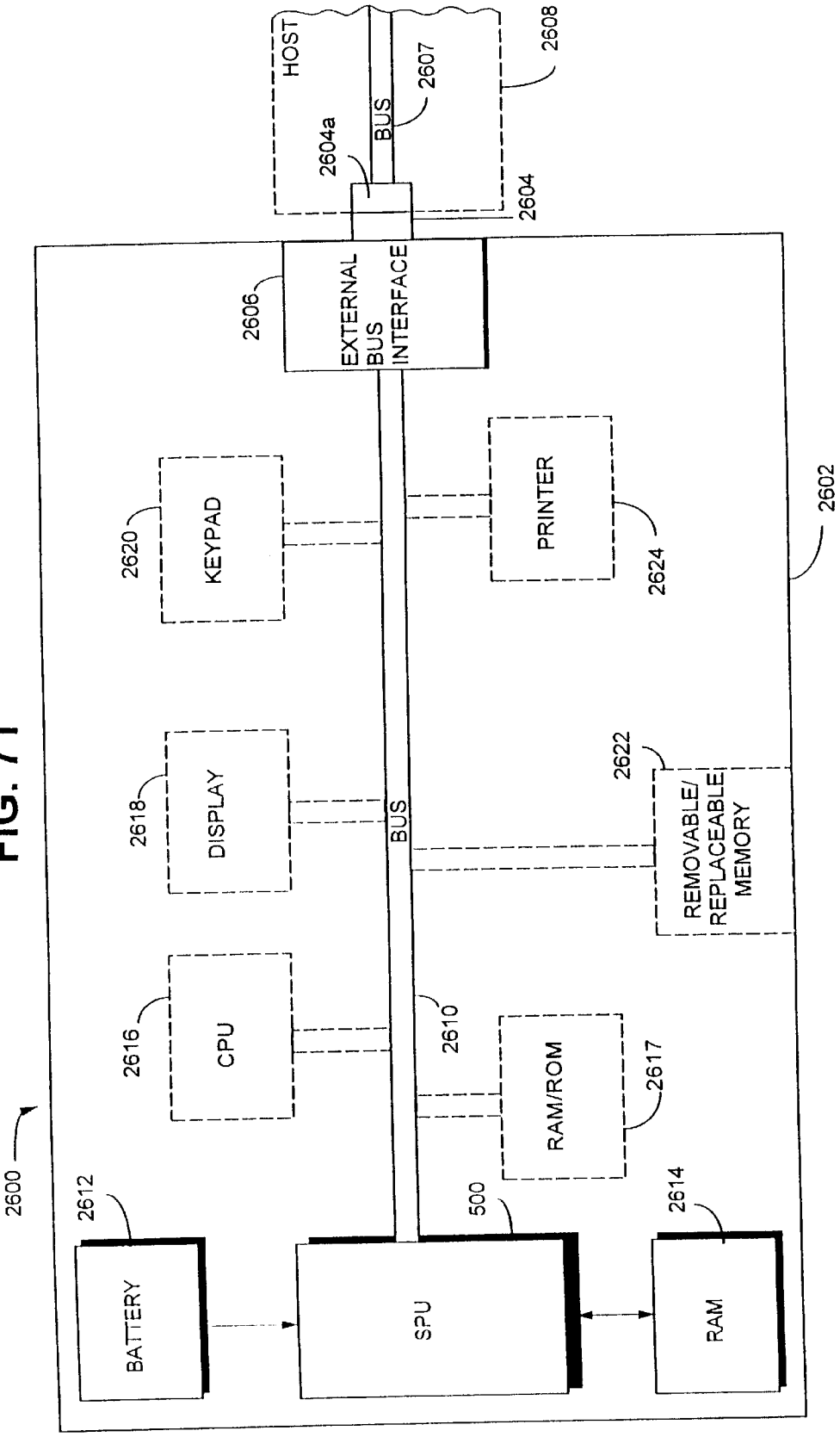

HOST
2608
2607
BUS
2604a
2604
2606
EXTERNAL BUS INTERFACE
2602
2620
KEYPAD
PRINTER
2624
2618
DISPLAY
BUS
2622
REMOVABLE/ REPLACEABLE MEMORY
2616
CPU
2610
RAM/ROM
2617
2600
2612
500
2614
BATTERY
SPU
RAM

FIG. 71

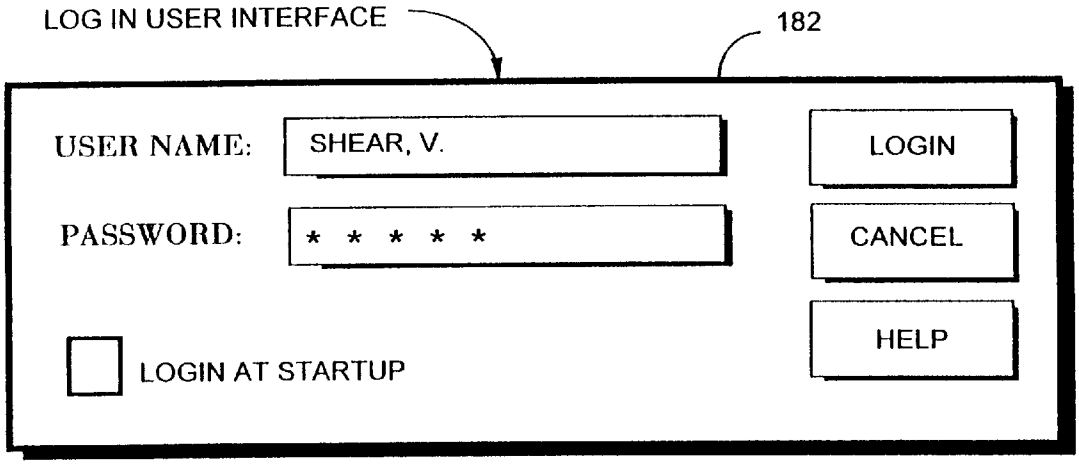


FIG. 72A

FIG. 72B

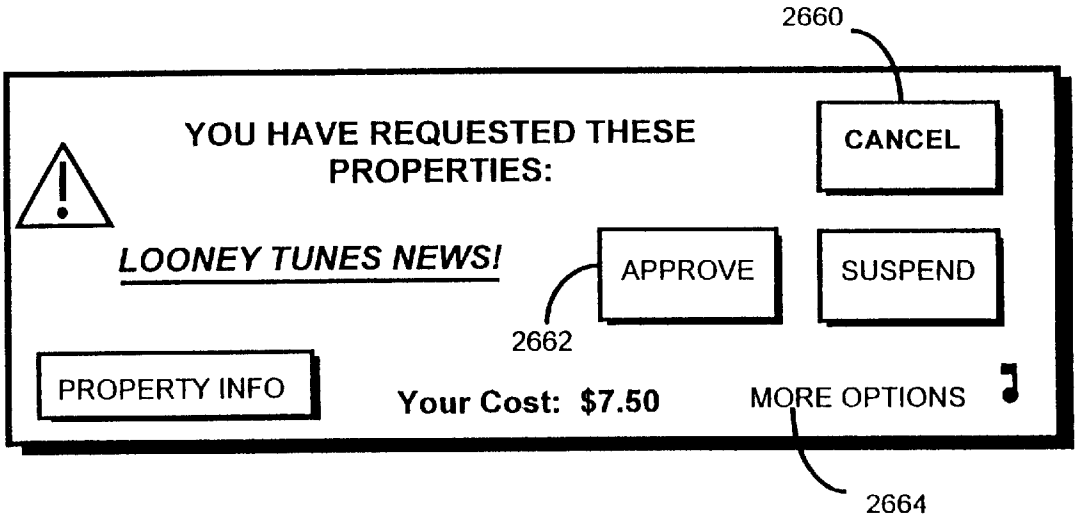


FIG. 72C

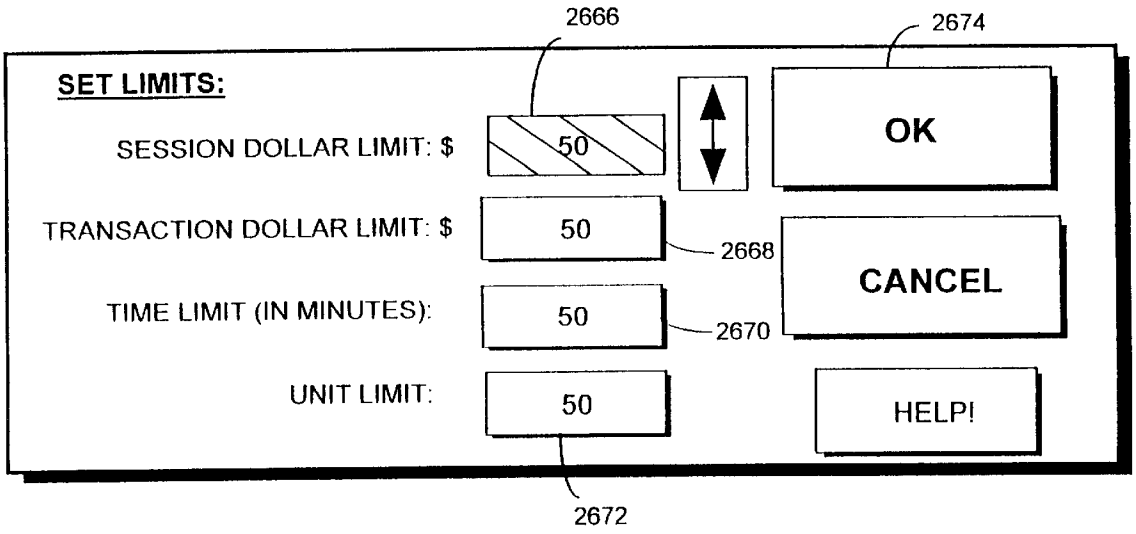

2666
2674
SET LIMITS:
SESSION DOLLAR LIMIT: $
50
OK
TRANSACTION DOLLAR LIMIT: $
50
2668
CANCEL
TIME LIMIT (IN MINUTES):
50
2670
UNIT LIMIT:
50
HELP!
2672

FIG. 72D

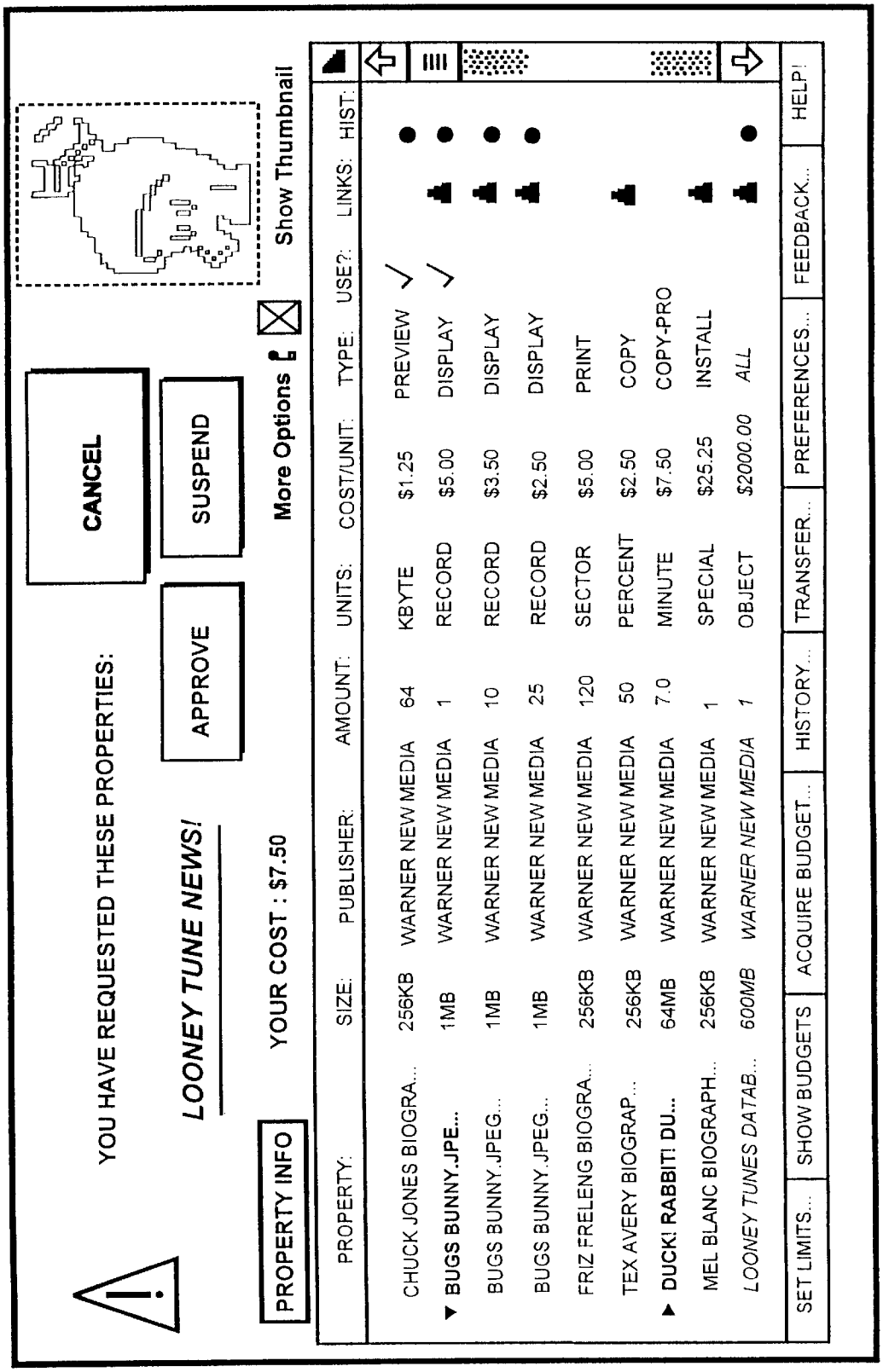

YOU HAVE REQUESTED THESE PROPERTIES:
LOONEY TUNE NEWS!
YOUR COST : $7.50
APPROVE
CANCEL
SUSPEND
More Options
Show Thumbnail
PROPERTY INFO
PROPERTY: SIZE: PUBLISHER: AMOUNT: UNITS: COST/UNIT: TYPE: USE?: LINKS: HIST:
CHUCK JONES BIOGRA... 256KB WARNER NEW MEDIA 64 KBYTE $1.25 PREVIEW
▼ BUGS BUNNY.JPE... 1MB WARNER NEW MEDIA 1 RECORD $5.00 DISPLAY
BUGS BUNNY.JPEG... 1MB WARNER NEW MEDIA 10 RECORD $3.50 DISPLAY
BUGS BUNNY.JPEG... 1MB WARNER NEW MEDIA 25 RECORD $2.50 DISPLAY
FRIZ FRELENG BIOGRA... 256KB WARNER NEW MEDIA 120 SECTOR $5.00 PRINT
TEX AVERY BIOGRAP... 256KB WARNER NEW MEDIA 50 PERCENT $2.50 COPY
▶ DUCK! RABBIT! DU... 64MB WARNER NEW MEDIA 7.0 MINUTE $7.50 COPY-PRO
MEL BLANC BIOGRAPH... 256KB WARNER NEW MEDIA 1 SPECIAL $25.25 INSTALL
LOONEY TUNES DATAB... 600MB WARNER NEW MEDIA 1 OBJECT $2000.00 ALL
SET LIMITS... SHOW BUDGETS ACQUIRE BUDGET... HISTORY... TRANSFER... PREFERENCES... FEEDBACK... HELP!

FIG. 73

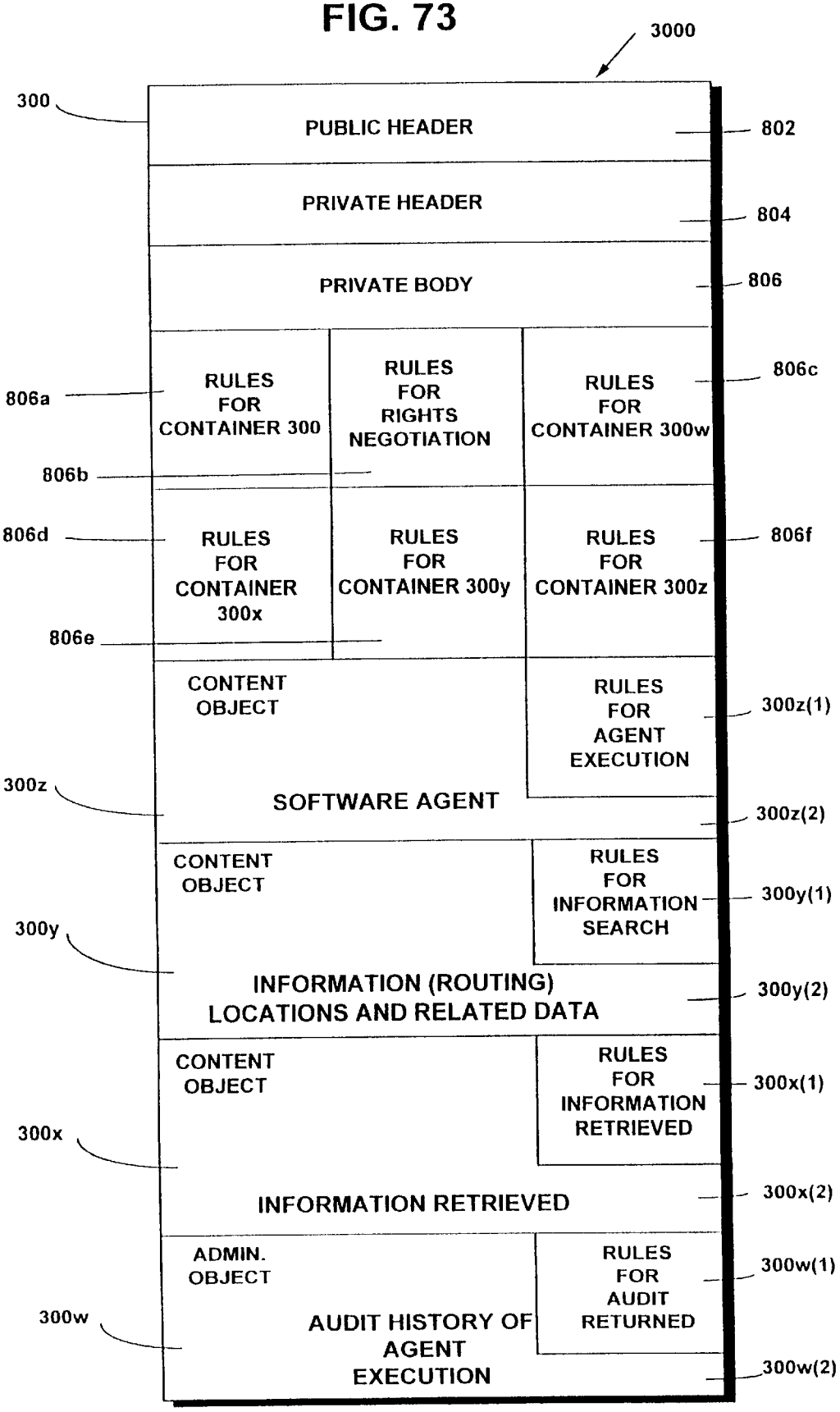

3000
300
PUBLIC HEADER
802
PRIVATE HEADER
804
PRIVATE BODY
806
806a
RULES FOR CONTAINER 300
RULES FOR RIGHTS NEGOTIATION
RULES FOR CONTAINER 300w
806c
806b
806d
RULES FOR CONTAINER 300x
RULES FOR CONTAINER 300y
RULES FOR CONTAINER 300z
806f
806e
CONTENT OBJECT
RULES FOR AGENT EXECUTION
300z(1)
300z
SOFTWARE AGENT
300z(2)
CONTENT OBJECT
RULES FOR INFORMATION SEARCH
300y(1)
300y
INFORMATION (ROUTING) LOCATIONS AND RELATED DATA
300y(2)
CONTENT OBJECT
RULES FOR INFORMATION RETRIEVED
300x(1)
300x
INFORMATION RETRIEVED
300x(2)
ADMIN. OBJECT
RULES FOR AUDIT RETURNED
300w(1)
300w
AUDIT HISTORY OF AGENT EXECUTION
300w(2)

FIG. 74

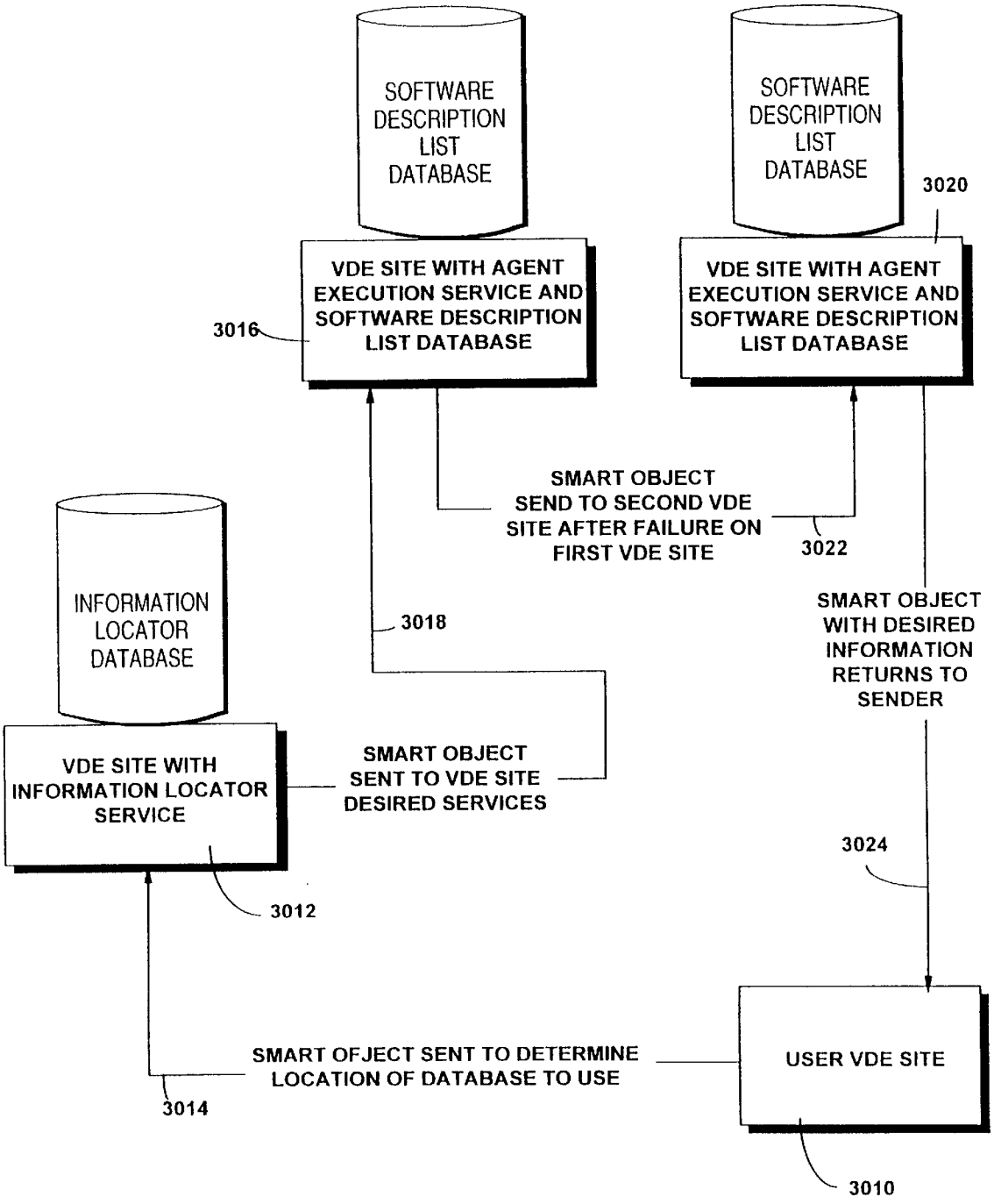

SOFTWARE DESCRIPTION LIST DATABASE
SOFTWARE DESCRIPTION LIST DATABASE
3020
VDE SITE WITH AGENT EXECUTION SERVICE AND SOFTWARE DESCRIPTION LIST DATABASE
3016
VDE SITE WITH AGENT EXECUTION SERVICE AND SOFTWARE DESCRIPTION LIST DATABASE
SMART OBJECT SEND TO SECOND VDE SITE AFTER FAILURE ON FIRST VDE SITE
3022
INFORMATION LOCATOR DATABASE
3018
SMART OBJECT WITH DESIRED INFORMATION RETURNS TO SENDER
VDE SITE WITH INFORMATION LOCATOR SERVICE
SMART OBJECT SENT TO VDE SITE DESIRED SERVICES
3024
3012
SMART OFJECT SENT TO DETERMINE LOCATION OF DATABASE TO USE
USER VDE SITE
3014
3010

FIG. 75A

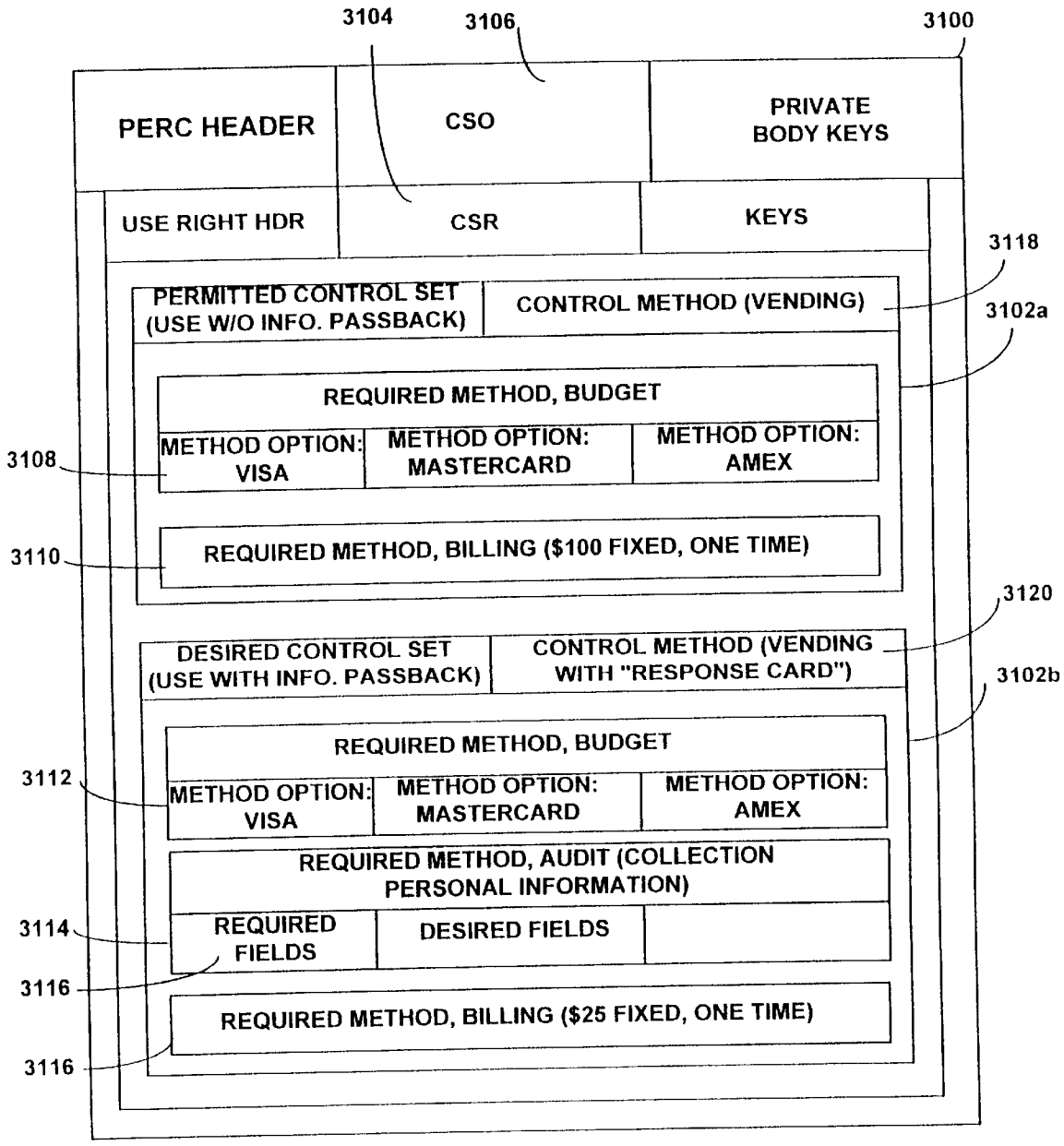

3104
3106
3100
PERC HEADER
CSO
PRIVATE BODY KEYS
USE RIGHT HDR
CSR
KEYS
3118
PERMITTED CONTROL SET (USE W/O INFO. PASSBACK)
CONTROL METHOD (VENDING)
3102a
REQUIRED METHOD, BUDGET
3108
METHOD OPTION: VISA
METHOD OPTION: MASTERCARD
METHOD OPTION: AMEX
3110
REQUIRED METHOD, BILLING ($100 FIXED, ONE TIME)
3120
DESIRED CONTROL SET (USE WITH INFO. PASSBACK)
CONTROL METHOD (VENDING WITH "RESPONSE CARD")
3102b
REQUIRED METHOD, BUDGET
3112
METHOD OPTION: VISA
METHOD OPTION: MASTERCARD
METHOD OPTION: AMEX
REQUIRED METHOD, AUDIT (COLLECTION PERSONAL INFORMATION)
3114
REQUIRED FIELDS
DESIRED FIELDS
3116
REQUIRED METHOD, BILLING ($25 FIXED, ONE TIME)
3116

FIG. 75B

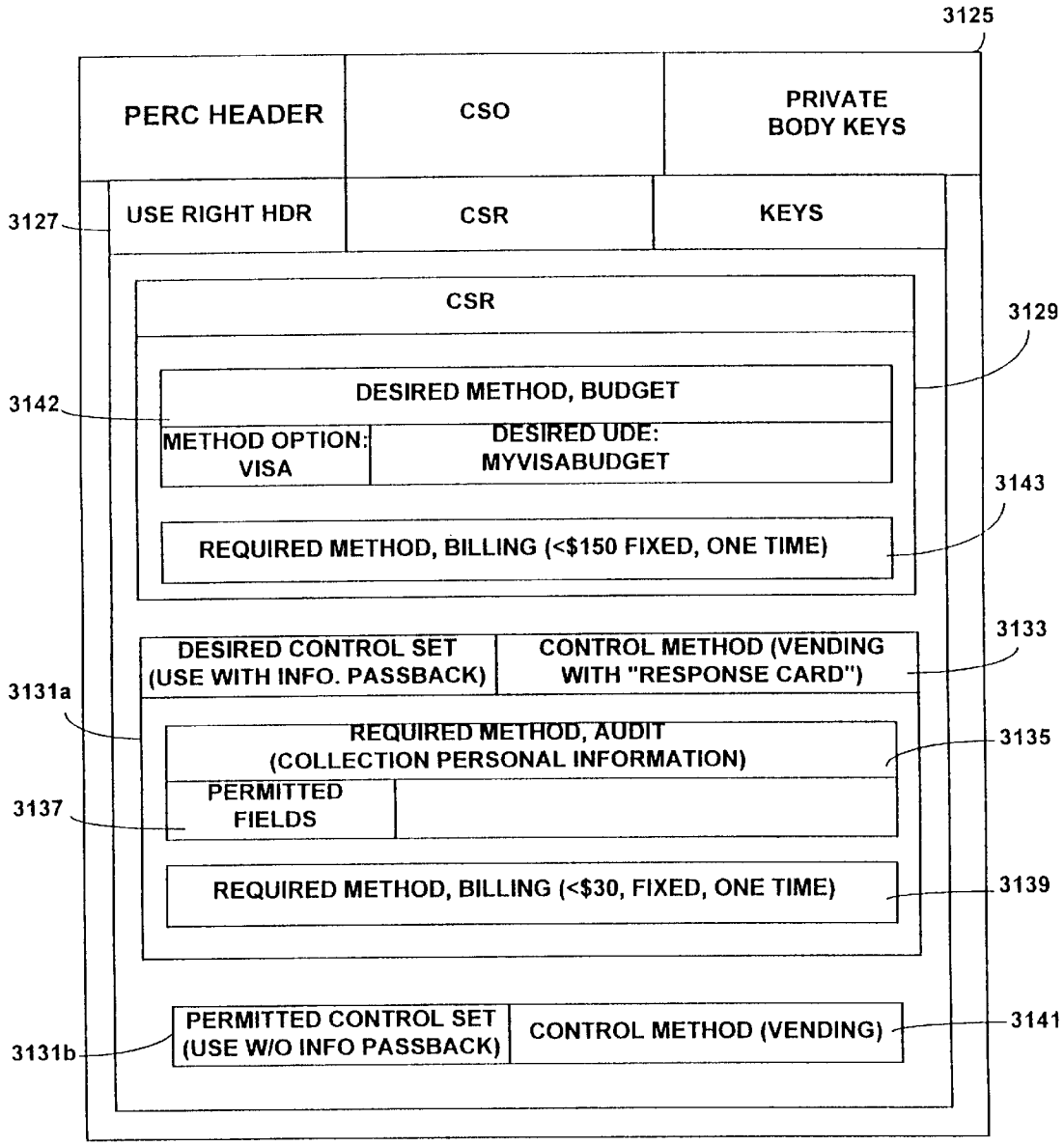

3125
PERC HEADER
CSO
PRIVATE BODY KEYS
3127
USE RIGHT HDR
CSR
KEYS
CSR
3129
DESIRED METHOD, BUDGET
3142
METHOD OPTION: VISA
DESIRED UDE: MYVISABUDGET
3143
REQUIRED METHOD, BILLING (<$150 FIXED, ONE TIME)
3133
DESIRED CONTROL SET (USE WITH INFO. PASSBACK)
CONTROL METHOD (VENDING WITH "RESPONSE CARD")
3131a
REQUIRED METHOD, AUDIT (COLLECTION PERSONAL INFORMATION)
3135
3137
PERMITTED FIELDS
REQUIRED METHOD, BILLING (<$30, FIXED, ONE TIME)
3139
3131b
PERMITTED CONTROL SET (USE W/O INFO PASSBACK)
CONTROL METHOD (VENDING)
3141

FIG. 75C

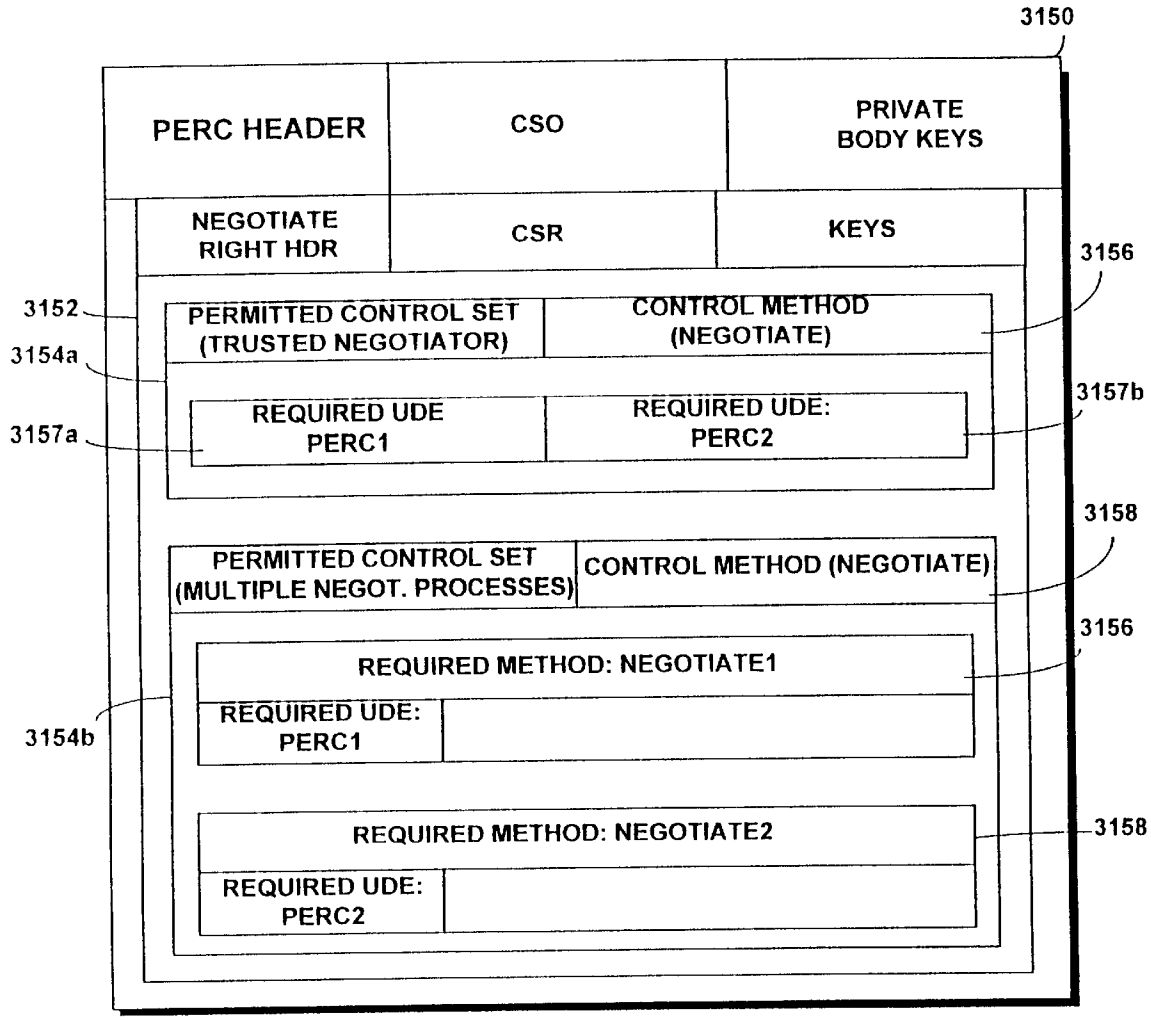
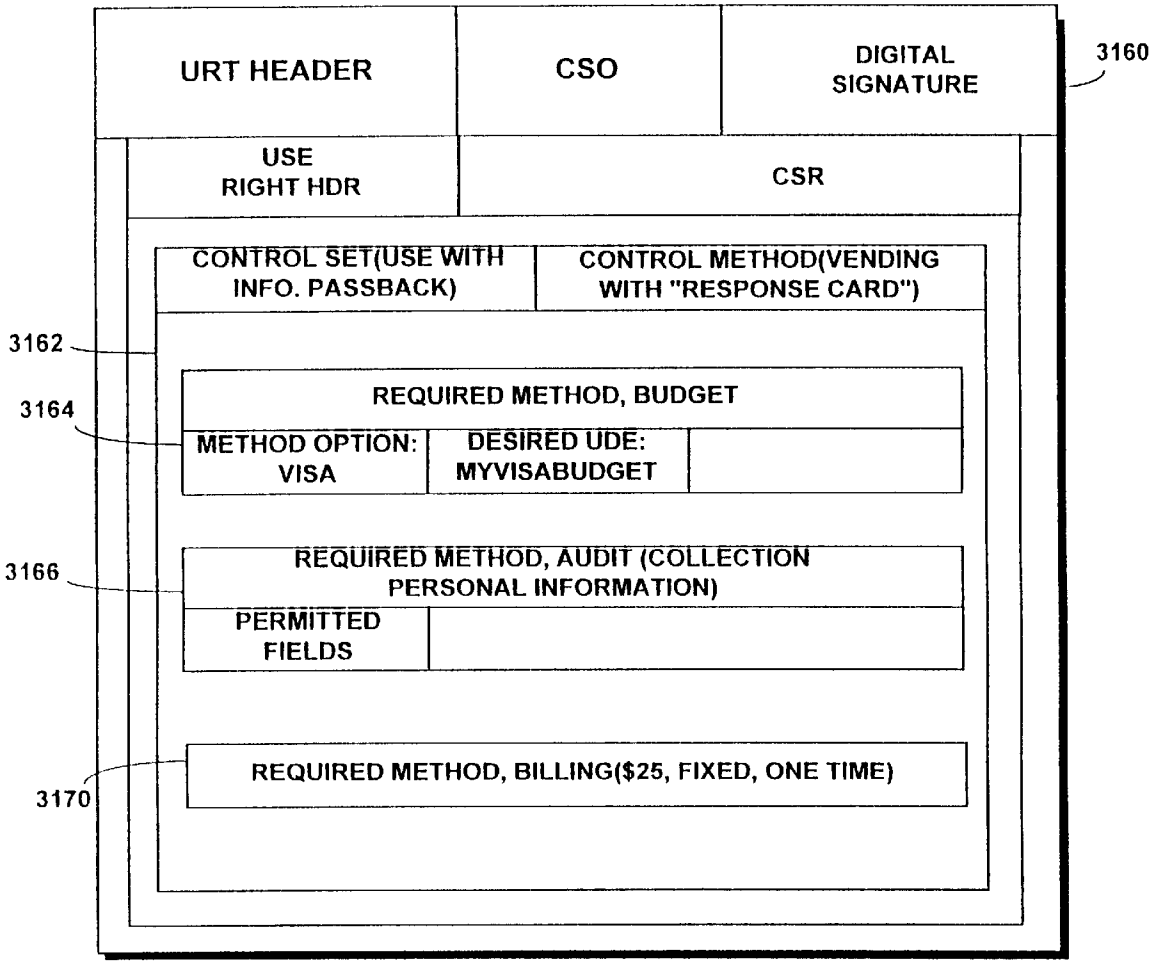

3150
PERC HEADER
CSO
PRIVATE BODY KEYS
NEGOTIATE RIGHT HDR
CSR
KEYS
3156
3152
PERMITTED CONTROL SET (TRUSTED NEGOTIATOR)
CONTROL METHOD (NEGOTIATE)
3154a
3157b
REQUIRED UDE PERC1
REQUIRED UDE: PERC2
3157a
3158
PERMITTED CONTROL SET (MULTIPLE NEGOT. PROCESSES)
CONTROL METHOD (NEGOTIATE)
3156
REQUIRED METHOD: NEGOTIATE1
REQUIRED UDE: PERC1
3154b
REQUIRED METHOD: NEGOTIATE2
3158
REQUIRED UDE: PERC2

FIG. 75D


URT HEADER
CSO
DIGITAL SIGNATURE
3160
USE RIGHT HDR
CSR
CONTROL SET(USE WITH INFO. PASSBACK)
CONTROL METHOD(VENDING WITH "RESPONSE CARD")
3162
REQUIRED METHOD, BUDGET
3164
METHOD OPTION: VISA
DESIRED UDE: MYVISABUDGET
REQUIRED METHOD, AUDIT (COLLECTION PERSONAL INFORMATION)
3166
PERMITTED FIELDS
REQUIRED METHOD, BILLING($25, FIXED, ONE TIME)
3170

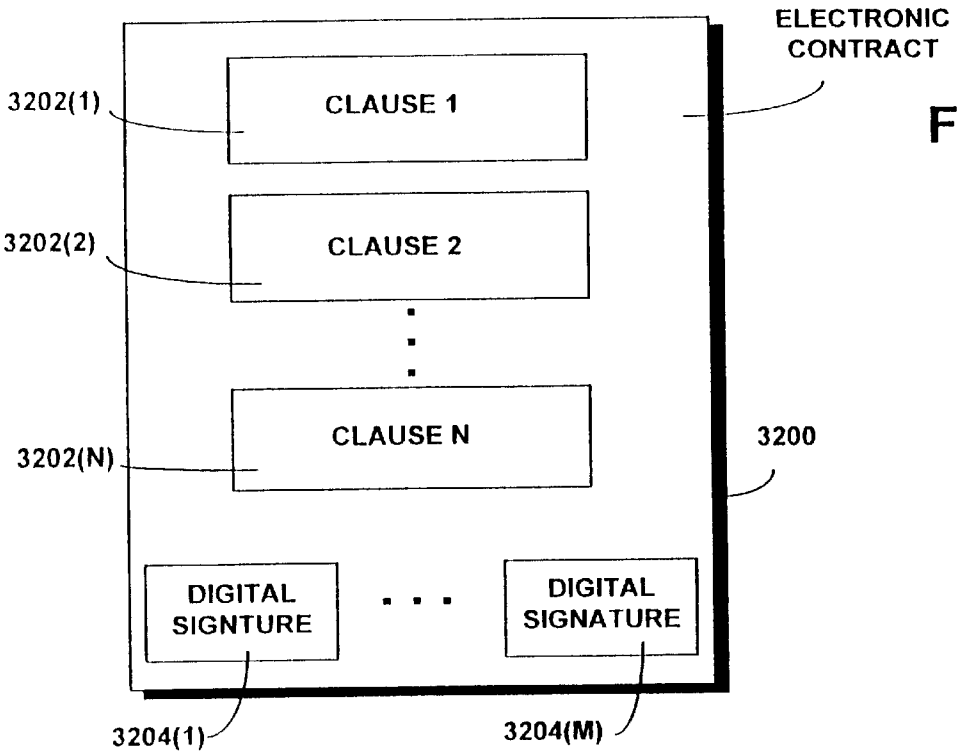


FIG. 75E

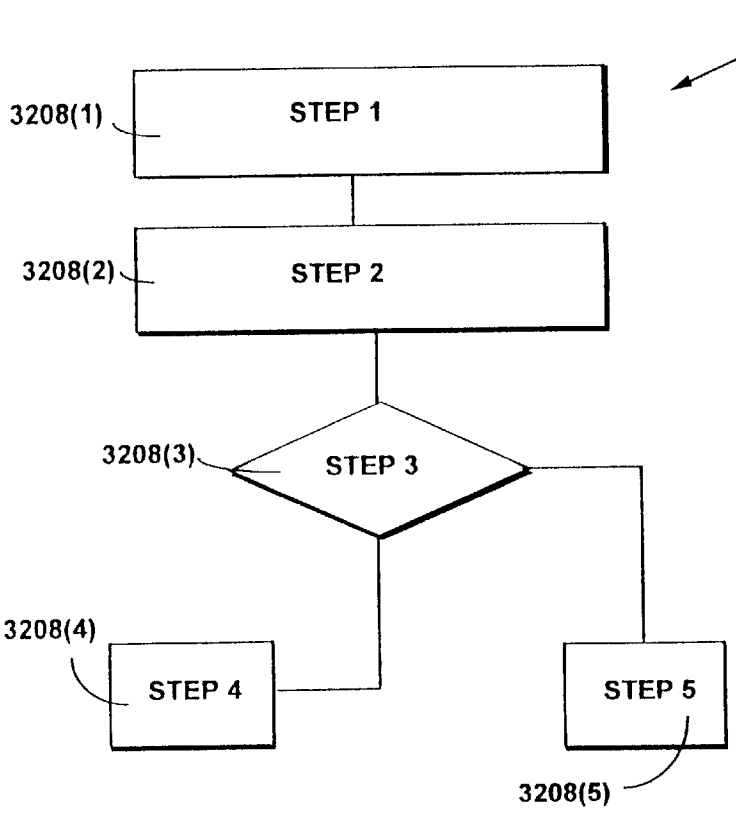


FIG. 75F

FIG. 76A

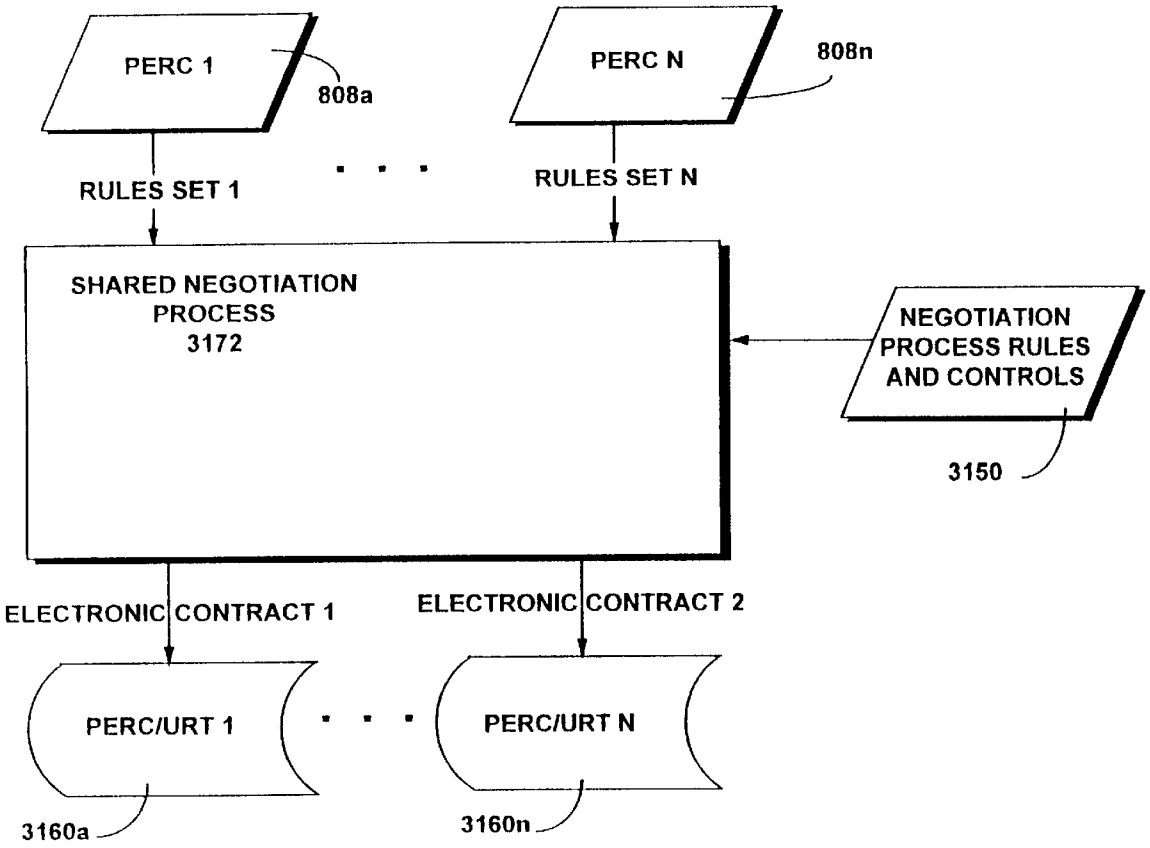

PERC 1
808a
PERC N
808n
RULES SET 1
RULES SET N
SHARED NEGOTIATION PROCESS 3172
NEGOTIATION PROCESS RULES AND CONTROLS
3150
ELECTRONIC CONTRACT 1
ELECTRONIC CONTRACT 2
PERC/URT 1
PERC/URT N
3160a
3160n

FIG. 76B

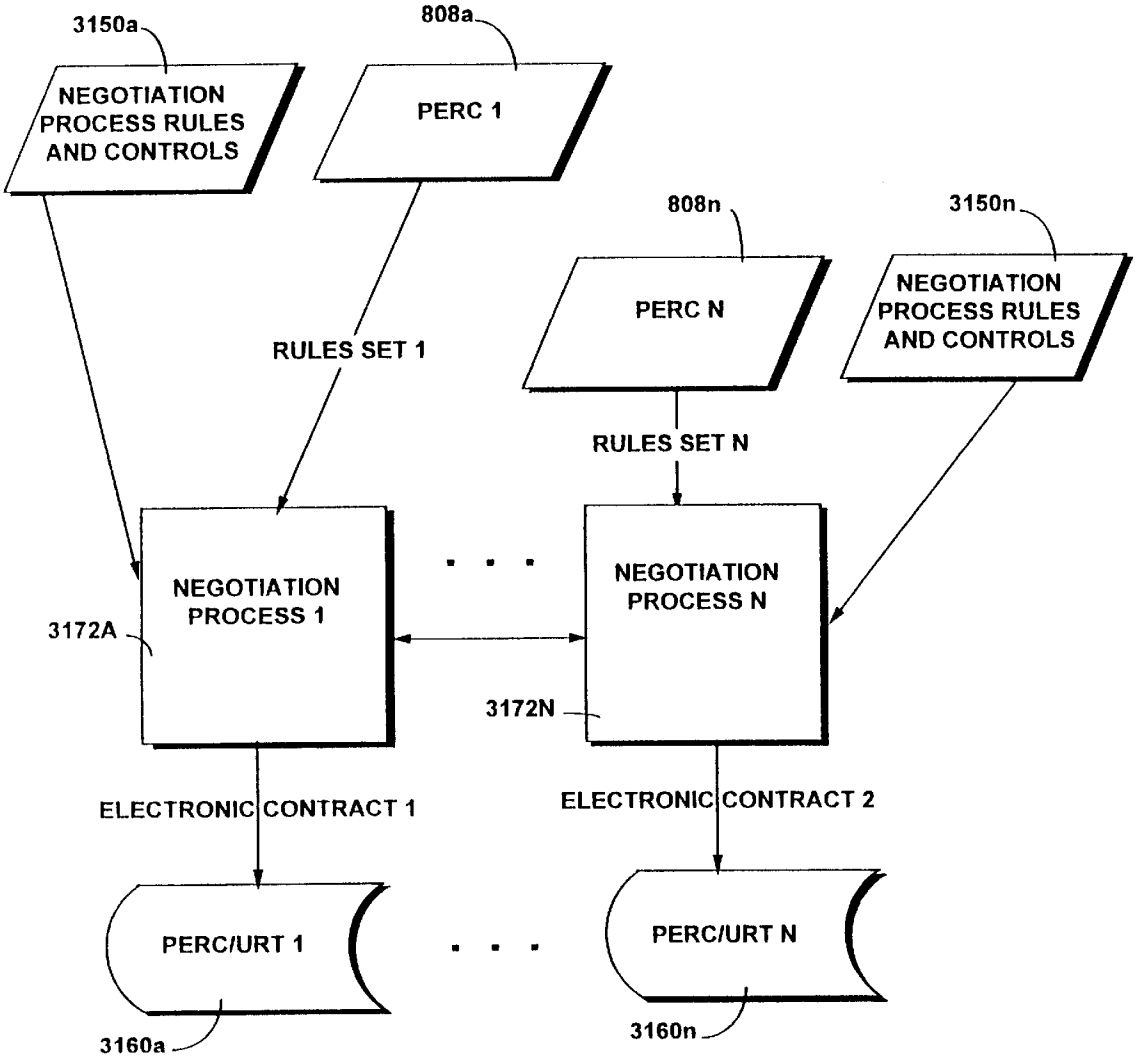
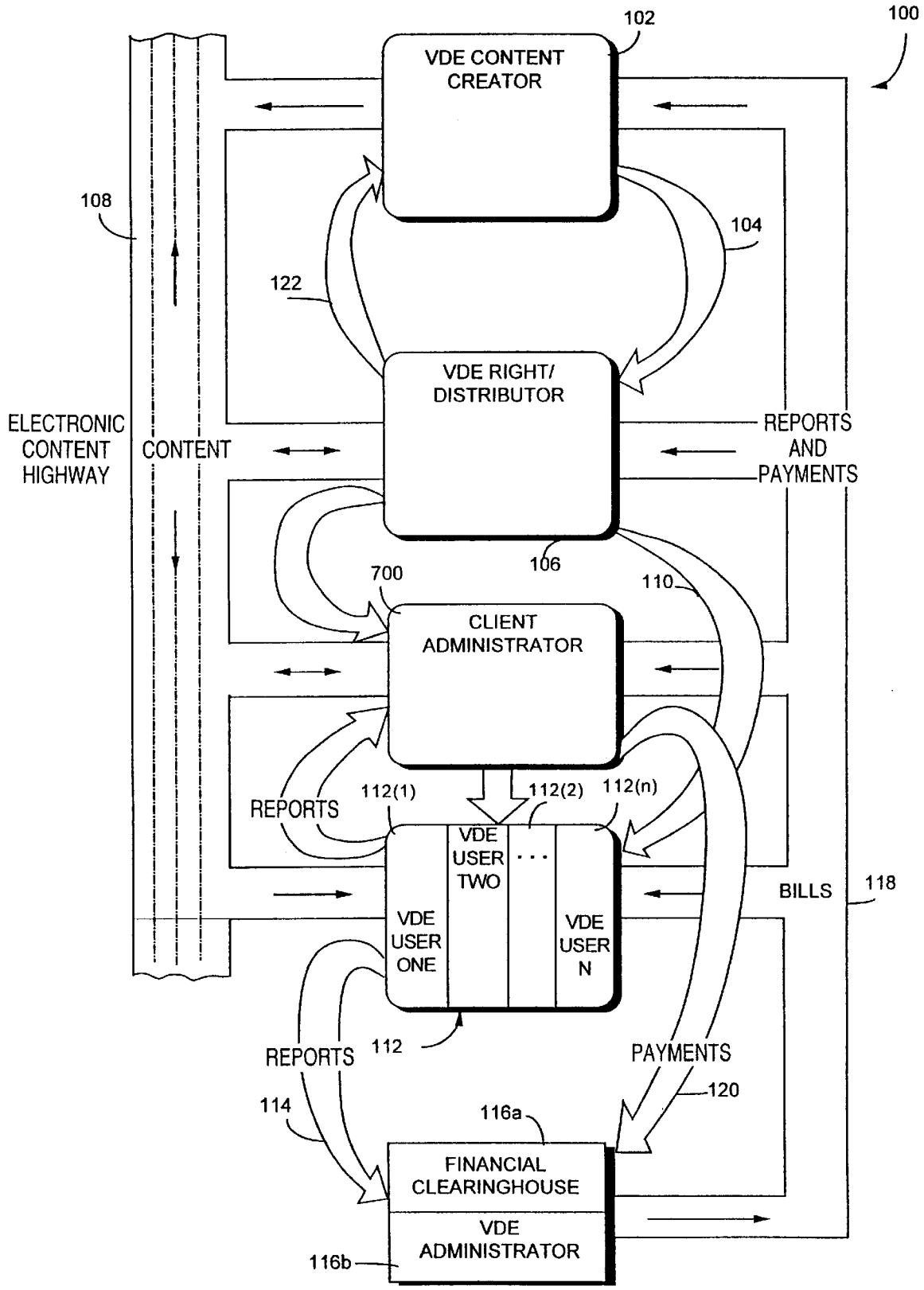

3150a
808a
NEGOTIATION PROCESS RULES AND CONTROLS
PERC 1
808n
3150n
PERC N
NEGOTIATION PROCESS RULES AND CONTROLS
RULES SET 1
RULES SET N
NEGOTIATION PROCESS 1
3172A
NEGOTIATION PROCESS N
3172N
ELECTRONIC CONTRACT 1
ELECTRONIC CONTRACT 2
PERC/URT 1
PERC/URT N
3160a
3160n

FIG. 77


102
100
VDE CONTENT CREATOR
108
104
122
VDE RIGHT/ DISTRIBUTOR
ELECTRONIC CONTENT HIGHWAY
CONTENT
REPORTS AND PAYMENTS
106
700
110
CLIENT ADMINISTRATOR
REPORTS
112(1)
112(2)
112(n)
VDE USER TWO
VDE USER ONE
VDE USER N
BILLS
118
112
REPORTS
PAYMENTS
120
114
116a
FINANCIAL CLEARINGHOUSE
VDE ADMINISTRATOR
116b

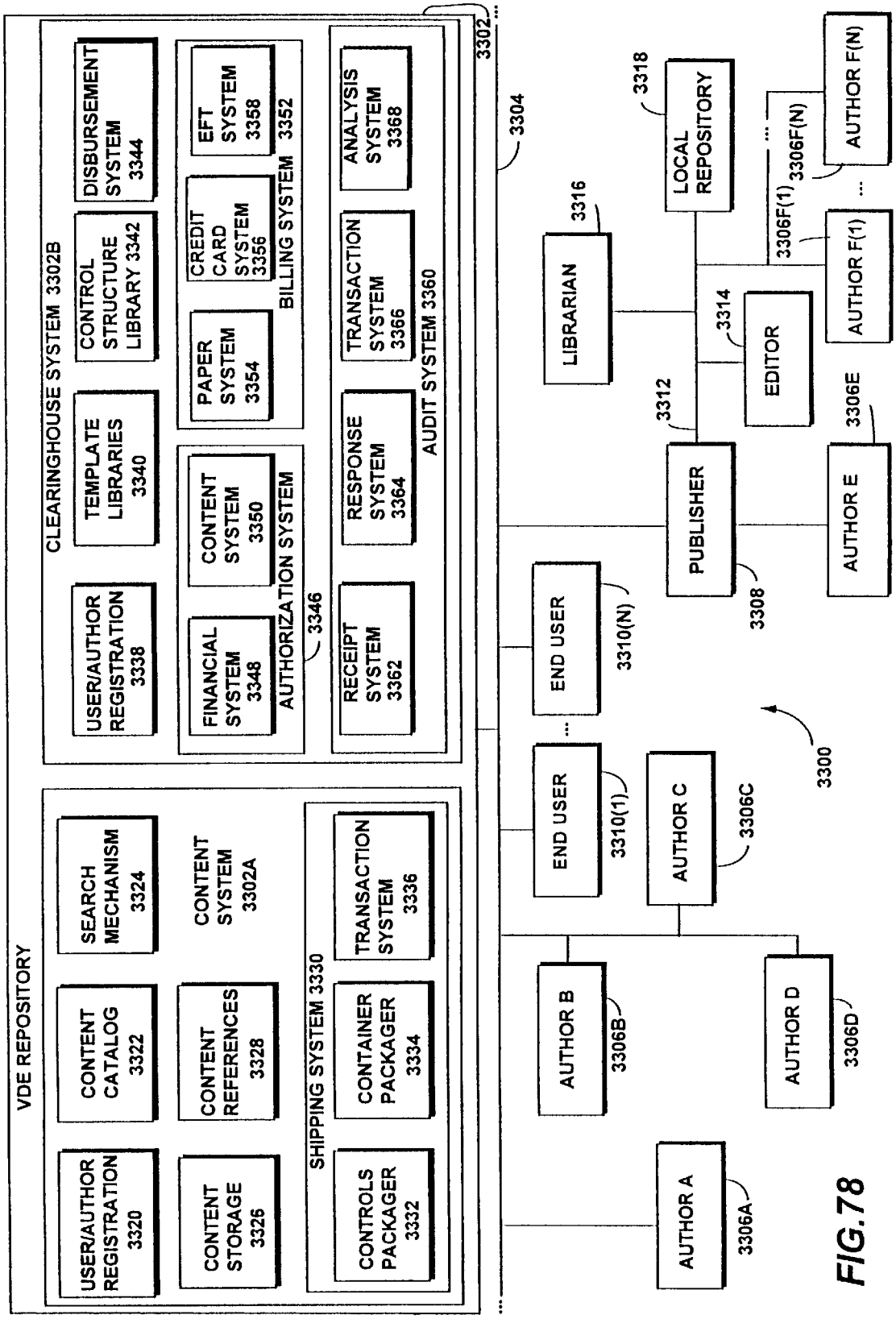

VDE REPOSITORY
USER/AUTHOR REGISTRATION 3320
CONTENT CATALOG 3322
SEARCH MECHANISM 3324
CONTENT STORAGE 3326
CONTENT REFERENCES 3328
CONTENT SYSTEM 3302A
CONTROLS PACKAGER 3332
CONTAINER PACKAGER 3334
SHIPPING SYSTEM 3330
TRANSACTION SYSTEM 3336
CLEARINGHOUSE SYSTEM 3302B
USER/AUTHOR REGISTRATION 3338
TEMPLATE LIBRARIES 3340
CONTROL STRUCTURE LIBRARY 3342
DISBURSEMENT SYSTEM 3344
FINANCIAL SYSTEM 3348
CONTENT SYSTEM 3350
AUTHORIZATION SYSTEM 3346
PAPER SYSTEM 3354
CREDIT CARD SYSTEM 3356
EFT SYSTEM 3358
BILLING SYSTEM 3352
RECEIPT SYSTEM 3362
RESPONSE SYSTEM 3364
TRANSACTION SYSTEM 3366
ANALYSIS SYSTEM 3368
AUDIT SYSTEM 3360
3302
3304
LIBRARIAN 3316
3318
LOCAL REPOSITORY
3306F(1)
3306F(N)
AUTHOR F(1)
AUTHOR F(N)
3314
EDITOR
3312
PUBLISHER
3308
AUTHOR E
3306E
END USER
3310(N)
END USER
3310(1)
AUTHOR C
3306C
3300
AUTHOR B
3306B
AUTHOR D
3306D
AUTHOR A
3306A

FIG. 78

FIG. 79

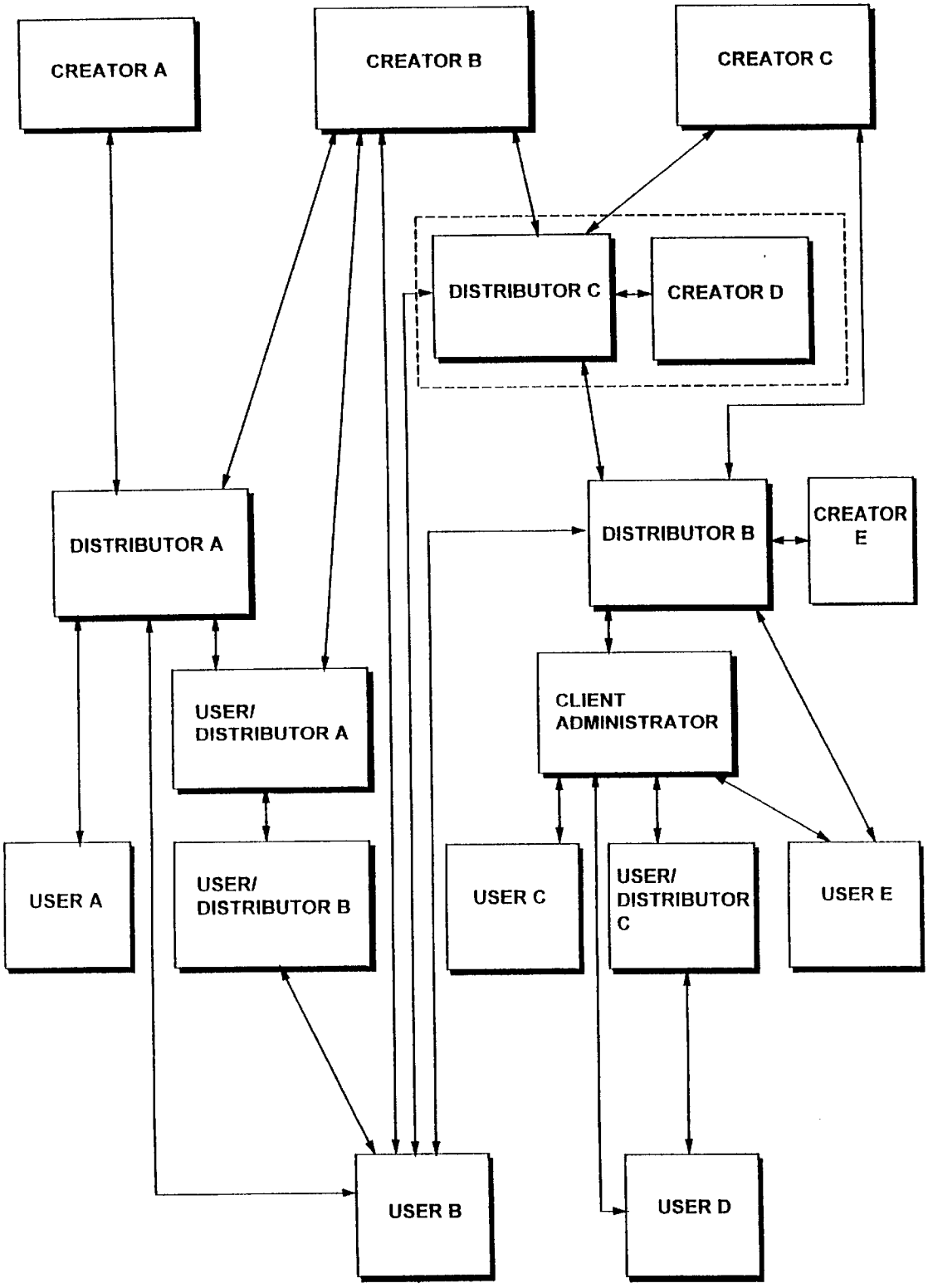
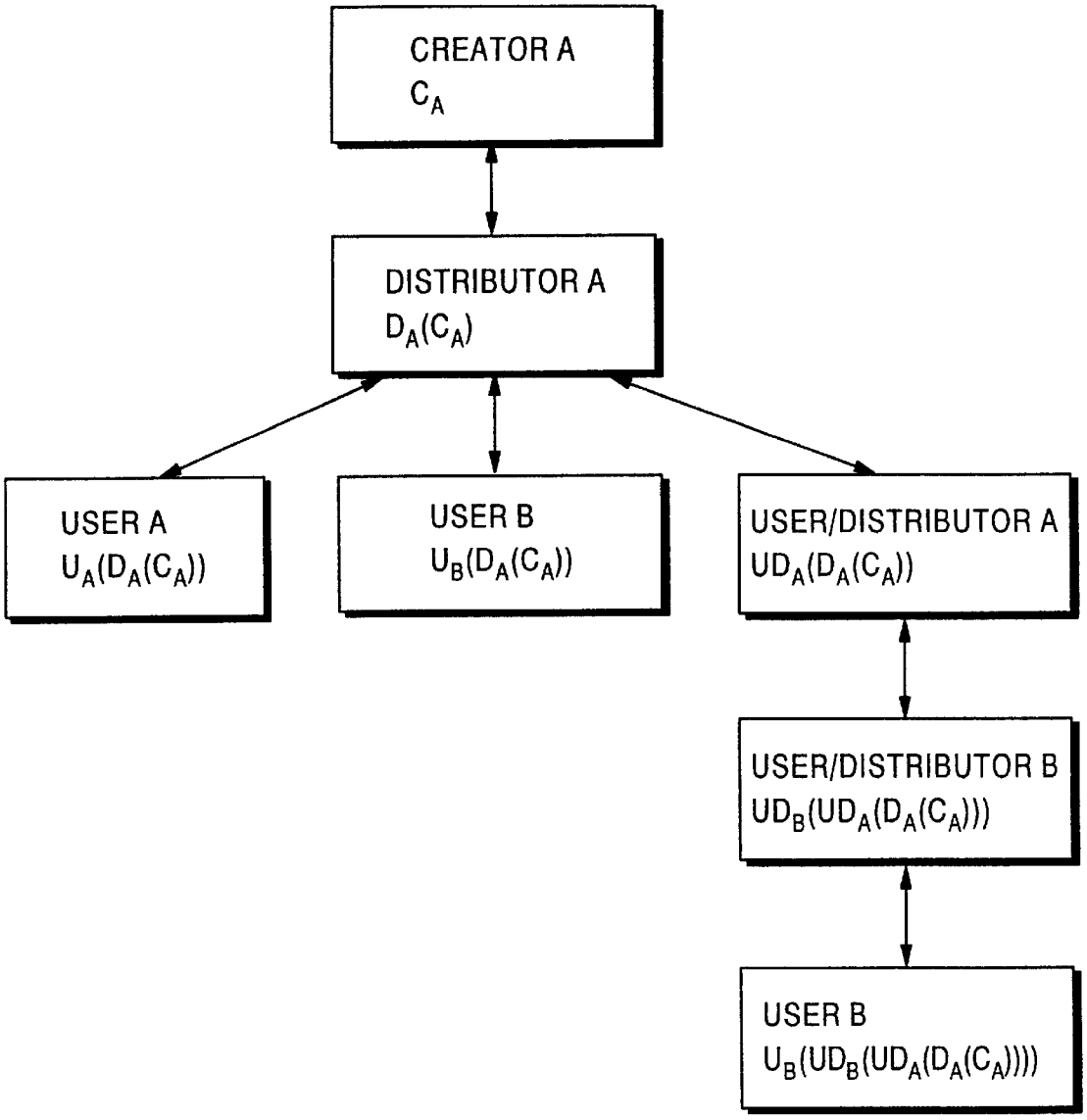

CREATOR A
CREATOR B
CREATOR C
DISTRIBUTOR C
CREATOR D
DISTRIBUTOR A
DISTRIBUTOR B
CREATOR E
USER/ DISTRIBUTOR A
CLIENT ADMINISTRATOR
USER A
USER/ DISTRIBUTOR B
USER C
USER/ DISTRIBUTOR C
USER E
USER B
USER D

FIG. 80


CREATOR A
C_A
DISTRIBUTOR A
$D_A(C_A)$
USER A
$U_A(D_A(C_A))$
USER B
$U_B(D_A(C_A))$
USER/DISTRIBUTOR A
$UD_A(D_A(C_A))$
USER/DISTRIBUTOR B
$UD_B(UD_A(D_A(C_A)))$
USER B
$U_B(UD_B(UD_A(D_A(C_A))))$

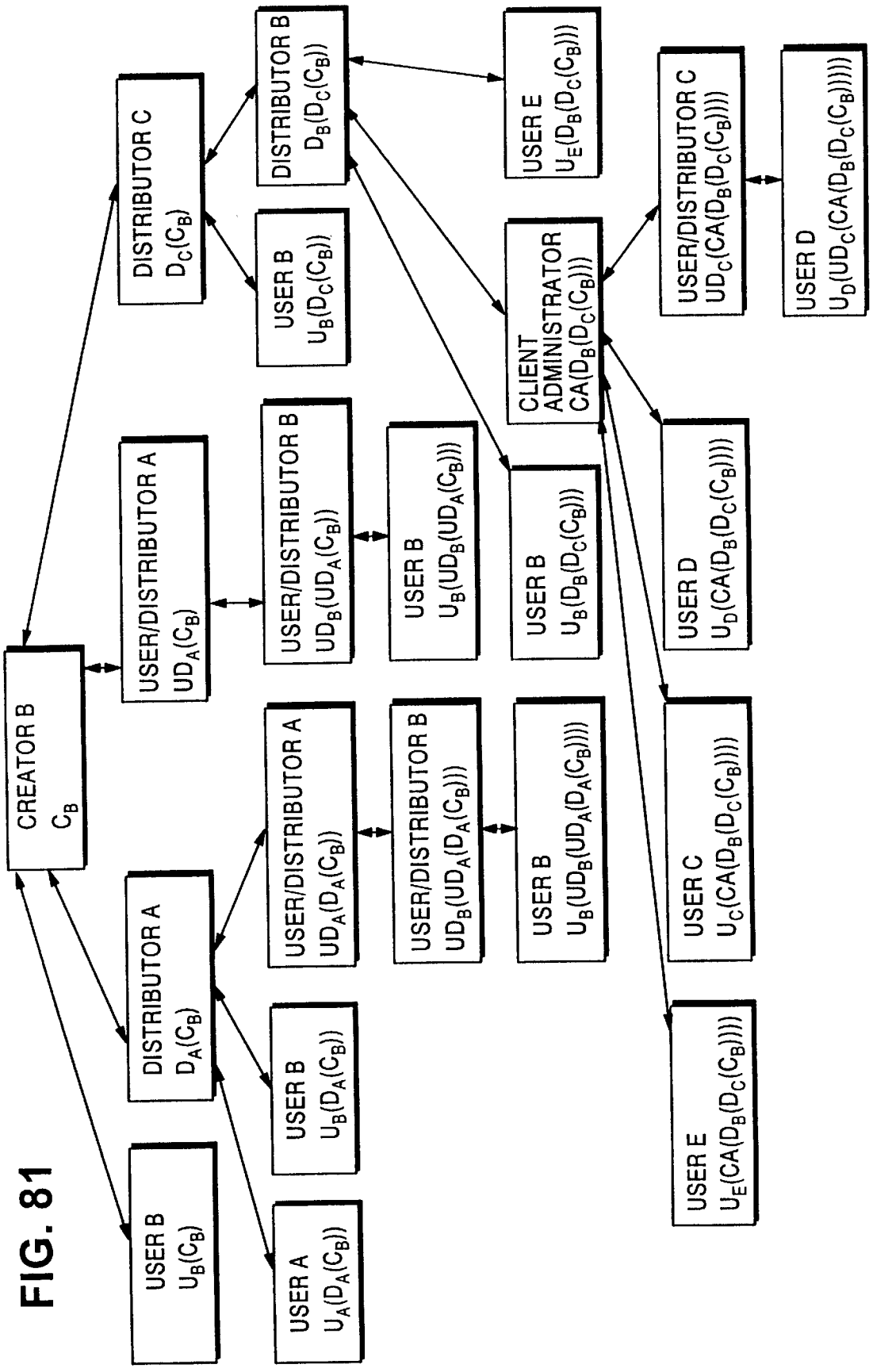

CREATOR B
C_B
DISTRIBUTOR C
$D_C(C_B)$
DISTRIBUTOR B
$D_B(D_C(C_B))$
USER E
$U_E(D_B(D_C(C_B)))$
USER/DISTRIBUTOR C
$UD_C(CA(D_B(D_C(C_B))))$
USER D
$U_D(UD_C(CA(D_B(D_C(C_B)))))$
USER B
$U_B(D_C(C_B))$
CLIENT ADMINISTRATOR
$CA(D_B(D_C(C_B)))$
USER/DISTRIBUTOR A
$UD_A(C_B)$
USER/DISTRIBUTOR B
$UD_B(UD_A(C_B))$
USER B
$U_B(UD_A(C_B))$
USER B
$U_B(D_B(D_C(C_B)))$
USER D
$U_D(CA(D_B(D_C(C_B))))$
DISTRIBUTOR A
$D_A(C_B)$
USER/DISTRIBUTOR A
$UD_A(D_A(C_B))$
USER/DISTRIBUTOR B
$UD_B(UD_A(D_A(C_B)))$
USER B
$U_B(UD_B(UD_A(D_A(C_B))))$
USER C
$U_C(CA(D_B(D_C(C_B))))$
USER B
$U_B(C_B)$
USER B
$U_B(D_A(C_B))$
USER A
$U_A(D_A(C_B))$
USER E
$U_E(CA(D_B(D_C(C_B))))$

FIG. 81

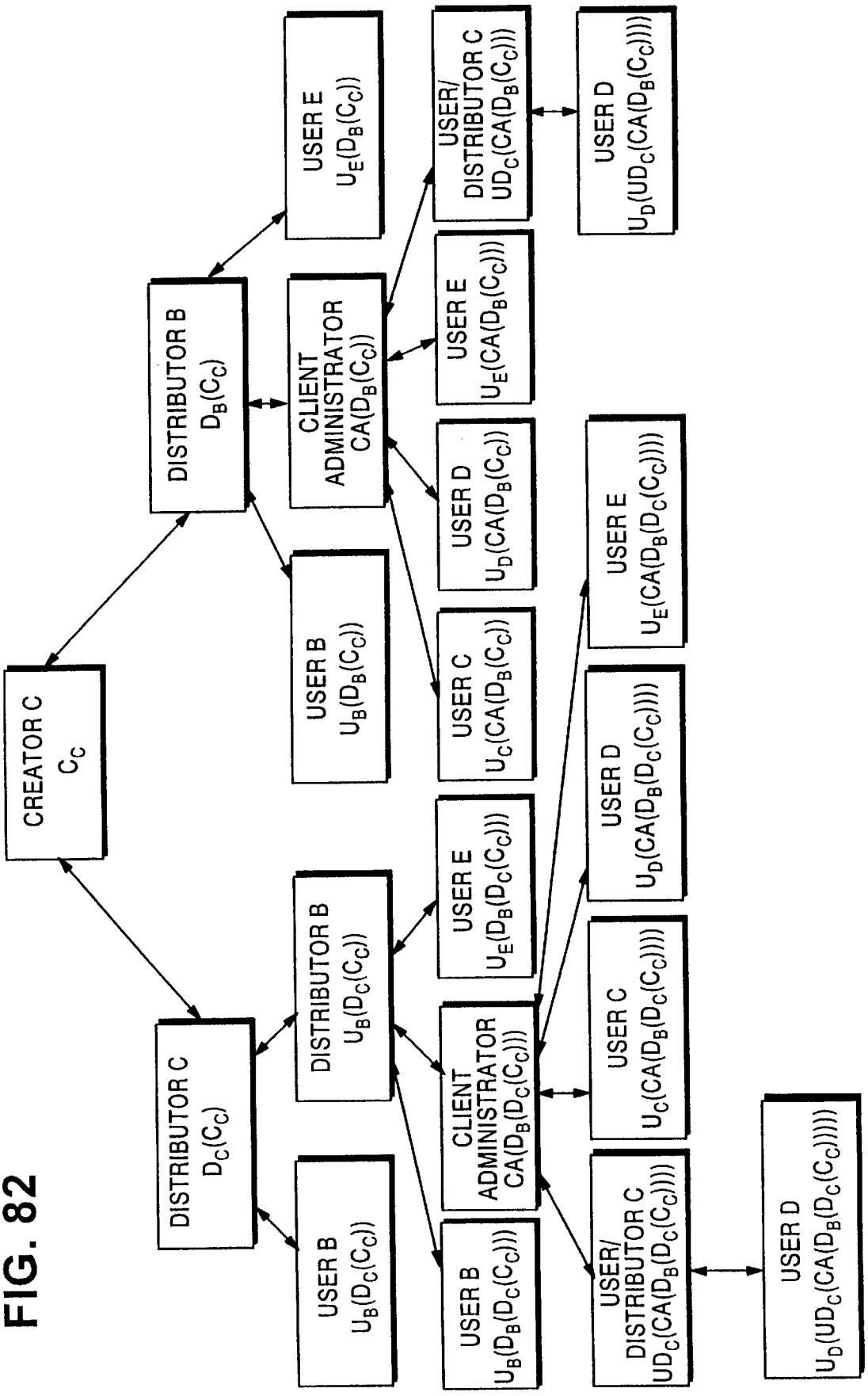
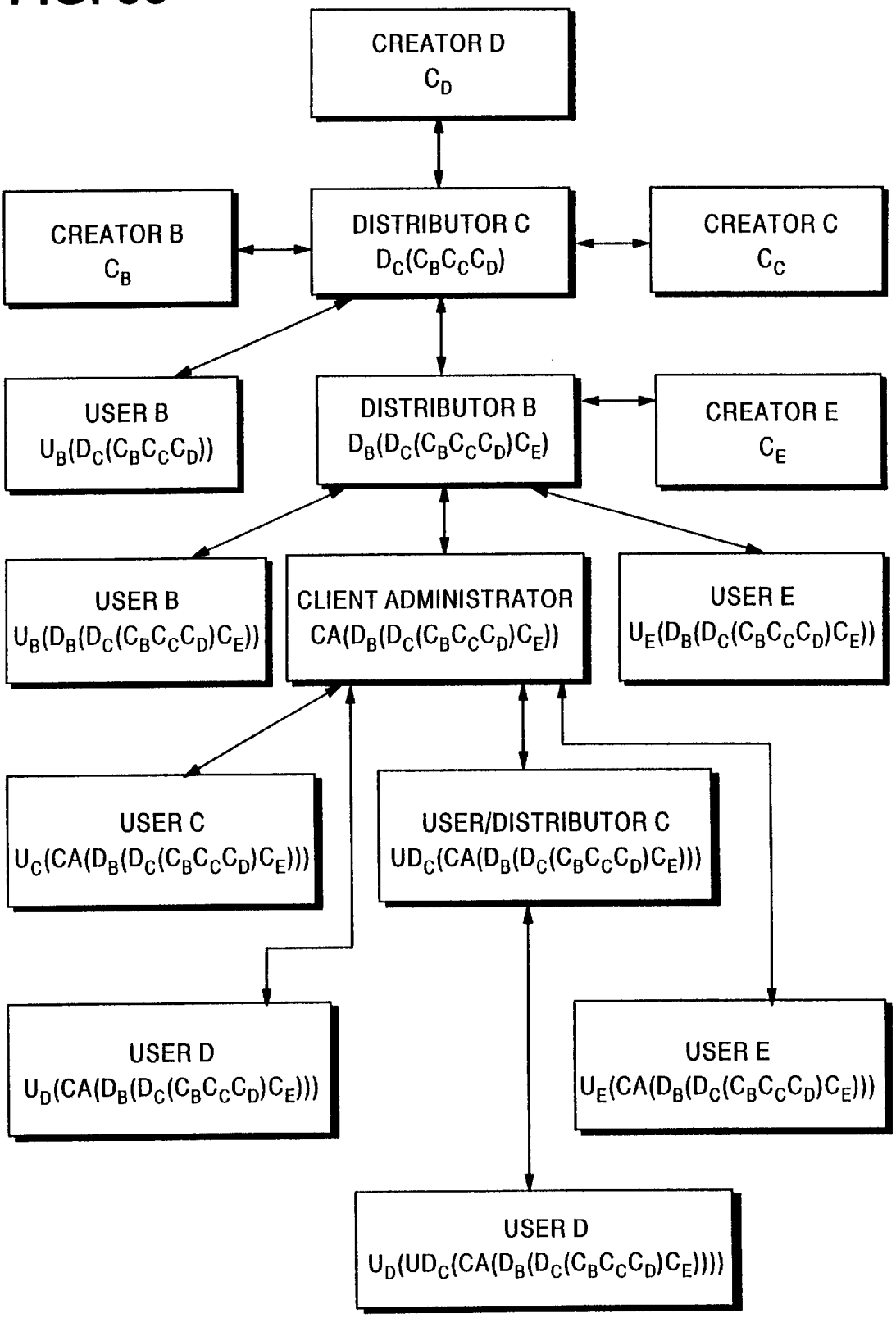

CREATOR C
C_C
DISTRIBUTOR B
$D_B(C_C)$
DISTRIBUTOR C
$D_C(C_C)$
USER E
$U_E(D_B(C_C))$
USER B
$U_B(D_B(C_C))$
CLIENT ADMINISTRATOR
$CA(D_B(C_C))$
USER/ DISTRIBUTOR C
$UD_C(CA(D_B(C_C)))$
USER D
$U_D(UD_C(CA(D_B(C_C))))$
USER E
$U_E(CA(D_B(C_C)))$
USER D
$U_D(CA(D_B(C_C)))$
USER C
$U_C(CA(D_B(C_C)))$
DISTRIBUTOR B
$U_B(D_C(C_C))$
USER B
$U_B(D_C(C_C))$
USER E
$U_E(D_B(D_C(C_C)))$
CLIENT ADMINISTRATOR
$CA(D_B(D_C(C_C)))$
USER E
$U_E(CA(D_B(D_C(C_C))))$
USER D
$U_D(CA(D_B(D_C(C_C))))$
USER C
$U_C(CA(D_B(D_C(C_C))))$
USER/ DISTRIBUTOR C
$UD_C(CA(D_B(D_C(C_C))))$
USER D
$U_D(UD_C(CA(D_B(D_C(C_C)))))$

FIG. 82

FIG. 83


CREATOR D
C_D
CREATOR B
C_B
DISTRIBUTOR C
$D_C(C_BC_CC_D)$
CREATOR C
C_C
USER B
$U_B(D_C(C_BC_CC_D))$
DISTRIBUTOR B
$D_B(D_C(C_BC_CC_D)C_E)$
CREATOR E
C_E
USER B
$U_B(D_B(D_C(C_BC_CC_D)C_E))$
CLIENT ADMINISTRATOR
$CA(D_B(D_C(C_BC_CC_D)C_E))$
USER E
$U_E(D_B(D_C(C_BC_CC_D)C_E))$
USER C
$U_C(CA(D_B(D_C(C_BC_CC_D)C_E)))$
USER/DISTRIBUTOR C
$UD_C(CA(D_B(D_C(C_BC_CC_D)C_E)))$
USER D
$U_D(CA(D_B(D_C(C_BC_CC_D)C_E)))$
USER E
$U_E(CA(D_B(D_C(C_BC_CC_D)C_E)))$
USER D
$U_D(UD_C(CA(D_B(D_C(C_BC_CC_D)C_E))))$

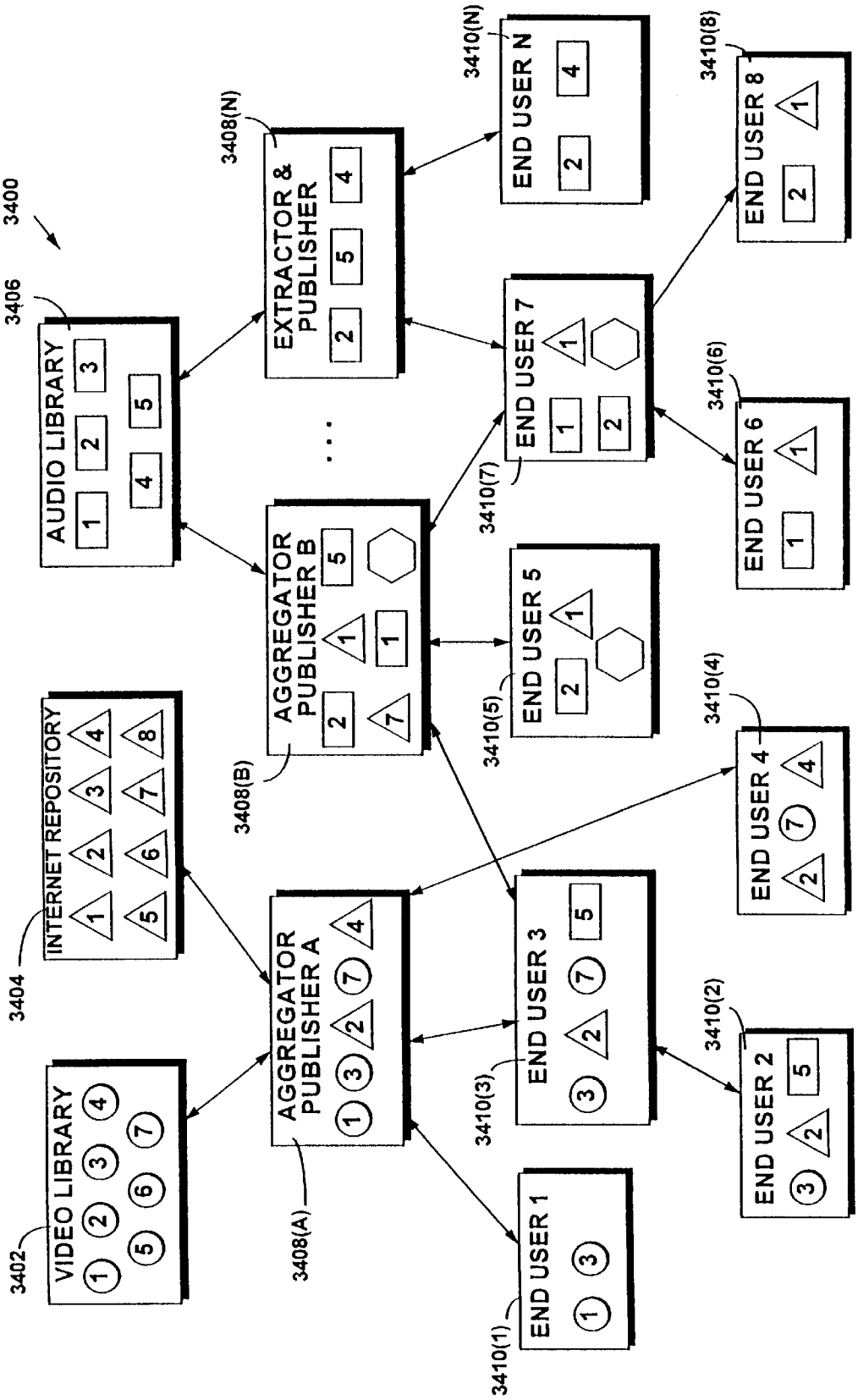
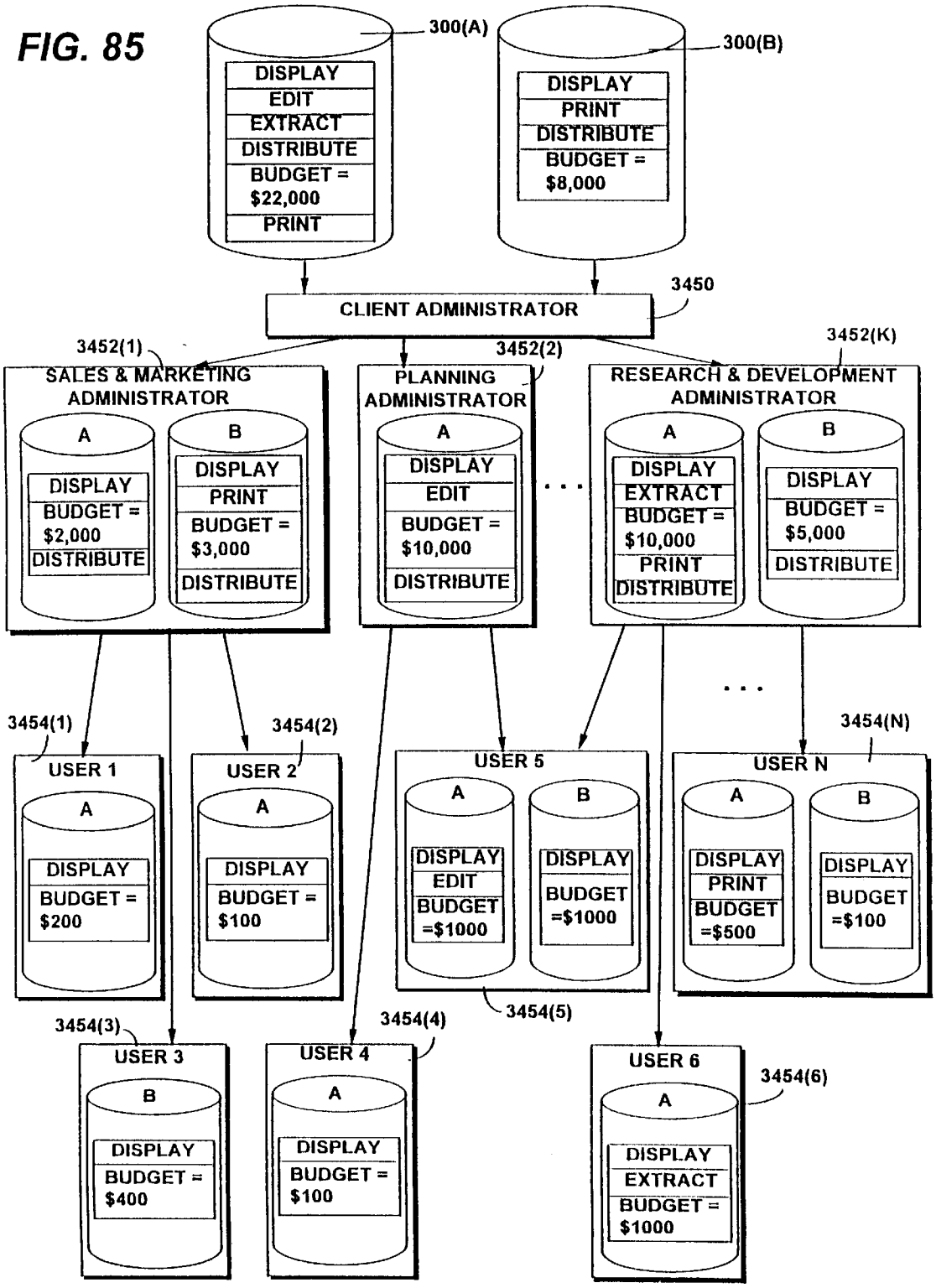

3400
3406
AUDIO LIBRARY
1
2
3
4
5
3408(N)
EXTRACTOR & PUBLISHER
2
4
5
3410(N)
END USER N
2
4
3410(8)
END USER 8
1
2
3410(7)
END USER 7
1
2
3410(6)
END USER 6
1
3408(B)
AGGREGATOR PUBLISHER B
1
2
5
7
3410(5)
END USER 5
1
2
3404
INTERNET REPOSITORY
1
2
3
4
5
6
7
8
3410(4)
END USER 4
2
4
7
3402
VIDEO LIBRARY
1
2
3
4
5
6
7
3408(A)
AGGREGATOR PUBLISHER A
1
2
3
4
7
3410(3)
END USER 3
2
3
5
7
3410(2)
END USER 2
2
3
5
3410(1)
END USER 1
1
3

FIG. 84

FIG. 85


300(A)
DISPLAY
EDIT
EXTRACT
DISTRIBUTE
BUDGET = $22,000
PRINT
300(B)
DISPLAY
PRINT
DISTRIBUTE
BUDGET = $8,000
3450
CLIENT ADMINISTRATOR
3452(1)
SALES & MARKETING ADMINISTRATOR
3452(2)
PLANNING ADMINISTRATOR
3452(K)
RESEARCH & DEVELOPMENT ADMINISTRATOR
A
DISPLAY
BUDGET = $2,000
DISTRIBUTE
B
DISPLAY
PRINT
BUDGET = $3,000
DISTRIBUTE
A
DISPLAY
EDIT
BUDGET = $10,000
DISTRIBUTE
A
DISPLAY
EXTRACT
BUDGET = $10,000
PRINT
DISTRIBUTE
B
DISPLAY
BUDGET = $5,000
DISTRIBUTE
3454(1)
USER 1
3454(2)
USER 2
USER 5
3454(N)
USER N
DISPLAY
BUDGET = $200
DISPLAY
BUDGET = $100
DISPLAY
EDIT
BUDGET =$1000
DISPLAY
BUDGET =$1000
DISPLAY
PRINT
BUDGET =$500
DISPLAY
BUDGET =$100
3454(5)
3454(3)
USER 3
3454(4)
USER 4
USER 6
3454(6)
DISPLAY
BUDGET = $400
DISPLAY
BUDGET = $100
DISPLAY
EXTRACT
BUDGET = $1000

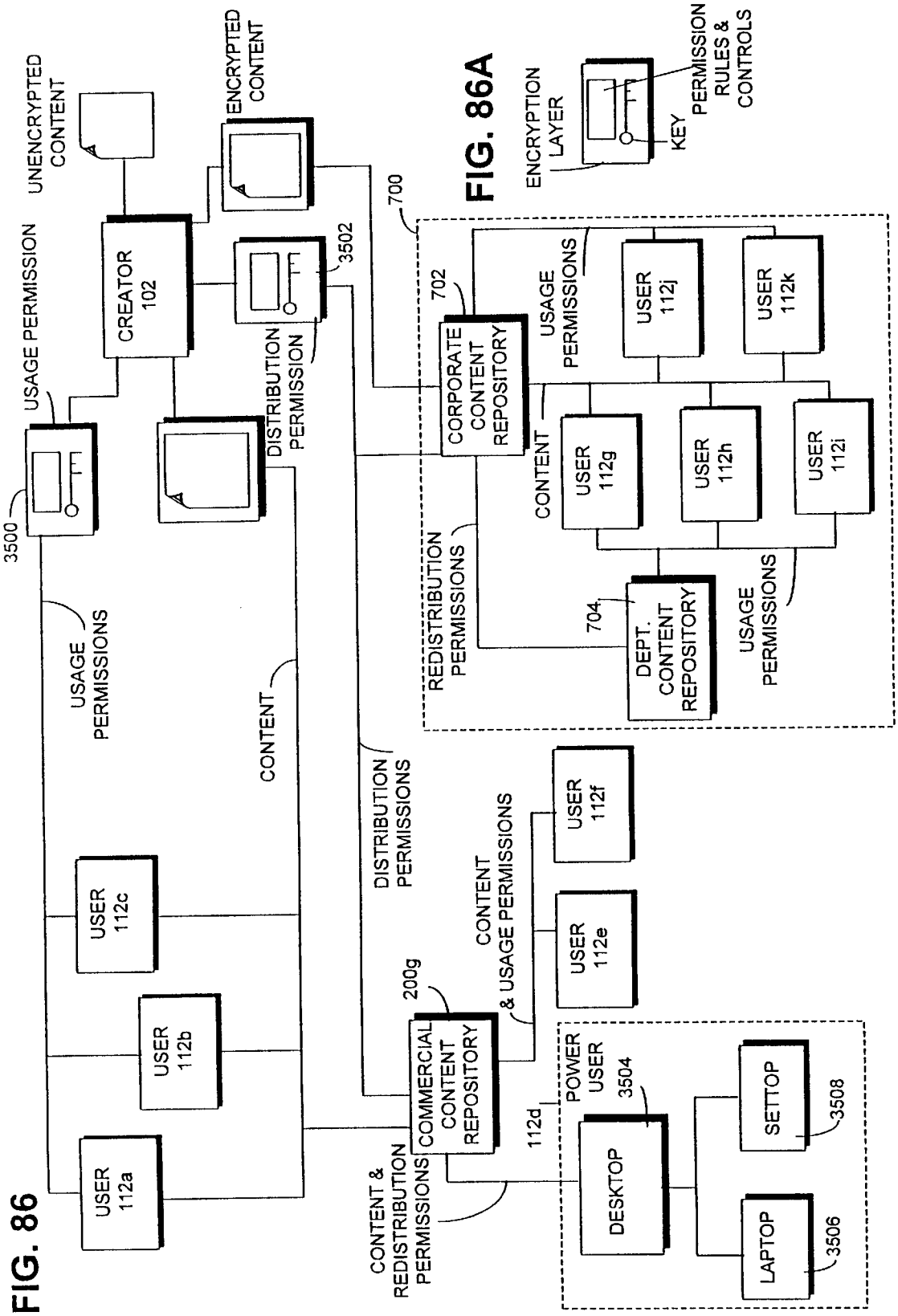


FIG. 86

FIG. 86A

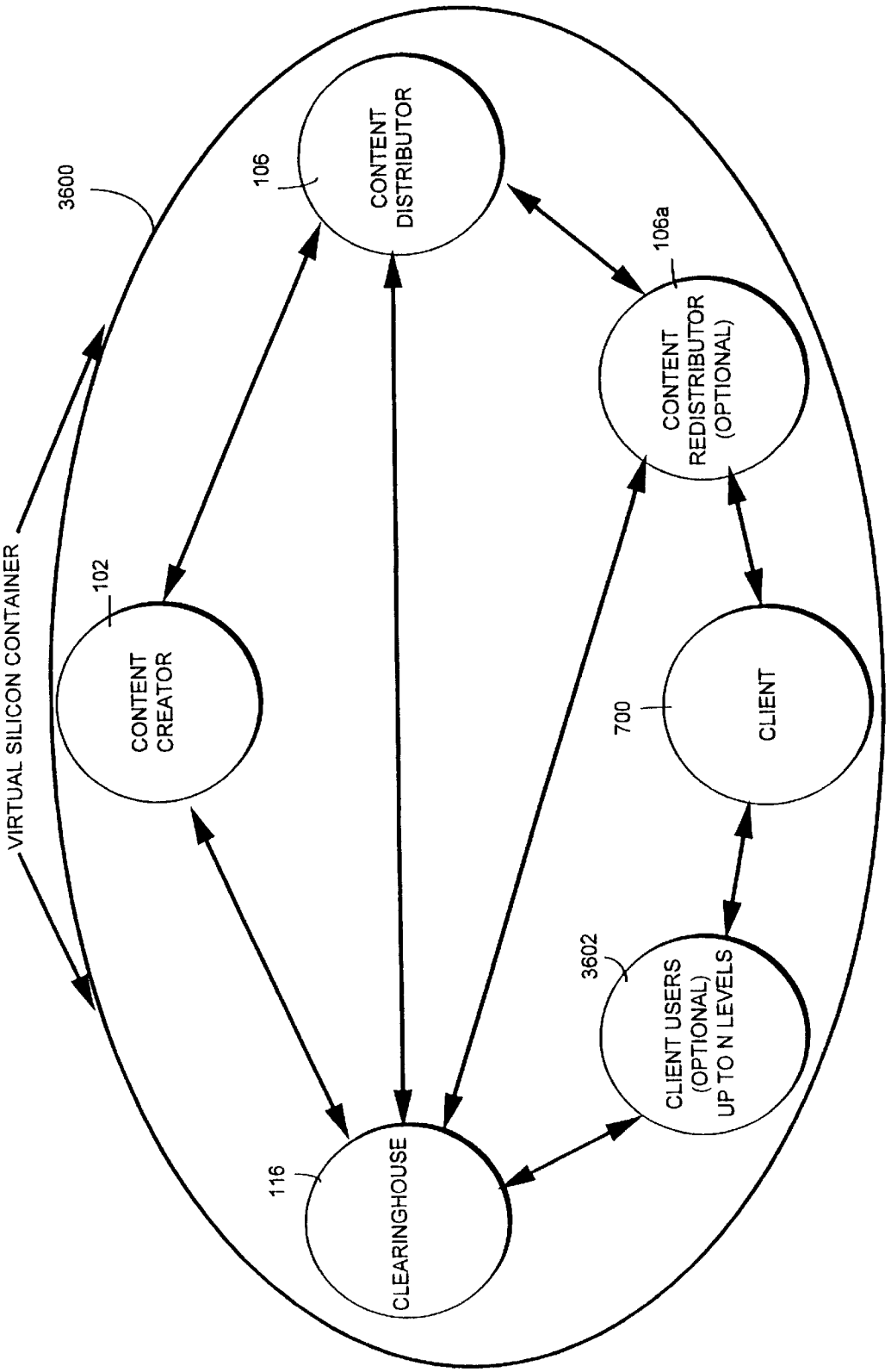

3600
VIRTUAL SILICON CONTAINER
106
CONTENT DISTRIBUTOR
106a
CONTENT REDISTRIBUTOR (OPTIONAL)
102
CONTENT CREATOR
700
CLIENT
3602
CLIENT USERS (OPTIONAL) UP TO N LEVELS
116
CLEARINGHOUSE

FIG. 87

SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION

This is a divisional of application Ser. No. 08/388,107, filed Feb. 13, 1995, abandoned.

FIELD(S) OF THE INVENTION(S)

This invention generally relates to computer and/or electronic security.

More particularly, this invention relates to systems and techniques for secure transaction management. This invention also relates to computer-based and other electronic appliance-based technologies that help to ensure that information is accessed and/or otherwise used only in authorized ways, and maintains the integrity, availability, and/or confidentiality of such information and processes related to such use.

The invention also relates to systems and methods for protecting rights of various participants in electronic commerce and other electronic or electronically-facilitated transactions.

The invention also relates to secure chains of handling and control for both information content and information employed to regulate the use of such content and consequences of such use. It also relates to systems and techniques that manage, including meter and/or limit and/or otherwise monitor use of electronically stored and/or disseminated information. The invention particularly relates to transactions, conduct and arrangements that make use of, including consequences of use of, such systems and/or techniques.

The invention also relates to distributed and other operating systems, environments and architectures. It also generally relates to secure architectures, including, for example, tamper-resistant hardware-based processors, that can be used to establish security at each node of a distributed system.

BACKGROUND AND SUMMARY OF THE INVENTION(S)

Telecommunications, financial transactions, government processes, business operations, entertainment, and personal business productivity all now depend on electronic appliances. Millions of these electronic appliances have been electronically connected together. These interconnected electronic appliances comprise what is increasingly called the "information highway." Many businesses, academicians, and government leaders are concerned about how to protect the rights of citizens and organizations who use this information (also "electronic" or "digital") highway.

Electronic Content

Today, virtually anything that can be represented by words, numbers, graphics, or system of commands and instructions can be formatted into electronic digital information. Television, cable, satellite transmissions, and on-line services transmitted over telephone lines, compete to distribute digital information and entertainment to homes and businesses. The owners and marketers of this content include software developers, motion picture and recording companies, publishers of books, magazines, and newspapers, and information database providers. The popularization of on-line services has also enabled the individual personal computer user to participate as a content provider. It is estimated that the worldwide market for electronic information in 1992 was approximately $40 billion and is

expected to grow to $200 billion by 1997, according to Microsoft Corporation. The present invention can materially enhance the revenue of content providers, lower the distribution costs and the costs for content, better support advertising and usage information gathering, and better satisfy the needs of electronic information users. These improvements can lead to a significant increase in the amount and variety of electronic information and the methods by which such information is distributed.

The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America's largest telecommunications, computer, entertainment and information provider companies to some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems.

Controlling Electronic Content

The present invention provides a new kind of "virtual distribution environment" (called "VDE" in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels "across" the "information highway." These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users, and others. VDE is the first general purpose, configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.

A fundamental problem for electronic content providers is extending their ability to control the use of proprietary information. Content providers often need to limit use to authorized activities and amounts. Participants in a business model involving, for example, provision of movies and advertising on optical discs may include actors, directors, script and other writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, and content end-users. These participants need the ability to embody their range of agreements and requirements, including use limitations, into an "extended" agreement comprising an overall electronic business model. This extended agreement is represented by electronic content control information that can automatically enforce agreed upon rights and obligations. Under VDE, such an extended agreement may comprise an electronic contract involving all business model participants. Such an agreement may alternatively, or in addition, be made up of electronic agreements between subsets of the business model participants. Through the use of VDE, electronic commerce can function in the same way as traditional commerce—that is commercial relationships regarding products and services can be shaped through the negotiation of one or more agreements between a variety of parties.

Commercial content providers are concerned with ensuring proper compensation for the use of their electronic information. Electronic digital information, for example a CD recording, can today be copied relatively easily and inexpensively. Similarly, unauthorized copying and use of software programs deprives rightful owners of billions of dollars in annual revenue according to the International Intellectual Property Alliance. Content providers and distributors have devised a number of limited function rights

protection mechanisms to protect their rights. Authorization passwords and protocols, license servers, "lock/unlock" distribution methods, and non-electronic contractual limitations imposed on users of shrink-wrapped software are a few of the more prevalent content protection schemes. In a commercial context, these efforts are inefficient and limited solutions.

Providers of "electronic currency" have also created protections for their type of content. These systems are not sufficiently adaptable, efficient, nor flexible enough to support the generalized use of electronic currency. Furthermore, they do not provide sophisticated auditing and control configuration capabilities. This means that current electronic currency tools lack the sophistication needed for many real-world financial business models. VDE provides means for anonymous currency and for "conditionally" anonymous currency, wherein currency related activities remain anonymous except under special circumstances.

VDE Control Capabilities

VDE allows the owners and distributors of electronic digital information to reliably bill for, and securely control, audit, and budget the use of, electronic information. It can reliably detect and monitor the use of commercial information products. VDE uses a wide variety of different electronic information delivery means: including, for example, digital networks, digital broadcast, and physical storage media such as optical and magnetic disks. VDE can be used by major network providers, hardware manufacturers, owners of electronic information, providers of such information, and clearinghouses that gather usage information regarding, and bill for the use of, electronic information.

VDE provides comprehensive and configurable transaction management, metering and monitoring technology. It can change how electronic information products are protected, marketed, packaged, and distributed. When used, VDE should result in higher revenues for information providers and greater user satisfaction and value. Use of VDE will normally result in lower usage costs, decreased transaction costs, more efficient access to electronic information, re-usability of rights protection and other transaction management implementations, greatly improved flexibility in the use of secured information, and greater standardization of tools and processes for electronic transaction management. VDE can be used to create an adaptable environment that fulfills the needs of electronic information owners, distributors, and users; financial clearinghouses; and usage information analyzers and resellers.

Rights and Control Information

In general, the present invention can be used to protect the rights of parties who have:

(a) proprietary or confidentiality interests in electronic information. It can, for example, help ensure that information is used only in authorized ways;

(b) financial interests resulting from the use of electronically distributed information. It can help ensure that content providers will be paid for use of distributed information; and

(c) interests in electronic credit and electronic currency storage, communication, and/or use including electronic cash, banking, and purchasing.

Protecting the rights of electronic community members involves a broad range of technologies. VDE combines these technologies in a way that creates a "distributed" electronic rights protection "environment." This environment secures and protects transactions and other processes important for rights protection. VDE, for example, provides the ability to prevent, or impede, interference with and/or observation of, important rights related transactions and processes. VDE, in its preferred embodiment, uses special purpose tamper resistant Secure Processing Units (SPUs) to help provide a high level of security for VDE processes and information storage and communication.

The rights protection problems solved by the present invention are electronic versions of basic societal issues. These issues include protecting property rights, protecting privacy rights, properly compensating people and organizations for their work and risk, protecting money and credit, and generally protecting the security of information. VDE employs a system that uses a common set of processes to manage rights issues in an efficient, trusted, and cost-effective way.

VDE can be used to protect the rights of parties who create electronic content such as, for example: records, games, movies, newspapers, electronic books and reference materials, personal electronic mail, and confidential records and communications. The invention can also be used to protect the rights of parties who provide electronic products, such as publishers and distributors; the rights of parties who provide electronic credit and currency to pay for use of products, for example, credit clearinghouses and banks; the rights to privacy of parties who use electronic content (such as consumers, business people, governments); and the privacy rights of parties described by electronic information, such as privacy rights related to information contained in a medical record, tax record, or personnel record.

In general, the present invention can protect the rights of parties who have:

(a) commercial interests in electronically distributed information—the present invention can help ensure, for example, that parties, will be paid for use of distributed information in a manner consistent with their agreement;

(b) proprietary and/or confidentiality interests in electronic information—the present invention can, for example, help ensure that data is used only in authorized ways;

(c) interests in electronic credit and electronic currency storage, communication, and/or use—this can include electronic cash, banking, and purchasing; and

(d) interests in electronic information derived, at least in part, from use of other electronic information.

VDE Functional Properties

VDE is a cost-effective and efficient rights protection solution that provides a unified, consistent system for securing and managing transaction processing. VDE can:

(a) audit and analyze the use of content,

(b) ensure that content is used only in authorized ways, and

(c) allow information regarding content usage to be used only in ways approved by content users.

In addition, VDE:

(a) is very configurable, modifiable, and re-usable;

(b) supports a wide range of useful capabilities that may be combined in different ways to accommodate most potential applications;

(c) operates on a wide variety of electronic appliances ranging from hand-held inexpensive devices to large mainframe computers;

(d) is able to ensure the various rights of a number of different parties, and a number of different rights protection schemes, simultaneously;

(e) is able to preserve the rights of parties through a series of transactions that may occur at different times and different locations;

(f) is able to flexibly accommodate different ways of securely delivering information and reporting usage; and

(g) provides for electronic analogues to "real" money and credit, including anonymous electronic cash, to pay for products and services and to support personal (including home) banking and other financial activities.

VDE economically and efficiently fulfills the rights protection needs of electronic community members. Users of VDE will not require additional rights protection systems for different information highway products and rights problems—nor will they be required to install and learn a new system for each new information highway application.

VDE provides a unified solution that allows all content creators, providers, and users to employ the same electronic rights protection solution. Under authorized circumstances, the participants can freely exchange content and associated content control sets. This means that a user of VDE may, if allowed, use the same electronic system to work with different kinds of content having different sets of content control information. The content and control information supplied by one group can be used by people who normally use content and control information supplied by a different group. VDE can allow content to be exchanged "universally" and users of an implementation of the present invention can interact electronically without fear of incompatibilities in content control, violation of rights, or the need to get, install, or learn a new content control system.

The VDE securely administers transactions that specify protection of rights. It can protect electronic rights including, for example:

(a) the property rights of authors of electronic content,

(b) the commercial rights of distributors of content,

(c) the rights of any parties who facilitated the distribution of content,

(d) the privacy rights of users of content,

(e) the privacy rights of parties portrayed by stored and/or distributed content, and

(f) any other rights regarding enforcement of electronic agreements.

VDE can enable a very broad variety of electronically enforced commercial and societal agreements. These agreements can include electronically implemented contracts, licenses, laws, regulations, and tax collection.

Contrast With Traditional Solutions

Traditional content control mechanisms often require users to purchase more electronic information than the user needs or desires. For example, infrequent users of shrink-wrapped software are required to purchase a program at the same price as frequent users, even though they may receive much less value from their less frequent use. Traditional systems do not scale cost according to the extent or character of usage and traditional systems can not attract potential customers who find that a foxed price is too high. Systems using traditional mechanisms are also not normally particularly secure. For example, shrink-wrapping does not prevent the constant illegal pirating of software once removed from either its physical or electronic package.

Traditional electronic information rights protection systems are often inflexible and inefficient and may cause a content provider to choose costly distribution channels that increase a product's price. In general these mechanisms restrict product pricing, configuration, and marketing flexibility. These compromises are the result of techniques for controlling information which cannot accommodate both different content models and content models which reflect the many, varied requirements, such as content delivery strategies, of the model participants. This can limit a provider's ability to deliver sufficient overall value to justify a given product's cost in the eyes of many potential users. VDE allows content providers and distributors to create applications and distribution networks that reflect content providers' and users' preferred business models. It offers users a uniquely cost effective and feature rich system that supports the ways providers want to distribute information and the ways users want to use such information. VDE supports content control models that ensure rights and allow content delivery strategies to be shaped for maximum commercial results.

Chain of Handling and Control

VDE can protect a collection of rights belonging to various parties having in rights in, or to, electronic information. This information may be at one location or dispersed across (and/or moving between) multiple locations. The information may pass through a "chain" of distributors and a "chain" of users. Usage information may also be reported through one or more "chains" of parties. In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.

VDE Applications and Software

VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties. These parties may include content providers, electronic hardware manufacturers, financial service providers, or electronic "infrastructure" companies such as cable or telecommunications companies. The control information implements "Rights Applications." Rights applications "run on" the "base software" of the preferred embodiment. This base software serves as a secure, flexible, general purpose foundation that can accommodate many different rights applications, that is, many different business models and their respective participant requirements.

A rights application under VDE is made up of special purpose pieces, each of which can correspond to one or more basic electronic processes needed for a rights protection environment. These processes can be combined together like building blocks to create electronic agreements that can protect the rights, and may enforce fulfillment of the obligations, of electronic information users and providers. One or more providers of electronic information can easily combine selected building blocks to create a rights application that is unique to a specific content distribution model. A group of these pieces can represent the capabilities needed to fulfill the agreement(s) between users and providers. These pieces accommodate many requirements of electronic commerce including:

the distribution of permissions to use electronic information;

the persistence of the control information and sets of control information managing these permissions;

configurable control set information that can be selected by users for use with such information;

data security and usage auditing of electronic information; and

a secure system for currency, compensation and debit management.

For electronic commerce, a rights application, under the preferred embodiment of the present invention, can provide electronic enforcement of the business agreements between all participants. Since different groups of components can be put together for different applications, the present invention can provide electronic control information for a wide variety of different products and markets. This means the present invention can provide a "unified," efficient, secure, and cost-effective system for electronic commerce and data security. This allows VDE to serve as a single standard for electronic rights protection, data security, and electronic currency and banking.

In a VDE, the separation between a rights application and its foundation permits the efficient selection of sets of control information that are appropriate for each of many different types of applications and uses. These control sets can reflect both rights of electronic community members, as well as obligations (such as providing a history of one's use of a product or paying taxes on one's electronic purchases). VDE flexibility allows its users to electronically implement and enforce common social and commercial ethics and practices. By providing a unified control system, the present invention supports a vast range of possible transaction related interests and concerns of individuals, communities, businesses, and governments. Due to its open design, VDE allows (normally under securely controlled circumstances) applications using technology independently created by users to be "added" to the system and used in conjunction with the foundation of the invention. In sum, VDE provides a system that can fairly reflect and enforce agreements among parties. It is a broad ranging and systematic solution that answers the pressing need for a secure, cost-effective, and fair electronic environment.

VDE Implementation

The preferred embodiment of the present invention includes various tools that enable system designers to directly insert VDE capabilities into their products. These tools include an Application Programmer's Interface ("API") and a Rights Permissioning and Management Language ("RPML"). The RPML provides comprehensive and detailed control over the use of the invention's features. VDE also includes certain user interface subsystems for satisfying the needs of content providers, distributors, and users.

Information distributed using VDE may take many forms. It may, for example, be "distributed" for use on an individual's own computer, that is the present invention can be used to provide security for locally stored data. Alternatively, VDE may be used with information that is dispersed by authors and/or publishers to one or more recipients. This information may take many forms including: movies, audio recordings, games, electronic catalog shopping, multimedia, training materials, E-mail and personal documents, object oriented libraries, software programming resources, and reference/record keeping information resources (such as business, medical, legal, scientific, governmental, and consumer databases).

Electronic rights protection provided by the present invention will also provide an important foundation for trusted and efficient home and commercial banking, electronic credit processes, electronic purchasing, true or conditionally anonymous electronic cash, and EDI (Electronic Data Interchange). VDE provides important enhancements for improving data security in organizations by providing "smart" transaction management features that can be far more effective than key and password based "go/no go" technology.

VDE normally employs an integration of cryptographic and other security technologies (e.g. encryption, digital signatures, etc.), with other technologies including: component, distributed, and event driven operating system technology, and related communications, object container, database, smart agent, smart card, and semiconductor design technologies.

I. Overview

A. VDE Solves Important Problems and Fills Critical Needs

The world is moving towards an integration of electronic information appliances. This interconnection of appliances provides a foundation for much greater electronic interaction and the evolution of electronic commerce. A variety of capabilities are required to implement an electronic commerce environment. VDE is the first system that provides many of these capabilities and therefore solves fundamental problems related to electronic dissemination of information.

Electronic Content

VDE allows electronic arrangements to be created involving two or more parties. These agreements can themselves comprise a collection of agreements between participants in a commercial value chain and/or a data security chain model for handling, auditing, reporting, and payment. It can provide efficient, reusable, modifiable, and consistent means for secure electronic content: distribution, usage control, usage payment, usage auditing, and usage reporting. Content may, for example, include:

- financial information such as electronic currency and credit;
- commercially distributed electronic information such as reference databases, movies, games, and advertising; and
- electronic properties produced by persons and organizations, such as documents, e-mail, and proprietary database information.

VDE enables an electronic commerce marketplace that supports differing, competitive business partnerships, agreements, and evolving overall business models.

The features of VDE allow it to function as the first trusted electronic information control environment that can conform to, and support, the bulk of conventional electronic commerce and data security requirements. In particular, VDE enables the participants in a business value chain model to create an electronic version of traditional business agreement terms and conditions and further enables these participants to shape and evolve their electronic commerce models as they believe appropriate to their business requirements.

VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment. VDE is not limited to being an application or application specific toolset that covers only a limited subset of electronic interaction activities and participants. Rather, VDE supports systems by which such applications can be created, modified, and/or reused. As a result, the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components. VDE can support a

single electronic "world" within which most forms of electronic transaction activities can be managed.

To answer the developing needs of rights owners and content providers and to provide a system that can accommodate the requirements and agreements of all parties that may be involved in electronic business models (creators, distributors, administrators, users, credit providers, etc.), VDE supplies an efficient, largely transparent, low cost and sufficiently secure system (supporting both hardware/software and software only models). VDE provides the widely varying secure control and administration capabilities required for:

1. Different types of electronic content,
2. Differing electronic content delivery schemes,
3. Differing electronic content usage schemes,
4. Different content usage platforms, and
5. Differing content marketing and model strategies.

VDE may be combined with, or integrated into, many separate computers and/or other electronic appliances. These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred embodiment comprises one or more "protected processing environments", one or more secure databases, and secure "component assemblies" and other items and processes that need to be kept secured. VDE can, for example, securely control electronic currency, payments, and/or credit management (including electronic credit and/or currency receipt, disbursement, encumbering, and/or allocation) using such a "secure subsystem."

VDE provides a secure, distributed electronic transaction management system for controlling the distribution and/or other usage of electronically provided and/or stored information. VDE controls auditing and reporting of electronic content and/or appliance usage. Users of VDE may include content creators who apply content usage, usage reporting, and/or usage payment related control information to electronic content and/or appliances for users such as end-user organizations, individuals, and content and/or appliance distributors. VDE also securely supports the payment of money owed (including money owed for content and/or appliance usage) by one or more parties to one or more other parties, in the form of electronic credit and/or currency.

Electronic appliances under control of VDE represent VDE 'nodes' that securely process and control; distributed electronic information and/or appliance usage, control information formulation, and related transactions. VDE can securely manage the integration of control information provided by two or more parties. As a result, VDE can construct an electronic agreement between VDE participants that represent a "negotiation" between, the control requirements of, two or more parties and enacts terms and conditions of a resulting agreement. VDE ensures the rights of each party to an electronic agreement regarding a wide range of electronic activities related to electronic information and/or appliance usage.

Through use of VDE's control system, traditional content providers and users can create electronic relationships that reflect traditional, non-electronic relationships. They can shape and modify commercial relationships to accommodate the evolving needs of, and agreements among, themselves. VDE does not require electronic content providers and users to modify their business practices and personal preferences to conform to a metering and control application program that supports limited, largely fixed functionality. Furthermore, VDE permits participants to develop business models not feasible with non-electronic commerce, for example, involving detailed reporting of content usage information, large numbers of distinct transactions at hitherto infeasibly low price points, "pass-along" control information that is enforced without involvement or advance knowledge of the participants, etc.

The present invention allows content providers and users to formulate their transaction environment to accommodate:

(1) desired content models, content control models, and content usage information pathways,

(2) a complete range of electronic media and distribution means,

(3) a broad range of pricing, payment, and auditing strategies,

(4) very flexible privacy and/or reporting models,

(5) practical and effective security architectures, and

(6) other administrative procedures that together with steps (1) through (5) can enable most "real world" electronic commerce and data security models, including models unique to the electronic world.

VDE's transaction management capabilities can enforce:

(1) privacy rights of users related to information regarding their usage of electronic information and/or appliances,

(2) societal policy such as laws that protect rights of content users or require the collection of taxes derived from electronic transaction revenue, and

(3) the proprietary and/or other rights of parties related to ownership of, distribution of, and/or other commercial rights related to, electronic information.

VDE can support "real" commerce in an electronic form, that is the progressive creation of commercial relationships that form, over time, a network of interrelated agreements representing a value chain business model. This is achieved in part by enabling content control information to develop through the interaction of (negotiation between) securely created and independently submitted sets of content and/or appliance control information. Different sets of content and/or appliance control information can be submitted by different parties in an electronic business value chain enabled by the present invention. These parties create control information sets through the use of their respective VDE installations. Independently, securely deliverable, component based control information allows efficient interaction among control information sets supplied by different parties.

VDE permits multiple, separate electronic arrangements to be formed between subsets of parties in a VDE supported electronic value chain model. These multiple agreements together comprise a VDE value chain "extended" agreement. VDE allows such constituent electronic agreements, and therefore overall VDE extended agreements, to evolve and reshape over time as additional VDE participants become involved in VDE content and/or appliance control information handling. VDE electronic agreements may also be extended as new control information is submitted by existing participants. With VDE, electronic commerce participants are free to structure and restructure their electronic commerce business activities and relationships. As a result, the present invention allows a competitive electronic commerce marketplace to develop since the use of VDE enables different, widely varying business models using the same or shared content.

A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information (normally in the form

of VDE objects containing one or more methods, data, or load module VDE components). This independently delivered control information can be integrated with senior and other pre-existing content control information to securely form derived control information using the negotiation mechanisms of the present invention. All requirements specified by this derived control information must be satisfied before VDE controlled content can be accessed or otherwise used. This means that, for example, all load modules and any mediating data which are listed by the derived control information as required must be available and securely perform their required function. In combination with other aspects of the present invention, securely, independently delivered control components allow electronic commerce participants to freely stipulate their business requirements and trade offs. As a result, much as with traditional, non-electronic commerce, the present invention allows electronic commerce (through a progressive stipulation of various control requirements by VDE participants) to evolve into forms of business that are the most efficient, competitive and useful.

VDE provides capabilities that rationalize the support of electronic commerce and electronic transaction management. This rationalization stems from the reusability of control structures and user interfaces for a wide variety of transaction management related activities. As a result, content usage control, data security, information auditing, and electronic financial activities, can be supported with tools that are reusable, convenient, consistent, and familiar. In addition, a rational approach—a transaction/distribution control standard—allows all participants in VDE the same foundation set of hardware control and security, authoring, administration, and management tools to support widely varying types of information, business market model, and/or personal objectives.

Employing VDE as a general purpose electronic transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model. For example, VDE allows content creators to use the same VDE foundation control arrangement for both content authoring and for licensing content from other content creators for inclusion into their products or for other use. Clearinghouses, distributors, content creators, and other VDE users can all interact, both with the applications running on their VDE installations, and with each other, in an entirely consistent manner, using and reusing (largely transparently) the same distributed tools, mechanisms, and consistent user interfaces, regardless of the type of VDE activity.

VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information. This includes, for example, commercially distributed content, electronic currency, electronic credit, business transactions (such as EDI), confidential communications, and the like. VDE can further be used to enable commercially provided electronic content to be made available to users in user defined portions, rather than constraining the user to use portions of content that were "predetermined" by a content creator and/or other provider for billing purposes.

VDE, for example, can employ:

(1) Secure metering means for budgeting and/or auditing electronic content and/or appliance usage;

(2) Secure flexible means for enabling compensation and/or billing rates for content and/or appliance usage, including electronic credit and/or currency mechanisms for payment means;

(3) Secure distributed database means for storing control and usage related information (and employing validated compartmentalization and tagging schemes);

(4) Secure electronic appliance control means;

(5) A distributed, secure, "virtual black box" comprised of nodes located at every user (including VDE content container creators, other content providers, client users, and recipients of secure VDE content usage information) site. The nodes of said virtual black box normally include a secure subsystem having at least one secure hardware element (a semiconductor element or other hardware module for securely executing VDE control processes), said secure subsystems being distributed at nodes along a pathway of information storage, distribution, payment, usage, and/or auditing. In some embodiments, the functions of said hardware element, for certain or all nodes, may be performed by software, for example, in host processing environments of electronic appliances;

(6) Encryption and decryption means;

(7) Secure communications means employing authentication, digital signaturing, and encrypted transmissions. The secure subsystems at said user nodes utilize a protocol that establishes and authenticates each node's and/or participant's identity, and establishes one or more secure host-to-host encryption keys for communications between the secure subsystems; and

(8) Secure control means that can allow each VDE installation to perform VDE content authoring (placing content into VDE containers with associated control information), content distribution, and content usage; as well as clearinghouse and other administrative and analysis activities employing content usage information.

VDE may be used to migrate most non-electronic, traditional information delivery models (including entertainment, reference materials, catalog shopping, etc.) into an adequately secure digital distribution and usage management and payment context. The distribution and financial pathways managed by a VDE arrangement may include:

content creator(s),

distributor(s),

redistributor(s),

client administrator(s),

client user(s),

financial and/or other clearinghouse(s),

and/or government agencies.

These distribution and financial pathways may also include:

advertisers,

market survey organizations, and/or

other parties interested in the user usage of information securely delivered and/or stored using VDE.

Normally, participants in a VDE arrangement will employ the same secure VDE foundation. Alternate embodiments

support VDE arrangements employing differing VDE foundations. Such alternate embodiments may employ procedures to ensure certain interoperability requirements are met.

Secure VDE hardware (also known as SPUs for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host Processing Environments (HPEs)), operate in conjunction with secure communications, systems integration software, and distributed software control information and support structures, to achieve the electronic contract/rights protection environment of the present invention. Together, these VDE components comprise a secure, virtual, distributed content and/or appliance control, auditing (and other administration), reporting, and payment environment. In some embodiments and where commercially acceptable, certain VDE participants, such as clearinghouses that normally maintain sufficiently physically secure non-VDE processing environments, may be allowed to employ HPEs rather VDE hardware elements and interoperate, for example, with VDE end-users and content providers. VDE components together comprise a configurable, consistent, secure and "trusted" architecture for distributed, asynchronous control of electronic content and/or appliance usage. VDE supports a "universe wide" environment for electronic content delivery, broad dissemination, usage reporting, and usage related payment activities.

VDE provides generalized configurability. This results, in part, from decomposition of generalized requirements for supporting electronic commerce and data security into a broad range of constituent "atomic" and higher level components (such as load modules, data elements, and methods) that may be variously aggregated together to for control methods for electronic commerce applications, commercial electronic agreements, and data security arrangements. VDE provides a secure operating environment employing VDE foundation elements along with secure independently deliverable VDE components that enable electronic commerce models and relationships to develop. VDE specifically supports the unfolding of distribution models in which content providers, over time, can expressly agree to, or allow, subsequent content providers and/or users to participate in shaping the control information for, and consequences of, use of electronic content and/or appliances. A very broad range of the functional attributes important for supporting simple to very complex electronic commerce and data security activities are supported by capabilities of the present invention. As a result, VDE supports most types of electronic information and/or appliance: usage control (including distribution), security, usage auditing, reporting, other administration, and payment arrangements.

VDE, in its preferred embodiment, employs object software technology and uses object technology to form "containers" for delivery of information that is (at least in part) encrypted or otherwise secured. These containers may contain electronic content products or other electronic information and some or all of their associated permissions (control) information. These container objects may be distributed along pathways involving content providers and/or content users. They may be securely moved among nodes of a Virtual Distribution Environment (VDE) arrangement, which nodes operate VDE foundation software and execute control methods to enact electronic information usage control and/or administration models. The containers delivered through use of the preferred embodiment of the present invention may be employed both for distributing VDE control instructions (information) and/or to encapsulate and electronically distribute content that has been at least partially secured.

Content providers who employ the present invention may include, for example, software application and game publishers, database publishers, cable, television, and radio broadcasters, electronic shopping vendors, and distributors of information in electronic document, book, periodical, e-mail and/or other forms. Corporations, government agencies, and/or individual "end-users" who act as storers of, and/or distributors of, electronic information, may also be VDE content providers (in a restricted model, a user provides content only to himself and employs VDE to secure his own confidential information against unauthorized use by other parties). Electronic information may include proprietary and/or confidential information for personal or internal organization use, as well as information, such as software applications, documents, entertainment materials, and/or reference information, which may be provided to other parties. Distribution may be by, for example, physical media delivery, broadcast and/or telecommunication means, and in the form of "static" files and/or streams of data. VDE may also be used, for example, for multi-site "real-time" interaction such as teleconferencing, interactive games, or on-line bulletin boards, where restrictions on, and/or auditing of, the use of all or portions of communicated information is enforced.

VDE provides important mechanisms for both enforcing commercial agreements and enabling the protection of privacy rights. VDE can securely deliver information from one party to another concerning the use of commercially distributed electronic content. Even if parties are separated by several "steps" in a chain (pathway) of handling for such content usage information, such information is protected by VDE through encryption and/or other secure processing. Because of that protection, the accuracy of such information is guaranteed by VDE, and the information can be trusted by all parties to whom it is delivered. Furthermore, VDE guarantees that all parties can trust that such information cannot be received by anyone other than the intended, authorized, party(ies) because it is encrypted such that only an authorized party, or her agents, can decrypt it. Such information may also be derived through a secure VDE process at a previous pathway-of-handling location to produce secure VDE reporting information that is then communicated securely to its intended recipient's VDE secure subsystem. Because VDE can deliver such information securely, parties to an electronic agreement need not trust the accuracy of commercial usage and/or other information delivered through means other than those under control of VDE.

VDE participants in a commercial value chain can be "commercially" confident (that is, sufficiently confident for commercial purposes) that the direct (constituent) and/or "extended" electronic agreements they entered into through the use of VDE can be enforced reliably. These agreements may have both "dynamic" transaction management related aspects, such as content usage control information enforced through budgeting, metering, and/or reporting of electronic information and/or appliance use, and/or they may include "static" electronic assertions, such as an end-user using the system to assert his or her agreement to pay for services, not to pass to unauthorized parties electronic information derived from usage of content or systems, and/or agreeing to observe copyright laws. Not only can electronically reported transaction related information be trusted under the present invention, but payment may be automated by the passing of payment tokens through a pathway of payment (which may or may not be the same as a pathway for reporting). Such payment can be contained within a VDE container created

automatically by a VDE installation in response to control information (located, in the preferred embodiment, in one or more permissions records) stipulating the "withdrawal" of credit or electronic currency (such as tokens) from an electronic account (for example, an account securely maintained by a user's VDE installation secure subsystem) based upon usage of VDE controlled electronic content and/or appliances (such as governments, financial credit providers, and users).

VDE allows the needs of electronic commerce participants to be served and it can bind such participants together in a universe wide, trusted commercial network that can be secure enough to support very large amounts of commerce. VDE's security and metering secure subsystem core will be present at all physical locations where VDE related content is (a) assigned usage related control information (rules and mediating data), and/or (b) used. This core can perform security and auditing functions (including metering) that operate within a "virtual black box," a collection of distributed, very secure VDE related hardware instances that are interconnected by secured information exchange (for example, telecommunication) processes and distributed database means. VDE further includes highly configurable transaction operating system technology, one or more associated libraries of load modules along with affiliated data, VDE related administration, data preparation, and analysis applications, as well as system software designed to enable VDE integration into host environments and applications. VDE's usage control information, for example, provide for property content and/or appliance related: usage authorization, usage auditing (which may include audit reduction), usage billing, usage payment, privacy filtering, reporting, and security related communication and encryption techniques.

VDE extensively employs methods in the form of software objects to augment configurability, portability, and security of the VDE environment. It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g, summary, table of contents) and secured content control information which ensures the performance of control information. Content control information governs content usage according to criteria set by holders of rights to an object's contents and/or according to parties who otherwise have rights associated with distributing such content (such as governments, financial credit providers, and users).

In part, security is enhanced by object methods employed by the present invention because the encryption schemes used to protect an object can efficiently be further used to protect the associated content control information (software control information and relevant data) from modification. Said object techniques also enhance portability between various computer and/or other appliance environments because electronic information in the form of content can be inserted along with (for example, in the same object container as) content control information (for said content) to produce a "published" object. As a result, various portions of said control information may be specifically adapted for different environments, such as for diverse computer platforms and operating systems, and said various portions may all be carried by a VDE container.

An objective of VDE is supporting a transaction/distribution control standard. Development of such a standard has many obstacles, given the security requirements and related hardware and communications issues, widely differing environments, information types, types of information usage, business and/or data security goals, varieties of participants, and properties of delivered information. A significant feature of VDE accommodates the many, varying distribution and other transaction variables by, in part, decomposing electronic commerce and data security functions into generalized capability modules executable within a secure hardware SPU and/or corresponding software subsystem and further allowing extensive flexibility in assembling, modifying, and/or replacing, such modules (e.g. load modules and/or methods) in applications run on a VDE installation foundation. This configurability and reconfigurability allows electronic commerce and data security participants to reflect their priorities and requirements through a process of iteratively shaping an evolving extended electronic agreement (electronic control model). This shaping can occur as content control information passes from one VDE participant to another and to the extent allowed by "in place" content control information. This process allows users of VDE to recast existing control information and/or add new control information as necessary (including the elimination of no longer required elements).

VDE supports trusted (sufficiently secure) electronic information distribution and usage control models for both commercial electronic content distribution and data security applications. It can be configured to meet the diverse requirements of a network of interrelated participants that may include content creators, content distributors, client administrators, end users, and/or clearinghouses and/or other content usage information users. These parties may constitute a network of participants involved in simple to complex electronic content dissemination, usage control, usage reporting, and/or usage payment. Disseminated content may include both originally provided and VDE generated information (such as content usage information) and content control information may persist through both chains (one or more pathways) of content and content control information handling, as well as the direct usage of content. The configurability provided by the present invention is particularly critical for supporting electronic commerce, that is enabling businesses to create relationships and evolve strategies that offer competitive value. Electronic commerce tools that are not inherently configurable and interoperable will ultimately fail to produce products (and services) that meet both basic requirements and evolving needs of most commerce applications.

VDE's fundamental configurability will allow a broad range of competitive electronic commerce business models to flourish. It allows business models to be shaped to maximize revenues sources, end-user product value, and operating efficiencies. VDE can be employed to support multiple, differing models, take advantage of new revenue opportunities, and deliver product configurations most desired by users. Electronic commerce technologies that do not, as the present invention does:

- support a broad range of possible, complementary revenue activities,
- offer a flexible array of content usage features most desired by customers, and
- exploit opportunities for operating efficiencies,

will result in products that are often intrinsically more costly and less appealing and therefore less competitive in the marketplace.

Some of the key factors contributing to the configurability intrinsic to the present invention include:

(a) integration into the fundamental control environment of a broad range of electronic appliances through portable API and programming language tools that efficiently support merging of control and auditing

capabilities in nearly any electronic appliance environment while maintaining overall system security;

(b) modular data structures;

(c) generic content model;

(d) general modularity and independence of foundation architectural components;

(e) modular security structures;

(f) variable length and multiple branching chains of control; and

(g) independent, modular control structures in the form of executable load modules that can be maintained in one or more libraries, and assembled into control methods and models, and where such model control schemes can "evolve" as control information passes through the VDE installations of participants of a pathway of VDE content control information handling.

Because of the breadth of issues resolved by the present invention, it can provide the emerging "electronic highway" with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions. VDE's electronic transaction management mechanisms can enforce the electronic rights and agreements of all parties participating in widely varying business and data security models, and this can be efficiently achieved through a single VDE implementation within each VDE participant's electronic appliance. VDE supports widely varying business and/or data security models that can involve a broad range of participants at various "levels" of VDE content and/or content control information pathways of handling. Different content control and/or auditing models and agreements may be available on the same VDE installation. These models and agreements may control content in relationship to, for example, VDE installations and/or users in general; certain specific users, installations, classes and/or other groupings of installations and/or users; as well as to electronic content generally on a given installation, to specific properties, property portions, classes and/or other groupings of content.

Distribution using VDE may package both the electronic content and control information into the same VDE container, and/or may involve the delivery to an end-user site of different pieces of the same VDE managed property from plural separate remote locations and/or in plural separate VDE content containers and/or employing plural different delivery means. Content control information may be partially or fully delivered separately from its associated content to a user VDE installation in one or more VDE administrative objects. Portions of said control information may be delivered from one or more sources. Control information may also be available for use by access from a user's VDE installation secure sub-system to one or more remote VDE secure sub-systems and/or VDE compatible, certified secure remote locations. VDE control processes such as metering, budgeting, decrypting and/or fingerprinting, may as relates to a certain user content usage activity, be performed in a user's local VDE installation secure subsystem, or said processes may be divided amongst plural secure subsystems which may be located in the same user VDE installations and/or in a network server and in the user installation. For example, a local VDE installation may perform decryption and save any, or all of, usage metering information related to content and/or electronic appliance usage at such user installation could be performed at the server employing secure (e.g., encrypted) communications between said secure subsystems. Said server location may also be used for near real time, frequent, or more periodic secure receipt of content usage information from said user installation, with, for example, metered information being maintained only temporarily at a local user installation.

Delivery means for VDE managed content may include electronic data storage means such as optical disks for delivering one portion of said information and broadcasting and/or telecommunicating means for other portions of said information. Electronic data storage means may include magnetic media, optical media, combined magneto-optical systems, flash RAM memory, bubble memory, and/or other memory storage means such as huge capacity optical storage systems employing holographic, frequency, and/or polarity data storage techniques. Data storage means may also employ layered disc techniques, such as the use of generally transparent and/or translucent materials that pass light through layers of data carrying discs which themselves are physically packaged together as one thicker disc. Data carrying locations on such discs may be, at least in part, opaque.

VDE supports a general purpose foundation for secure transaction management, including usage control, auditing, reporting, and/or payment. This general purpose foundation is called "VDE Functions" ("VDEFs"). VDE also supports a collection of "atomic" application elements (e.g., load modules) that can be selectively aggregated together to form various VDEF capabilities called control methods and which serve as VDEF applications and operating system functions. When a host operating environment of an electronic appliance includes VDEF capabilities, it is called a "Rights Operating System" (ROS). VDEF load modules, associated data, and methods form a body of information that for the purposes of the present invention are called "control information." VDEF control information may be specifically associated with one or more pieces of electronic content and/or it may be employed as a general component of the operating system capabilities of a VDE installation.

VDEF transaction control elements reflect and enact content specific and/or more generalized administrative (for example, general operating system) control information. VDEF capabilities which can generally take the form of applications (application models) that have more or less configurability which can be shaped by VDE participants, through the use, for example, of VDE templates, to employ specific capabilities, along, for example, with capability parameter data to reflect the elements of one or more express electronic agreements between VDE participants in regards to the use of electronic content such as commercially distributed products. These control capabilities manage the use of, and/or auditing of use of, electronic content, as well as reporting information based upon content use, and any payment for said use. VDEF capabilities may "evolve" to reflect the requirements of one or more successive parties who receive or otherwise contribute to a given set of control information. Frequently, for a VDE application for a given content model (such as distribution of entertainment on CD-ROM, content delivery from an Internet repository, or electronic catalog shopping and advertising, or some combination of the above) participants would be able to securely select from amongst available, alternative control methods and apply related parameter data, wherein such selection of control method and/or submission of data would constitute their "contribution" of control information. Alternatively, or in addition, certain control methods that have been expressly certified as securely interoperable and compatible with said application may be independently submitted by a participant

as part of such a contribution. In the most general example, a generally certified load module (certified for a given VDE arrangement and/or content class) may be used with many or any VDE application that operates in nodes of said arrangement. These parties, to the extent they are allowed, can independently and securely add, delete, and/or otherwise modify the specification of load modules and methods, as well as add, delete or otherwise modify related information.

Normally the party who creates a VDE content container defines the general nature of the VDEF capabilities that will and/or may apply to certain electronic information. A VDE content container is an object that contains both content (for example, commercially distributed electronic information products such as computer software programs, movies, electronic publications or reference materials, etc.) and certain control information related to the use of the object's content. A creating party may make a VDE container available to other parties. Control information delivered by, and/or otherwise available for use with, VDE content containers comprise (for commercial content distribution purposes) VDEF control capabilities (and any associated parameter data) for electronic content. These capabilities may constitute one or more "proposed" electronic agreements (and/or agreement functions available for selection and/or use with parameter data) that manage the use and/or the consequences of use of such content and which can enact the terms and conditions of agreements involving multiple parties and their various rights and obligations.

A VDE electronic agreement may be explicit, through a user interface acceptance by one or more parties, for example by a "junior" party who has received control information from a "senior" party, or it may be a process amongst equal parties who individually assert their agreement. Agreement may also result from an automated electronic process during which terms and conditions are "evaluated" by certain VDE participant control information that assesses whether certain other electronic terms and conditions attached to content and/or submitted by another party are acceptable (do not violate acceptable control information criteria). Such an evaluation process may be quite simple, for example a comparison to ensure compatibility between a portion of, or all senior, control terms and conditions in a table of terms and conditions and the submitted control information of a subsequent participant in a pathway of content control information handling, or it may be a more elaborate process that evaluates the potential outcome of, and/or implements a negotiation process between, two or more sets of control information submitted by two or more parties. VDE also accommodates a semi-automated process during which one or more VDE participants directly, through user interface means, resolve "disagreements" between control information sets by accepting and/or proposing certain control information that may be acceptable to control information representing one or more other parties interests and/or responds to certain user interface queries for selection of certain alternative choices and/or for certain parameter information, the responses being adopted if acceptable to applicable senior control information.

When another party (other than the first applier of rules), perhaps through a negotiation process, accepts, and/or adds to and/or otherwise modifies, "in place" content control information, a VDE agreement between two or more parties related to the use of such electronic content may be created (so long as any modifications are consistent with senior control information). Acceptance of terms and conditions related to certain electronic content may be direct and express, or it may be implicit as a result of use of content (depending, for example, on legal requirements, previous exposure to such terms and conditions, and requirements of in place control information).

VDEF capabilities may be employed, and a VDE agreement may be entered into, by a plurality of parties without the VDEF capabilities being directly associated with the controlling of certain, specific electronic information. For example, certain one or more VDEF capabilities may be present at a VDE installation, and certain VDE agreements may have been entered into during the registration process for a content distribution application, to be used by such installation for securely controlling VDE content usage, auditing, reporting and/or payment. Similarly, a specific VDE participant may enter into a VDE user agreement with a VDE content or electronic appliance provider when the user and/or her appliance register with such provider as a VDE installation and/or user. In such events, VDEF in place control information available to the user VDE installation may require that certain VDEF methods are employed, for example in a certain sequence, in order to be able to use all and/or certain classes, of electronic content and/or VDE applications.

VDE ensures that certain prerequisites necessary for a given transaction to occur are met. This includes the secure execution of any required load modules and the availability of any required, associated data. For example, required load modules and data (e.g. in the for of a method) might specify that sufficient credit from an authorized source must be confirmed as available. It might further require certain one or more load modules execute as processes at an appropriate time to ensure that such credit will be used in order to pay for user use of the content. A certain content provider might, for example, require metering the number of copies made for distribution to employees of a given software program (a portion of the program might be maintained in encrypted form and require the presence of a VDE installation to run). This would require the execution of a metering method for copying of the property each time a copy was made for another employee. This same provider might also charge fees based on the total number of different properties licensed from them by the user and a metering history of their licensing of properties might be required to maintain this information.

VDE provides organization, community, and/or universe wide secure environments whose integrity is assured by processes securely controlled in VDE participant user installations (nodes). VDE installations, in the preferred embodiment, may include both software and tamper resistant hardware semiconductor elements. Such a semiconductor arrangement comprises, at least in part, special purpose circuitry that has been designed to protect against tampering with, or unauthorized observation of, the information and functions used in performing the VDE's control functions. The special purpose secure circuitry provided by the present invention includes at least one of: a dedicated semiconductor arrangement known as a Secure Processing Unit (SPU) and/or a standard microprocessor, microcontroller, and/or other processing logic that accommodates the requirements of the present invention and functions as an SPU. VDE's secure hardware may be found incorporated into, for example, a fax/modem chip or chip pack, I/O controller, video display controller, and/or other available digital processing arrangements. It is anticipated that portions of the present invention's VDE secure hardware capabilities may ultimately be standard design elements of central processing units (CPUs) for computers and various other electronic devices.

Designing VDE capabilities into one or more standard microprocessor, microcontroller and/or other digital processing components may materially reduce VDE related hardware costs by employing the same hardware resources for both the transaction management uses contemplated by the present invention and for other, host electronic appliance functions. This means that a VDE SPU can employ (share) circuitry elements of a "standard" CPU. For example, if a "standard" processor can operate in protected mode and can execute VDE related instructions as a protected activity, then such an embodiment may provide sufficient hardware security for a variety of applications and the expense of a special purpose processor might be avoided. Under one preferred embodiment of the present invention, certain memory (e.g., RAM, ROM, NVRAM) is maintained during VDE related instruction processing in a protected mode (for example, as supported by protected mode microprocessors). This memory is located in the same package as the processing logic (e.g. processor). Desirably, the packaging and memory of such a processor would be designed using security techniques that enhance its resistance to tampering.

The degree of overall security of the VDE system is primarily dependent on the degree of tamper resistance and concealment of VDE control process execution and related data storage activities. Employing special purpose semiconductor packaging techniques can significantly contribute to the degree of security. Concealment and tamper-resistance in semiconductor memory (e.g., RAM, ROM, NVRAM) can be achieved, in part, by employing such memory within an SPU package, by encrypting data before it is sent to external memory (such as an external RAM package) and decrypting encrypted data within the CPU/RAM package before it is executed. This process is used for important VDE related data when such data is stored on unprotected media, for example, standard host storage, such as random access memory, mass storage, etc. In that event, a VDE SPU would encrypt data that results from a secure VDE execution before such data was stored in external memory.

Summary of Some Important Features Provided by VDE in Accordance With the Present Invention

VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, VDE includes features that:

"sufficiently" impede unauthorized and/or uncompensated use of electronic information and/or appliances through the use of secure communication, storage, and transaction management technologies. VDE supports a model wide, distributed security implementation which creates a single secure "virtual" transaction processing and information storage environment. VDE enables distributed VDE installations to securely store and communicate information and remotely control the execution processes and the character of use of electronic information at other VDE installations and in a wide variety of ways;

support low-cost, efficient, and effective security architectures for transaction control, auditing, reporting, and related communications and information storage. VDE may employ tagging related security techniques, the time-ageing of encryption keys, the compartmentalization of both stored control information (including differentially tagging such stored information to ensure against substitution and tampering) and distributed content (to, for many content applications, employ one or more content encryption keys that are unique to the specific VDE installation and/or user), private key techniques such as triple DES to encrypt content, public key techniques such as RSA to protect communications and to provide the benefits of digital signature and authentication to securely bind together the nodes of a VDE arrangement, secure processing of important transaction management executable code, and a combining of a small amount of highly secure, hardware protected storage space with a much larger "exposed" mass media storage space storing secured (normally encrypted and tagged) control and audit information. VDE employs special purpose hardware distributed throughout some or all locations of a VDE implementation: a) said hardware controlling important elements of: content preparation (such as causing such content to be placed in a VDE content container and associating content control information with said content), content and/or electronic appliance usage auditing, content usage analysis, as well as content usage control; and b) said hardware having been designed to securely handle processing load module control activities, wherein said control processing activities may involve a sequence of required control factors;

support dynamic user selection of information subsets of a VDE electronic information product (VDE controlled content). This contrasts with the constraints of having to use a few high level individual, pre-defined content provider information increments such as being required to select a whole information product or product section in order to acquire or otherwise use a portion of such product or section. VDE supports metering and usage control over a variety of increments (including "atomic" increments, and combinations of different increment types) that are selected ad hoc by a user and represent a collection of pre-identified one or more increments (such as one or more blocks of a preidentified nature, e.g., bytes, images, logically related blocks) that form a generally arbitrary, but logical to a user, content "deliverable." VDE control information (including budgeting, pricing and metering) can be configured so that it can specifically apply, as appropriate, to ad hoc selection of different, unanticipated variable user selected aggregations of information increments and pricing levels can be, at least in part, based on quantities and/or nature of mixed increment selections (for example, a certain quantity of certain text could mean associated images might be discounted by 15%; a greater quantity of text in the "mixed" increment selection might mean the images are discounted 20%). Such user selected aggregated information increments can reflect the actual requirements of a user for information and is more flexible than being limited to a single, or a few, high level, (e.g. product, document, database record) predetermined increments. Such high level increments may include quantities of information not desired by the user and as a result be more costly than the subset of information needed by the user if such a subset was available. In sum, the present invention allows information contained in electronic information products to be supplied according to user specification. Tailoring to user specification allows the present invention to provide the greatest value to users, which in turn will generate the greatest amount of electronic commerce activity. The user, for example, would be able to define an aggrega-

tion of content derived from various portions of an available content product, but which, as a deliverable for use by the user, is an entirely unique aggregated increment. The user may, for example, select certain numbers of bytes of information from various portions of an information product, such as a reference work, and copy them to disc in unencrypted form and be billed based on total number of bytes plus a surcharge on the number of "articles" that provided the bytes. A content provider might reasonably charge less for such a user defined information increment since the user does not require all of the content from all of the articles that contained desired information. This process of defining a user desired information increment may involve artificial intelligence database search tools that contribute to the location of the most relevant portions of information from an information product and cause the automatic display to the user of information describing search criteria hits for user selection or the automatic extraction and delivery of such portions to the user. VDE further supports a wide variety of predefined increment types including:

bytes,

images,

content over time for audio or video, or any other increment that can be identified by content provider data mapping efforts, such as:

sentences,

paragraphs,

articles,

database records, and

byte offsets representing increments of logically related information.

VDE supports as many simultaneous predefined increment types as may be practical for a given type of content and business model.

securely store at a user's site potentially highly detailed information reflective of a user's usage of a variety of different content segment types and employing both inexpensive "exposed" host mass storage for maintaining detailed information in the form of encrypted data and maintaining summary information for security testing in highly secure special purpose VDE installation nonvolatile memory (if available).

support trusted chain of handling capabilities for pathways of distributed electronic information and/or for content usage related information. Such chains may extend, for example, from a content creator, to a distributor, a redistributor, a client user, and then may provide a pathway for securely reporting the same and/or differing usage information to one or more auditors, such as to one or more independent clearinghouses and then back to the content providers, including content creators. The same and/or different pathways employed for certain content handling, and related content control information and reporting information handling, may also be employed as one or more pathways for electronic payment handling (payment is characterized in the present invention as administrative content) for electronic content and/or appliance usage. These pathways are used for conveyance of all or portions of content, and/or content related control information. Content creators and other providers can specify the pathways that, partially or fully, must be used to disseminate commercially distributed property content, content control information, payment administrative content, and/or associated usage reporting information. Control information specified by content providers may also specify which specific parties must or may (including, for example, a group of eligible parties from which a selection may be made) handle conveyed information. It may also specify what transmission means (for example telecommunication carriers or media types) and transmission hubs must or may be used.

support flexible auditing mechanisms, such as employing "bitmap meters," that achieve a high degree of efficiency of operation and throughput and allow, in a practical manner, the retention and ready recall of information related to previous usage activities and related patterns. This flexibility is adaptable to a wide variety of billing and security control strategies such as:

upgrade pricing (e.g. suite purchases),

pricing discounts (including quantity discounts),

billing related time duration variables such as discounting new purchases based on the timing of past purchases, and

security budgets based on quantity of different, logically related units of electronic information used over an interval of time.

Use of bitmap meters (including "regular" and "wide" bitmap meters) to record usage and/or purchase of information, in conjunction with other elements of the preferred embodiment of the present invention, uniquely supports efficient maintenance of usage history for: (a) rental, (b) flat fee licensing or purchase, (c) licensing or purchase discounts based upon historical usage variables, and (d) reporting to users in a manner enabling users to determine whether a certain item was acquired, or acquired within a certain time period (without requiring the use of conventional database mechanisms, which are highly inefficient for these applications). Bitmap meter methods record activities associated with electronic appliances, properties, objects, or portions thereof, and/or administrative activities that are independent of specific properties, objects, etc., performed by a user and/or electronic appliance such that a content and/or appliance provider and/or controller of an administrative activity can determine whether a certain activity has occurred at some point, or during a certain period, in the past (for example, certain use of a commercial electronic content product and/or appliance). Such determinations can then be used as part of pricing and/or control strategies of a content and/or appliance provider, and/or controller of an administrative activity. For example, the content provider may choose to charge only once for access to a portion of a property, regardless of the number of times that portion of the property is accessed by a user.

support "launchable" content, that is content that can be provided by a content provider to an end-user, who can then copy or pass along the content to other end-user parties without requiring the direct participation of a content provider to register and/or otherwise initialize the content for use. This content goes "out of (the traditional distribution) channel" in the form of a "traveling object." Traveling objects are containers that securely carry at least some permissions information and/or methods that are required for their use (such methods need not be carried by traveling objects if the required methods will be available at, or directly available to, a destination VDE installation). Certain travelling objects may be used at some or all VDE instal-

lations of a given VDE arrangement since they can make available the content control information necessary for content use without requiring the involvement of a commercial VDE value chain participant or data security administrator (e.g. a control officer or network administrator). As long as traveling object control information requirements are available at the user VDE installation secure subsystem (such as the presence of a sufficient quantity of financial credit from an authorized credit provider), at least some travelling object content may be used by a receiving party without the need to establish a connection with a remote VDE authority (until, for example, budgets are exhausted or a time content usage reporting interval has occurred). Traveling objects can travel "out-of-channel," allowing, for example, a user to give a copy of a traveling object whose content is a software program, a movie or a game, to a neighbor, the neighbor being able to use the traveling object if appropriate credit (e.g. an electronic clearinghouse account from a clearinghouse such as VISA or AT&T) is available. Similarly, electronic information that is generally available on an Internet, or a similar network, repository might be provided in the form of a traveling object that can be downloaded and subsequently copied by the initial downloader and then passed along to other parties who may pass the object on to additional parties.

provide very flexible and extensible user identification according to individuals, installations, by groups such as classes, and by function and hierarchical identification employing a hierarchy of levels of client identification (for example, client organization ID, client department ID, client network ID, client project ID, and client employee ID, or any appropriate subset of the above).

provide a general purpose, secure, component based content control and distribution system that functions as a foundation transaction operating system environment that employs executable code pieces crafted for transaction control and auditing. These code pieces can be reused to optimize efficiency in creation and operation of trusted, distributed transaction management arrangements. VDE supports providing such executable code in the form of "atomic" load modules and associated data. Many such load modules are inherently configurable, aggregatable, portable, and extensible and singularly, or in combination (along with associated data), run as control methods under the VDE transaction operating environment. VDE can satisfy the requirements of widely differing electronic commerce and data security applications by, in part, employing this general purpose transaction management foundation to securely process VDE transaction related control methods. Control methods are created primarily through the use of one or more of said executable, reusable load module code pieces (normally in the form of executable object components) and associated data. The component nature of control methods allows the present invention to efficiently operate as a highly configurable content control system. Under the present invention, content control models can be iteratively and asynchronously shaped, and otherwise updated to accommodate the needs of VDE participants to the extent that such shaping and otherwise updating conforms to constraints applied by a VDE application, if any (e.g., whether new component assemblies are accepted and, if so, what certification requirements exist for such component assemblies or whether any or certain participants may shape any or certain control information by selection amongst optional control information (permissions record) control methods. This iterative (or concurrent) multiple participant process occurs as a result of the submission and use of secure, control information components (executable code such as load modules and/or methods, and/or associated data). These components may be contributed independently by secure communication between each control information influencing VDE participant's VDE installation and may require certification for use with a given application, where such certification was provided by a certification service manager for the VDE arrangement who ensures secure interoperability and/or reliability (e.g., bug control resulting from interaction) between appliances and submitted control methods. The transaction management control functions of a VDE electronic appliance transaction operating environment interact with non-secure transaction management operating system functions to properly direct transaction processes and data related to electronic information security, usage control, auditing, and usage reporting. VDE provides the capability to manages resources related to secure VDE content and/or appliance control information execution and data storage.

facilitate creation of application and/or system functionality under VDE and to facilitate integration into electronic appliance environments of load modules and methods created under the present invention. To achieve this, VDE employs an Application Programmer's Interface (API) and/or a transaction operating system (such as a ROS) programming language with incorporated functions, both of which support the use of capabilities and can be used to efficiently and tightly integrate VDE functionality into commercial and user applications.

support user interaction through: (a) "Pop-Up" applications which, for example, provide messages to users and enable users to take specific actions such as approving a transaction, (b) stand-alone VDE applications that provide administrative environments for user activities such as: end-user preference specifications for limiting the price per transaction, unit of time, and/or session, for accessing history information concerning previous transactions, for reviewing financial information such as budgets, expenditures (e.g. detailed and/or summary) and usage analysis information, and (c) VDE aware applications which, as a result of the use of a VDE API and/or a transaction management (for example, ROS based) programming language embeds VDE "awareness" into commercial or internal software (application programs, games, etc.) so that VDE user control information and services are seamlessly integrated into such software and can be directly accessed by a user since the underlying functionality has been integrated into the commercial software's native design. For example, in a VDE aware word processor application, a user may be able to "print" a document into a VDE content container object, applying specific control information by selecting from amongst a series of different menu templates for different purposes (for example, a confidential memo template for internal organization purposes may restrict the ability to "keep," that is to make an electronic copy of the memo).

employ "templates" to ease the process of configuring capabilities of the present invention as they relate to

specific industries or businesses. Templates are applications or application add-ons under the present invention. Templates support the efficient specification and/or manipulation of criteria related to specific content types, distribution approaches, pricing mechanisms, user interactions with content and/or administrative activities, and/or the like. Given the very large range of capabilities and configurations supported by the present invention, reducing the range of configuration opportunities to a manageable subset particularly appropriate for a given business model allows the full configurable power of the present invention to be easily employed by "typical" users who would be otherwise burdened with complex programming and/or configuration design responsibilities template applications can also help ensure that VDE related processes are secure and optimally bug free by reducing the risks associated with the contribution of independently developed load modules, including unpredictable aspects of code interaction between independent modules and applications, as well as security risks associated with possible presence of viruses in such modules. VDE, through the use of templates, reduces typical user configuration responsibilities to an appropriately focused set of activities including selection of method types (e.g. functionality) through menu choices such as multiple choice, icon selection, and/or prompting for method parameter data (such as identification information, prices, budget limits, dates, periods of time, access rights to specific content, etc.) that supply appropriate and/or necessary data for control information purposes. By limiting the typical (non-programming) user to a limited subset of configuration activities whose general configuration environment (template) has been preset to reflect general requirements corresponding to that user, or a content or other business model can very substantially limit difficulties associated with content containerization (including placing initial control information on content), distribution, client administration, electronic agreement implementation, end-user interaction, and clearinghouse activities, including associated interoperability problems (such as conflicts resulting from security, operating system, and/or certification incompatibilities). Use of appropriate VDE templates can assure users that their activities related to content VDE containerization, contribution of other control information, communications, encryption techniques and/or keys, etc. will be in compliance with specifications for their distributed VDE arrangement. VDE templates constitute preset configurations that can normally be reconfigurable to allow for new and/or modified templates that reflect adaptation into new industries as they evolve or to reflect the evolution or other change of an existing industry. For example, the template concept may be used to provide individual, overall frameworks for organizations and individuals that create, modify, market, distribute, consume, and/or otherwise use movies, audio recordings and live performances, magazines, telephony based retail sales, catalogs, computer software, information data bases, multimedia, commercial communications, advertisements, market surveys, infomercials, games, CAD/CAM services for numerically controlled machines, and the like. As the context surrounding these templates changes or evolves, template applications provided under the present invention may be modified to meet these changes for broad use, or for more focused activities. A given VDE participant may have a plurality of templates available for different tasks. A party that places content in its initial VDE container may have a variety of different, configurable templates depending on the type of content and/or business model related to the content. An end-user may have different configurable templates that can be applied to different document types (e-mail, secure internal documents, database records, etc.) and/or subsets of users (applying differing general sets of control information to different bodies of users, for example, selecting a list of users who may, under certain preset criteria, use a certain document). Of course, templates may, under certain circumstances have fixed control information and not provide for user selections or parameter data entry.

support plural, different control models regulating the use and/or auditing of either the same specific copy of electronic information content and/or differently regulating different copies (occurrences) of the same electronic information content. Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments. This includes supporting variable control information for budgeting and auditing usage as applied to a variety of predefined increments of electronic information, including employing a variety of different budgets and/or metering increments for a given electronic information deliverable for: billing units of measure, credit limit, security budget limit and security content metering increments, and/or market surveying and customer profiling content metering increments. For example, a CD-ROM disk with a database of scientific articles might be in part billed according to a formula based on the number of bytes decrypted, number of articles containing said bytes decrypted, while a security budget might limit the use of said database to no more than 5% of the database per month for users on the wide area network it is installed on.

provide mechanisms to persistently maintain trusted content usage and reporting control information through both a sufficiently secure chain of handling of content and content control information and through various forms of usage of such content wherein said persistence of control may survive such use. Persistence of control includes the ability to extract information from a VDE container object by creating a new container whose contents are at least in part secured and that contains both the extracted content and at least a portion of the control information which control information of the original container and/or are at least in part produced by control information of the original container for this purpose and/or VDE installation control information stipulates should persist and/or control usage of content in the newly formed container. Such control information can continue to manage usage of container content if the container is "embedded" into another VDE managed object, such as an object which contains plural embedded VDE containers, each of which contains content derived (extracted) from a different source.

enables users, other value chain participants (such as clearinghouses and government agencies), and/or user organizations, to specify preferences or requirements related to their use of electronic content and/or appli-

ances. Content users, such as end-user customers using commercially distributed content (games, information resources, software programs, etc.), can define, if allowed by senior control information, budgets, and/or other control information, to manage their own internal use of content. Uses include, for example, a user setting a limit on the price for electronic documents that the user is willing to pay without prior express user authorization, and the user establishing the character of metering information he or she is willing to allow to be collected (privacy protection). This includes providing the means for content users to protect the privacy of information derived from their use of a VDE installation and content and/or appliance usage auditing. In particular, VDE can prevent information related to a participant's usage of electronic content from being provided to other parties without the participant's tacit or explicit agreement.

provide mechanisms that allow control information to "evolve" and be modified according, at least in part, to independently, securely delivered further control information. Said control information may include executable code (e.g., Load modules) that has been certified as acceptable (e.g., reliable and trusted) for use with a specific VDE application, class of applications, and/or a VDE distributed arrangement. This modification (evolution) of control information can occur upon content control information (load modules and any associated data) circulating to one or more VDE participants in a pathway of handling of control information, or it may occur upon control information being received from a VDE participant. Handlers in a pathway of handling of content control information, to the extent each is authorized, can establish, modify, and/or contribute to, permission, auditing, payment, and reporting control information related to controlling, analyzing, paying for, and/or reporting usage of, electronic content and/or appliances (for example, as related to usage of VDE controlled property content). Independently delivered (from an independent source which is independent except in regards to certification), at least in part secure, control information can be employed to securely modify content control information when content control information has flowed from one party to another party in a sequence of VDE content control information handling. This modification employs, for example, one or more VDE component assemblies being securely processed in a VDE secure subsystem. In an alternate embodiment, control information may be modified by a senior party through use of their VDE installation secure sub-system after receiving submitted, at least in part secured, control information from a "junior" party, normally in the form of a VDE administrative object. Control information passing along VDE pathways can represent a mixed control set, in that it may include: control information that persisted through a sequence of control information handlers, other control information that was allowed to be modified, and further control information representing new control information and/or mediating data. Such a control set represents an evolution of control information for disseminated content. In this example the overall content control set for a VDE content container is "evolving" as it securely (e.g. communicated in encrypted form and using authentication and digital signaturing techniques) passes, at least in part, to a new participant's VDE installation where the proposed control information is securely received and handled. The received control information may be integrated (through use of the receiving parties' VDE installation secure sub-system) with in-place control information through a negotiation process involving both control information sets. For example, the modification, within the secure sub-system of a content provider's VDE installation, of content control information for a certain VDE content container may have occurred as a result of the incorporation of required control information provided by a financial credit provider. Said credit provider may have employed their VDE installation to prepare and securely communicate (directly or indirectly) said required control information to said content provider. Incorporating said required control information enables a content provider to allow the credit provider's credit to be employed by a content end-user to compensate for the end-user's use of VDE controlled content and/or appliances, so long as said end-user has a credit account with said financial credit provider and said credit account has sufficient credit available. Similarly, control information requiring the payment of taxes and/or the provision of revenue information resulting from electronic commerce activities may be securely received by a content provider. This control information may be received, for example, from a government agency. Content providers might be required by law to incorporate such control information into the control information for commercially distributed content and/or services related to appliance usage. Proposed control information is used to an extent allowed by senior control information and as determined by any negotiation trade-offs that satisfy priorities stipulated by each set (the received set and the proposed set). VDE also accommodates different control schemes specifically applying to different participants (e.g., individual participants and/or participant classes (types)) in a network of VDE content handling participants.

support multiple simultaneous control models for the same content property and/or property portion. This allows, for example, for concurrent business activities which are dependent on electronic commercial product content distribution, such as acquiring detailed market survey information and/or supporting advertising, both of which can increase revenue and result in lower content costs to users and greater value to content providers. Such control information and/or overall control models may be applied, as determined or allowed by control information, in differing manners to different participants in a pathway of content, reporting, payment, and/or related control information handling. VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities, and/or to different parties in a content and/or appliance usage model, such that different parties (or classes of VDE users, for example) are subject to differing control information managing their use of electronic information content. For example, differing control models based on the category of a user as a distributor of a VDE controlled content object or an end-user of such content may result in different budgets being applied. Alternatively, for example, a one distributor may have the right to distribute a different array of properties than another distributor (from a common content collection provided, for example, on optical disc). An individual,

and/or a class or other grouping of end-users, may have different costs (for example, a student, senior citizen, and/or poor citizen user of content who may be provided with the same or differing discounts) than a "typical" content user.

support provider revenue information resulting from customer use of content and/or appliances, and/or provider and/or end-user payment of taxes, through the transfer of credit and/or electronic currency from said end-user and/or provider to a government agency, might occur "automatically" as a result of such received control information causing the generation of a VDE content container whose content includes customer content usage information reflecting secure, trusted revenue summary information and/or detailed user transaction listings (level of detail might depend, for example on type or size of transaction—information regarding a bank interest payment to a customer or a transfer of a large (e.g. over $10,000) might be, by law, automatically reported to the government). Such summary and/or detailed information related to taxable events and/or currency, and/or creditor currency transfer, may be passed along a pathway of reporting and/or payment to the government in a VDE container. Such a container may also be used for other VDE related content usage reporting information.

support the flowing of content control information through different "branches" of content control information handling so as to accommodate, under the present invention's preferred embodiment, diverse controlled distributions of VDE controlled content. This allows different parties to employ the same initial electronic content with differing (perhaps competitive) control strategies. In this instance, a party who first placed control information on content can make certain control assumptions and these assumptions would evolve into more specific and/or extensive control assumptions. These control assumptions can evolve during the branching sequence upon content model participants submitting control information changes, for example, for use in "negotiating" with "in place" content control information. This can result in new or modified content control information and/or it might involve the selection of certain one or more already "in-place" content usage control methods over in-place alternative methods, as well as the submission of relevant control information parameter data. This form of evolution of different control information sets applied to different copies of the same electronic property content and/or appliance results from VDE control information flowing "down" through different branches in an overall pathway of handling and control and being modified differently as it diverges down these different pathway branches. This ability of the present invention to support multiple pathway branches for the flow of both VDE content control information and VDE managed content enables an electronic commerce marketplace which supports diverging, competitive business partnerships, agreements, and evolving overall business models which can employ the same content properties combined, for example, in differing collections of content representing differing at least in part competitive products.

enable a user to securely extract, through the use of the secure subsystem at the user's VDE installation, at least a portion of the content included within a VDE content container to produce a new, secure object (content container), such that the extracted information is maintained in a continually secure manner through the extraction process. Formation of the new VDE container containing such extracted content shall result in control information consistent with, or specified by, the source VDE content container, and/or local VDE installation secure subsystem as appropriate, content control information. Relevant control information, such as security and administrative information, derived, at least in part, from the parent (source) object's control information, will normally be automatically inserted into a new VDE content container object containing extracted VDE content. This process typically occurs under the control framework of a parent object and/or VDE installation control information executing at the user's VDE installation secure subsystem (with, for example, at least a portion of this inserted control information being stored securely in encrypted form in one or more permissions records). In an alternative embodiment, the derived content control information applied to extracted content may be in part or whole derived from, or employ, content control information stored remotely from the VDE installation that performed the secure extraction such as at a remote server location. As with the content control information for most VDE managed content, features of the present invention allows the content's control information to:

(a) "evolve," for example, the extractor of content may add new control methods and/or modify control parameter data, such as VDE application compliant methods, to the extent allowed by the content's in-place control information. Such new control information might specify, for example, who may use at least a portion of the new object, and/or how said at least a portion of said extracted content may be used (e.g. when at least a portion may be used, or what portion or quantity of portions may be used);

(b) allow a user to combine additional content with at least a portion of said extracted content, such as material authored by the extractor and/or content (for example, images, video, audio, and/or text) extracted from one or more other VDE container objects for placement directly into the new container;

(c) allow a user to securely edit at least a portion of said content while maintaining said content in a secure form within said VDE content container;

(d) append extracted content to a pre-existing VDE content container object and attach associated control information—in these cases, user added information may be secured, e.g., encrypted, in part or as a whole, and may be subject to usage and/or auditing control information that differs from the those applied to previously in place object content;

(e) preserve VDE control over one or more portions of extracted content after various forms of usage of said portions, for example, maintain content in securely stored form while allowing "temporary" on screen display of content or allowing a software program to be maintained in secure form but transiently decrypt any encrypted executing portion of said program (all, or only a portion, of said program may be encrypted to secure the program).

Generally, the extraction features of the present invention allow users to aggregate and/or disseminate and/or otherwise use protected electronic content information extracted from content container sources while maintaining secure VDE capabilities thus preserving the

rights of providers in said content information after various content usage processes.

support the aggregation of portions of VDE controlled content, such portions being subject to differing VDE content container control information, wherein various of said portions may have been provided by independent, different content providers from one or more different locations remote to the user performing the aggregation. Such aggregation, in the preferred embodiment of the present invention, may involve preserving at least a portion of the control information (e.g., executable code such as load modules) for each of various of said portions by, for example, embedding some or all of such portions individually as VDE content container objects within an overall VDE content container and/or embedding some or all of such portions directly into a VDE content container. In the latter case, content control information of said content container may apply differing control information sets to various of such portions based upon said portions original control information requirements before aggregation. Each of such embedded VDE content containers may have its own control information in the form of one or more permissions records. Alternatively, a negotiation between control information associated with various aggregated portions of electronic content, may produce a control information set that would govern some or all of the aggregated content portions. The VDE content control information produced by the negotiation may be uniform (such as having the same load modules and/or component assemblies, and/or it may apply differing such content control information to two or more portions that constitute an aggregation of VDE controlled content such as differing metering, budgeting, billing and/or payment models. For example, content usage payment may be automatically made, either through a clearinghouse, or directly, to different content providers for different potions.

enable flexible metering of, or other collection of information related to, use of electronic content and/or electronic appliances. A feature of the present invention enables such flexibility of metering control mechanisms to accommodate a simultaneous, broad array of: (a) different parameters related to electronic information content use; (b) different increment units bytes, documents, properties, paragraphs, images, etc.) and/or other organizations of such electronic content; and/or (c) different categories of user and/or VDE installation types, such as client organizations, departments, projects, networks, and/or individual users, etc. This feature of the present invention can be employed for content security, usage analysis (for example, market surveying), and/or compensation based upon the use and/or exposure to VDE managed content. Such metering is a flexible basis for ensuring payment for content royalties, licensing, purchasing, and/or advertising. A feature of the present invention provides for payment means supporting flexible electronic currency and credit mechanisms, including the ability to securely maintain audit trails reflecting information related to use of such currency or credit. VDE supports multiple differing hierarchies of client organization control information wherein an organization client administrator distributes control information specifying the usage rights of departments, users, and/or projects. Likewise, a department (division) network manager can function as a distributor (budgets, access rights, etc.) for department networks, projects, and/or users, etc.

provide scalable, integratable, standardized control means for use on electronic appliances ranging from inexpensive consumer (for example, television set-top appliances) and professional devices (and hand-held PDAs) to servers, mainframes, communication switches, etc. The scalable transaction management/auditing technology of the present invention will result in more efficient and reliable interoperability amongst devices functioning in electronic commerce and/or data security environments. As standardized physical containers have become essential to the shipping of physical goods around the world, allowing these physical containers to universally "fit" unloading equipment, efficiently use truck and train space, and accommodate known arrays of objects (for example, boxes) in an efficient manner, so VDE electronic content containers may, as provided by the present invention, be able to efficiently move electronic information content (such as commercially published properties, electronic currency and credit, and content audit information), and associated content control information, around the world. Interoperability is fundamental to efficient electronic commerce. The design of the VDE foundation, VDE load modules, and VDE containers, are important features that enable the VDE node operating environment to be compatible with a very broad range of electronic appliances. The ability, for example, for control methods based on load modules to execute in very "small" and inexpensive secure sub-system environments, such as environments with very little read/write memory, while also being able to execute in large memory sub-systems that may be used in more expensive electronic appliances, supports consistency across many machines. This consistent VDE operating environment, including its control structures and container architecture, enables the use of standardized VDE content containers across a broad range of device types and host operating environments. Since VDE capabilities can be seamlessly integrated as extensions, additions, and/or modifications to fundamental capabilities of electronic appliances and host operating systems, VDE containers, content control information, and the VDE foundation will be able to work with many device types and these device types will be able to consistently and efficiently interpret and enforce VDE control information. Through this integration users can also benefit from a transparent interaction with many of the capabilities of VDE. VDE integration with software operating on a host electronic appliance supports a variety of capabilities that would be unavailable or less secure without such integration. Through integration with one or more device applications and/or device operating environments, many capabilities of the present invention can be presented as inherent capabilities of a given electronic appliance, operating system, or appliance application. For example, features of the present invention include: (a) VDE system software to in part extend and/or modify host operating systems such that they possesses VDE capabilities, such as enabling secure transaction processing and electronic information storage; (b) one or more application programs that in part represent tools associated with VDE operation; and/or (c) code to be integrated into application programs, wherein such code incorporates references into VDE system software to integrate VDE capabilities and makes such applications VDE aware (for example, word processors, database

retrieval applications, spreadsheets, multimedia presentation authoring tools, film editing software, music editing software such as MIDI applications and the like, robotics control systems such as those associated with CAD/CAM environments and NCM software and the like, electronic mail systems, teleconferencing software, and other data authoring, creating, handling, and/or usage applications including combinations of the above). These one or more features (which may also be implemented in firmware or hardware) may be employed in conjunction with a VDE node secure hardware processing capability, such as a microcontroller(s), microprocessor(s), other CPU(s) or other digital processing logic.

employ audit reconciliation and usage pattern evaluation processes that assess, through certain, normally network based, transaction processing reconciliation and threshold checking activities, whether certain violations of security of a VDE arrangement have occurred. These processes are performed remote to VDE controlled content end-user VDE locations by assessing, for example, purchases, and/or requests, for electronic properties by a given VDE installation. Applications for such reconciliation activities include assessing whether the quantity of remotely delivered VDE controlled content corresponds to the amount of financial credit and/or electronic currency employed for the use of such content. A trusted organization can acquire information from content providers concerning the cost for content provided to a given VDE installation and/or user and compare this cost for content with the credit and/or electronic currency disbursements for that installation and/or user. Inconsistencies in the amount of content delivered versus the amount of disbursement can prove, and/or indicate, depending on the circumstances, whether the local VDE installation has been, at least to some degree, compromised (for example, certain important system security functions, such as breaking encryption for at least some portion of the secure subsystem and/or VDE controlled content by uncovering one or more keys). Determining whether irregular patterns (e.g. unusually high demand) of content usage, or requests for delivery of certain kinds of VDE controlled information during a certain time period by one or more VDE installations and/or users (including, for example, groups of related users whose aggregate pattern of usage is suspicious) may also be useful in determining whether security at such one or more installations, and/or by such one or more users, has been compromised, particularly when used in combination with an assessment of electronic credit and/or currency provided to one or more VDE users and/or installations, by some or all of their credit and/or currency suppliers, compared with the disbursements made by such users and/or installations.

support security techniques that materially increase the time required to "break" a system's integrity. This includes using a collection of techniques that minimizes the damage resulting from comprising some aspect of the security features of the present inventions.

provide a family of authoring, administrative, reporting, payment, and billing tool user applications that comprise components of the present invention's trusted/secure, universe wide, distributed transaction control and administration system. These components support VDE related: object creation (including placing control information on content), secure object distribution and management (including distribution control information, financial related, and other usage analysis), client internal VDE activities administration and control, security management, user interfaces, payment disbursement, and clearinghouse related functions. These components are designed to support highly secure, uniform, consistent, and standardized: electronic commerce and/or data security pathway(s) of handling, reporting, and/or payment; content control and administration; and human factors (e.g. user interfaces).

support the operation of a plurality of clearinghouses, including, for example, both financial and user clearinghouse activities, such as those performed by a client administrator in a large organization to assist in the organization's use of a VDE arrangement, including usage information analysis, and control of VDE activities by individuals and groups of employees such as specifying budgets and the character of usage rights available under VDE for certain groups of and/or individual, client personnel, subject to control information series to control information submitted by the client administrator. At a clearinghouse, one or more VDE installations may operate together with a trusted distributed database environment (which may include concurrent database processing means). A financial clearinghouse normally receives at its location securely delivered content usage information, and user requests (such as requests for further credit, electronic currency, and/or higher credit limit). Reporting of usage information and user requests can be used for supporting electronic currency, billing, payment and credit related activities, and/or for user profile analysis and/or broader market survey analysis and marketing (consolidated) list generation or other information derived, at least in part, from said usage information. this information can be provided to content providers or other parties, through secure, authenticated encrypted communication to the VDE installation secure subsystems. Clearinghouse processing means would normally be connected to specialized I/O means, which may include high speed telecommunication switching means that may be used for secure communications between a clearinghouse and other VDE pathway participants.

securely support electronic currency and credit usage control, storage, and communication at, and between, VDE installations. VDE further supports automated passing of electronic currency and/or credit information, including payment tokens (such as in the form of electronic currency or credit) or other payment information, through a pathway of payment, which said pathway may or may not be the same as a pathway for content usage information reporting. Such payment may be placed into a VDE container created automatically by a VDE installation in response to control information stipulating the "withdrawal" of credit or electronic currency from an electronic credit or currency account based upon an amount owed resulting from usage of VDE controlled electronic content and/or appliances. Payment credit or currency may then be automatically communicated in protected (at least in part encrypted) form through telecommunication of a VDE container to an appropriate party such as a clearinghouse, provider of original property content or appliance, or an agent for such provider (other than a clearinghouse). Payment information may be packaged

in said VDE content container with, or without, related content usage information, such as metering information. An aspect of the present invention further enables certain information regarding currency use to be specified as unavailable to certain, some, or all VDE parties ("conditionally" to filly anonymous currency) and/or further can regulate certain content information, such as currency and/or credit use related information (and/or other electronic information usage data) to be available only under certain strict circumstances, such as a court order (which may itself require authorization through the use of a court controlled VDE installation that may be required to securely access "conditionally" anonymous information). Currency and credit information, under the preferred embodiment of the present invention, is treated as administrative content;

support fingerprinting (also known as watermarking) for embedding in content such that when content protected under the present invention is released in clear form from a VDE object (displayed, printed, communicated, extracted, and/or saved), information representing the identification of the user and/or VDE installation responsible for transforming the content into clear for is embedded into the released content. Fingerprinting is useful in providing an ability to identify who extracted information in clear form a VDE container, or who made a copy of a VDE object or a portion of its contents. Since the identity of the user and/or other identifying information may be embedded in an obscure or generally concealed manner, in VDE container content and/or control information, potential copyright violators may be deterred from unauthorized extraction or copying. Fingerprinting normally is embedded into unencrypted electronic content or control information, though it can be embedded into encrypted content and later placed in unencrypted content in a secure VDE installation sub-system as the encrypted content carrying the fingerprinting information is decrypted. Electronic information, such as the content of a VDE container, may be fingerprinted as it leaves a network (such as Internet) location bound for a receiving party. Such repository information may be maintained in unencrypted form prior to communication and be encrypted as it leaves the repository. Fingerprinting would preferably take place as the content leaves the repository, but before the encryption step. Encrypted repository content can be decrypted, for example in a secure VDE sub-system, fingerprint information can be inserted, and then the content can be re-encrypted for transmission. Embedding identification information of the intended recipient user and/or VDE installation into content as it leaves, for example, an Internet repository, would provide important information that would identify or assist in identifying any party that managed to compromise the security of a VDE installation or the delivered content. If a party produces an authorized clear form copy of VDE controlled content, including making unauthorized copies of an authorized clear form copy, fingerprint information would point back to that individual and/or his or her VDE installation. Such hidden information will act as a strong disincentive that should dissuade a substantial portion of potential content "pirates" from stealing other parties electronic information. Fingerprint information identifying a receiving party and/or VDE installation can be embedded into a VDE object before, or during, decryption, replication, or communication of VDE content objects to receivers. Fingerprinting electronic content before it is encrypted for transfer to a customer or other user provides information that can be very useful for identifying who received certain content which may have then been distributed or made available in unencrypted form. This information would be useful in tracking who may have "broken" the security of a VDE installation and was illegally making certain electronic content available to others. Fingerprinting may provide additional, available information such as time and/or date of the release (for example extraction) of said content information. Locations for inserting fingerprints may be specified by VDE installation and/or content container control information. This information may specify that certain areas and/or precise locations within properties should be used for fingerprinting, such as one or more certain fields of information or information types. Fingerprinting information may be incorporated into a property by modifying in a normally undetectable way color frequency and/or the brightness of certain image pixels, by slightly modifying certain audio signals as to frequency, by modifying font character formation, etc. Fingerprint information, itself, should be encrypted so as to make it particularly difficult for tampered fingerprints to be interpreted as valid. Variations in fingerprint locations for different copies of the same property; "false" fingerprint information; and multiple copies of fingerprint information within a specific property or other content which copies employ different fingerprinting techniques such as information distribution patterns, frequency and/or brightness manipulation, and encryption related techniques, are features of the present invention for increasing the difficulty of an unauthorized individual identifying fingerprint locations and erasing and/or modifying fingerprint information.

provide smart object agents that can carry requests, data, and/or methods, including budgets, authorizations, credit or currency, and content. For example, smart objects may travel to and/or from remote information resource locations and fulfill requests for electronic information content. Smart objects can, for example, be transmitted to a remote location to perform a specified database search on behalf of a user or otherwise "intelligently" search remote one or more repositories of information for user desired information. After identifying desired information at one or more remote locations, by for example, performing one or more database searches, a smart object may return via communication to the user in the form of a secure "return object" containing retrieved information. A user may be charged for the remote retrieving of information, the returning of information to the user's VDE installation, and/or the use of such information. In the latter case, a user may be charged only for the information in the return object that the user actually uses. Smart objects may have the means to request use of one or more services and/or resources. Services include locating other services and/or resources such as information resources, language or format translation, processing, credit (or additional credit) authorization, etc. Resources include reference databases, networks, high powered or specialized computing resources (the smart object may carry information to another computer to be efficiently processed and then return the information to the sending VDE installation), remote object

repositories, etc. Smart objects can make efficient use of remote resources (e.g. centralized databases, super computers, etc.) while providing a secure means for charging users based on information and/or resources actually used.

support both "translations" of VDE electronic agreements elements into modern language printed agreement elements (such as English language agreements) and translations of electronic rights protection/transaction management modern language agreement elements to electronic VDE agreement elements. This feature requires maintaining a library of textual language that corresponds to VDE load modules and/or methods and/or component assemblies. As VDE methods are proposed and/or employed for VDE agreements, a listing of textual terms and conditions can be produced by a VDE user application which, in a preferred embodiment, provides phrases, sentences and/or paragraphs that have been stored and correspond to said methods and/or assemblies. This feature preferably employs artificial intelligence capabilities to analyze and automatically determine, and/or assist one or more users to determine, the proper order and relationship between the library elements corresponding to the chosen methods and/or assemblies so as to compose some or all portions of a legal or descriptive document. One or more users, and/or preferably an attorney (if the document a legal, binding agreement), would review the generated document material upon completion and employ such additional textual information and/or editing as necessary to describe non electronic transaction elements of the agreement and make any other improvements that may be necessary. These features further support employing modern language tools that allow one or more users to make selections from choices and provide answers to questions and to produce a VDE electronic agreement from such a process. This process can be interactive and the VDE agreement formulation process may employ artificial intelligence expert system technology that learns from responses and, where appropriate and based at least in part on said responses, provides further choices and/or questions which "evolves" the desired VDE electronic agreement.

support the use of multiple VDE secure subsystems in a single VDE installation. Various security and/or performance advantages may be realized by employing a distributed VDE design within a single VDE installation. For example, designing a hardware based VDE secure subsystem into an electronic appliance VDE display device, and designing said subsystem's integration with said display device so that it is as close as possible to the point of display, will increase the security for video materials by making it materially more difficult to "steal" decrypted video information as it moves from outside to inside the video system. Ideally, for example, a VDE secure hardware module would be in the same physical package as the actual display monitor, such as within the packaging of a video monitor or other display device, and such device would be designed, to the extent commercially practical, to be as tamper resistant as reasonable. As another example, embedding a VDE hardware module into an I/O peripheral may have certain advantages from the standpoint of overall system throughput. If multiple VDE instances are employed within the same VDE installation, these instances will ideally share resources to the extent practical, such as VDE instances storing certain control information and content and/or appliance usage information on the same mass storage device and in the same VDE management database.

requiring reporting and payment compliance by employing exhaustion of budgets and time ageing of keys. For example, a VDE commercial arrangement and associated content control information may involve a content provider's content and the use of clearinghouse credit for payment for end-user usage of said content. Control information regarding said arrangement may be delivered to a user's (of said content) VDE installation and/or said financial clearinghouse's VDE installation. Said control information might require said clearinghouse to prepare and telecommunicate to said content provider both content usage based information in a certain form, and content usage payment in the form of electronic credit (such credit might be "owned" by the provider after receipt and used in lieu of the availability or adequacy of electronic currency) and/or electronic currency. This delivery of information and payment may employ trusted VDE installation secure subsystems to securely, and in some embodiments, automatically, provide in the manner specified by said control information, said usage information and payment content. Features of the present invention help ensure that a requirement that a clearinghouse report such usage information and payment content will be observed. For example, if one participant to a VDE electronic agreement fails to observe such information reporting and/or paying obligation, another participant can stop the delinquent party from successfully participating in VDE activities related to such agreement. For example, if required usage information and payment was not reported as specified by content control information, the "injured" party can fail to provide, through failing to securely communicate from his VDE installation secure subsystem, one or more pieces of secure information necessary for the continuance of one or more critical processes. For example, failure to report information and/or payment from a clearinghouse to a content provider (as well as any security failures or other disturbing irregularities) can result in the content provider not providing key and/or budget refresh information to the clearinghouse, which information can be necessary to authorize use of the clearinghouse's credit for usage of the provider's content and which the clearinghouse would communicate to end-user's during a content usage reporting communication between the clearinghouse and end-user. As another example, a distributor that failed to make payments and/or report usage information to a content provider might find that their budget for creating permissions records to distribute the content provider's content to users, and/or a security budget limiting one or more other aspect of their use of the provider's content, are not being refreshed by the content provider, once exhausted or timed-out (for example, at a predetermined date). In these and other cases, the offended party might decide not to refresh time ageing keys that had "aged out." Such a use of time aged keys has a similar impact as failing to refresh budgets or time-aged authorizations.

support smart card implementations of the present invention in the form of portable electronic appliances, including cards that can be employed as secure credit, banking, and/or money cards. A feature of the present

invention is the use of portable VDEs as transaction cards at retail and other establishments, wherein such cards can "dock" with an establishment terminal that has a VDE secure sub-system and/or an online connection to a VDE secure and/or otherwise secure and compatible subsystem, such as a "trusted" financial clearinghouse (e.g., VISA, Mastercard). The VDE card and the terminal (and/or online connection) can securely exchange information related to a transaction, with credit and/or electronic currency being transferred to a merchant and/or clearinghouse and transaction information flowing back to the card. Such a card can be used for transaction activities of all sorts. A docking station, such as a PCMCIA connector on an electronic appliance, such as a personal computer, can receive a consumer's VDE card at home. Such a station/card combination can be used for on-line transactions in the same manner as a VDE installation that is permanently installed in such an electronic appliance. The card can be used as an "electronic wallet" and contain electronic currency as well as credit provided by a clearinghouse. The card can act as a convergence point for financial activities of a consumer regarding many, if not all, merchant, banking, and on-line financial transactions, including supporting home banking activities. A consumer can receive his paycheck and/or investment earnings and/or "authentic" VDE content container secured detailed information on such receipts, through on-line connections. A user can send digital currency to another party with a VDE arrangement, including giving away such currency. A VDE card can retain details of transactions in a highly secure and database organized fashion so that financially related information is both consolidated and very easily retrieved and/or analyzed. Because of the VDE security, including use of effective encryption, authentication, digital signaturing, and secure database structures, the records contained within a VDE card arrangement may be accepted as valid transaction records for government and/or corporate recordkeeping requirements. In some embodiments of the present invention a VDE card may employ docking station and/or electronic appliance storage means and/or share other VDE arrangement means local to said appliance and/or available across a network, to augment the information storage capacity of the VDE card, by for example, storing dated, and/or archived, backup information. Taxes relating to some or all of an individual's financial activities may be automatically computed based on "authentic" information securely stored and available to said VDE card. Said information may be stored in said card, in said docking station, in an associated electronic appliance, and/or other device operatively attached thereto, and/or remotely, such as at a remote server site. A card's data, e.g. transaction history, can be backed up to an individual's personal computer or other electronic appliance and such an appliance may have an integrated VDE installation of its own. A current transaction, recent transactions (for redundancy), or all or other selected card data may be backed up to a remote backup repository, such a VDE compatible repository at a financial clearinghouse, during each or periodic docking for a financial transaction and/or information communication such as a user/merchant transaction. Backing up at least the current transaction during a connection with another party's VDE installation (for example a VDE installation that is also on a financial or general purpose electronic network), by posting transaction information to a remote clearinghouse and/or bank, can ensure that sufficient backup is conducted to enable complete reconstruction of VDE card internal information in the event of a card failure or loss.

support certification processes that ensure authorized interoperability between various VDE installations so as to prevent VDE arrangements and/or installations that unacceptably deviate in specification protocols from other VDE arrangements and/or installations from interoperating in a manner that may introduce security (integrity and/or confidentiality of VDE secured information), process control, and/or software compatibility problems. Certification validates the identity of VDE installations and/or their components, as well as VDE users. Certification data can also serve as information that contributes to determining the decommissioning or other change related to VDE sites.

support the separation of fundamental transaction control processes through the use of event (triggered) based method control mechanisms. These event methods trigger one or more other VDE methods (which are available to a secure VDE sub-system) and are used to carry out VDE managed transaction related processing. These triggered methods include independently (separably) and securely processable component billing management methods, budgeting management methods, metering management methods, and related auditing management processes. As a result of this feature of the present invention, independent triggering of metering, auditing, billing, and budgeting methods, the present invention is able to efficiently, concurrently support multiple financial currencies (e.g. dollars, marks, yen) and content related budgets, and/or billing increments as well as very flexible content distribution models.

support, complete, modular separation of the control structures related to (1) content event triggering, (2) auditing, (3) budgeting (including specifying no right of use or unlimited right of use), (4) billing, and (5) user identity (VDE installation, client name, department, network, and/or user, etc.). The independence of these VDE control structures provides a flexible system which allows plural relationships between two or more of these structures, for example, the ability to associate a financial budget with different event trigger structures (that are put in place to enable controlling content based on its logical portions). Without such separation between these basic VDE capabilities, it would be more difficult to efficiently maintain separate metering, budgeting, identification, and/or billing activities which involve the same, differing (including overlapping), or entirely different, portions of content for metering, billing, budgeting, and user identification, for example, paying fees associated with usage of content, performing home banking, managing advertising services, etc. VDE modular separation of these basic capabilities supports the programming of plural, "arbitrary" relationships between one or differing content portions (and/or portion units) and budgeting, auditing, and/or billing control information. For example, under VDE, a budget limit of $200 dollars or 300 German Marks a month may be enforced for decryption of a certain database and 2 U.S. Dollars or 3 German Marks may be charged for each record of said database decrypted (depending on user selected currency). Such usage can be metered while an additional audit for user profile

purposes can be prepared recording the identity of each filed displayed. Additionally, further metering can be conducted regarding the number of said database bytes that have been decrypted, and a related security budget may prevent the decrypting of more than 5% of the total bytes of said database per year. The user may also, under VDE (if allowed by senior control information), collect audit information reflecting usage of database fields by different individuals and client organization departments and ensure that differing rights of access and differing budgets limiting database usage can be applied to these client individuals and groups. Enabling content providers and users to practically employ such diverse sets of user identification, metering, budgeting, and billing control information results, in part, from the use of such independent control capabilities. As a result, VDE can support great configurability in creation of plural control models applied to the same electronic property and the same and/or plural control models applied to differing or entirely different content models (for example, home banking versus electronic shopping).

Methods, Other Control Information, and VDE Objects

VDE control information (e.g., methods) that collectively control use of VDE managed properties (database, document, individual commercial product), are either shipped with the content itself (for example, in a content container) and/or one or more portions of such control information is shipped to distributors and/or other users in separably deliverable "administrative objects." A subset of the methods for a property may in part be delivered with each property while one or more other subsets of methods can be delivered separately to a user or otherwise made available for use (such as being available remotely by telecommunication means). Required methods (methods listed as required for property and/or appliance use) must be available as specified if VDE controlled content (such as intellectual property distributed within a VDE content container) is to be used. Methods that control content may apply to a plurality of VDE container objects, such as a class or other grouping of such objects. Methods may also be required by certain users or classes of users and/or VDE installations and/or classes of installations for such parties to use one or more specific, or classes of, objects.

A feature of VDE provided by the present invention is that certain one or more methods can be specified as required in order for a VDE installation and/or user to be able to use certain and/or all content. For example, a distributor of a certain type of content might be allowed by "senior" participants (by content creators, for example) to require a method which prohibits end-users from electronically saving decrypted content, a provider of credit for VDE transactions might require an audit method that records the time of an electronic purchase, and/or a user might require a method that summarizes usage information for reporting to a clearinghouse (e.g. billing information) in a way that does not convey confidential, personal information regarding detailed usage behavior.

A further feature of VDE provided by the present invention is that creators, distributors, and users of content can select from among a set of predefined methods (if available) to control container content usage and distribution functions and/or they may have the right to provide new customized methods to control at least certain usage functions (such "new" methods may be required to be certified for trustedness and interoperability to the VDE installation and/or for of a group of VDE applications). As a result, VDE provides a very high degree of configurability with respect to how the distribution and other usage of each property or object (or one or more portions of objects or properties as desired and/or applicable) will be controlled. Each VDE participant in a VDE pathway of content control information may set methods for some or all of the content in a VDE container, so long as such control information does not conflict with senior control information already in place with respect to:

(1) certain or all VDE managed content,

(2) certain one or more VDE users and/or groupings of users,

(3) certain one or more VDE nodes and/or groupings of nodes, and/or

(4) certain one or more VDE applications and/or arrangements.

For example, a content creator's VDE control information for certain content can take precedence over other submitted VDE participant control information and, for example, if allowed by senior control information, a content distributor's control information may itself take precedence over a client administrator's control information, which may take precedence over an end-user's control information. A path of distribution participant's ability to set such electronic content control information can be limited to certain control information (for example, method mediating data such as pricing and/or sales dates) or it may be limited only to the extent that one or more of the participant's proposed control information conflicts with control information set by senior control information submitted previously by participants in a chain of handling of the property, or managed in said participant's VDE secure subsystem.

VDE control information may, in part or in full, (a) represent control information directly put in place by VDE content control information pathway participants, and/or (b) comprise control information put in place by such a participant on behalf of a party who does not directly handle electronic content (or electronic appliance) permissions records information (for example control information inserted by a participant on behalf of a financial clearinghouse or government agency). Such control information methods (and/or load modules and/or mediating data and/or component assemblies) may also be put in place by either an electronic automated, or a semi-automated and human assisted, control information (control set) negotiating process that assesses whether the use of one or more pieces of submitted control information will be integrated into and/or replace existing control information (and/or chooses between alternative control information based upon interaction with in-place control information) and how such control information may be used.

Control information may be provided by a party who does not directly participate in the handling of electronic content (and/or appliance) and/or control information for such content (and/or appliance). Such control information may be provided in secure form using VDE installation secure sub-system managed communications (including, for example, authenticating the deliverer of at least in part encrypted control information) between such not directly participating one or more parties' VDE installation secure subsystems, and a pathway of VDE content control information participant's VDE installation secure subsystem. This control information may relate to, for example, the right to access credit supplied by a financial services provider, the enforcement of regulations or laws enacted by a government agency, or the requirements of a customer of VDE managed content usage information (reflecting usage of content by one or more parties other than such customer)

relating to the creation, handling and/or manner of reporting of usage information received by such customer. Such control information may, for example, enforce societal requirements such as laws related to electronic commerce.

VDE content control information may apply differently to different pathway of content and/or control information handling participants. Furthermore, permissions records rights may be added, altered, and/or removed by a VIDE participant if they are allowed to take such action. Rights of VDE participants may be defined in relation to specific parties and/or categories of parties and/or other groups of parties in a chain of handling of content and/or content control information (e.g., permissions records). Modifications to control information that may be made by a given, eligible party or parties, may be limited in the number of modifications, and/or degree of modification, they may make.

At least one secure subsystem in electronic appliances of creators, distributors, auditors, clearinghouses, client administrators, and end-users (understanding that two or more of the above classifications may describe a single user) provides a "sufficiently" secure (for the intended applications) environment for:

1. Decrypting properties and control information;
2. Storing control and metering related information;
3. Managing communications;
4. Processing core control programs, along with associated data, that constitute control information for electronic content and/or appliance rights protection, including the enforcing of preferences and requirements of VDE participants.

Normally, most usage, audit, reporting, payment, and distribution control methods are themselves at least in part encrypted and are executed by the secure subsystem of a VDE installation. Thus, for example, billing and metering records can be securely generated and updated, and encryption and decryption keys are securely utilized, within a secure subsystem. Since VDE also employs secure (e.g. encrypted and authenticated) communications when passing information between the participant location (nodes) secure subsystems of a VDE arrangement, important components of a VDE electronic agreement can be reliably enforced with sufficient security (sufficiently trusted) for the intended commercial purposes. A VDE electronic agreement for a value chain can be composed, at least in part, of one or more subagreements between one or more subsets of the value chain participants. These subagreements are comprised of one or more electronic contract "compliance" elements (methods including associated parameter data) that ensure the protection of the rights of VDE participants.

The degree of trustedness of a VDE arrangement will be primarily based on whether hardware SPUs are employed at participant location secure subsystems and the effectiveness of the SPU hardware security architecture, software security techniques when an SPU is emulated in software, and the encryption algorithm(s) and keys that are employed for securing content, control information, communications, and access to VDE node (VDE installation) secure subsystems. Physical facility and user identity authentication security procedures may be used instead of hardware SPUs at certain nodes, such as at an established financial clearinghouse, where such procedures may provide sufficient security for trusted interoperability with a VDE arrangement employing hardware SPUs at user nodes.

The updating of property management files at each location of a VDE arrangement, to accommodate new or modified control information, is performed in the VDE secure subsystem and under the control of secure management file updating programs executed by the protected subsystem. Since all secure communications are at least in part encrypted and the processing inside the secure subsystem is concealed from outside observation and interference, the present invention ensures that content control information can be enforced. As a result, the creator and/or distributor and/or client administrator and/or other contributor of secure control information for each property (for example, an end-user restricting the kind of audit information he or she will allow to be reported and/or a financial clearinghouse establishing certain criteria for use of its credit for payment for use of distributed content) can be confident that their contributed and accepted control information will be enforced (within the security limitations of a given VDE security implementation design). This control information can determine, for example:

(1) How and/or to whom electronic content can be provided, for example, how an electronic property can be distributed;

(2) How one or more objects and/or properties, or portions of an object or property, can be directly used, such as decrypted, displayed, printed, etc;

(3) How payment for usage of such content and/or content portions may or must be handled; and

(4) How audit information about usage information related to at least a portion of a property should be collected, reported, and/or used.

Seniority of contributed control information, including resolution of conflicts between content control information submitted by multiple parties, is normally established by:

(1) the sequence in which control information is put in place by various parties (in place control information normally takes precedence over subsequently submitted control information),

(2) the specifics of VDE content and/or appliance control information. For example, in-place control information can stipulate which subsequent one or more piece of control from one or more parties or class of parties will take precedence over control information submitted by one or more yet different parties and/or classes of parties, and/or

(3) negotiation between control information sets from plural parties, which negotiation establishes what control information shall constitute the resulting control information set for a given piece of VDE managed content and/or VDE installation.

Electronic Agreements and Rights Protection

An important feature of VDE is that it can be used to assure the administration of, and adequacy of security and rights protection for, electronic agreements implemented through the use of the present invention. Such agreements may involve one or more of:

(1) creators, publishers, and other distributors, of electronic information,

(2) financial service (e.g. credit) providers,

(3) users of (other than financial service providers) information arising from content usage such as content specific demographic information and user specific descriptive information. Such users may include market analysts, marketing list compilers for direct and directed marketing, and government agencies,

(4) end users of content,

(5) infrastructure service and device providers such as telecommunication companies and hardware manufac-

turers (semiconductor and electronic appliance and/or other computer system manufacturers) who receive compensation based upon the use of their services and/or devices, and

(6) certain parties described by electronic information.

VDE supports commercially secure "extended" value chain electronic agreements. VDE can be configured to support the various underlying agreements between parties that comprise this extended agreement. These agreements can define important electronic commerce considerations including:

(1) security,

(2) content use control, including electronic distribution,

(3) privacy (regarding, for example, information concerning parties described by medical, credit, tax, personal, and/or of other forms of confidential information),

(4) management of financial processes, and

(5) pathways of handling for electronic content, content and/or appliance control information, electronic content and/or appliance usage information and payment and/or credit.

VDE agreements may define the electronic commerce relationship of two or more parties of a value chain, but such agreements may, at times, not directly obligate or otherwise directly involve other VDE value chain participants. For example, an electronic agreement between a content creator and a distributor may establish both the price to the distributor for a creator's content (such as for a property distributed in a VDE container object) and the number of copies of this object that this distributor may distribute to end-users over a given period of time. In a second agreement, a value chain end-user may be involved in a three party agreement in which the end-user agrees to certain requirements for using the distributed product such as accepting distributor charges for content use and agreeing to observe the copyright rights of the creator. A third agreement might exist between the distributor and a financial clearinghouse that allows the distributor to employ the clearinghouse's credit for payment for the product if the end-user has a separate (fourth) agreement directly with the clearinghouse extending credit to the end-user. A fifth, evolving agreement may develop between all value chain participants as content control information passes along its chain of handling. This evolving agreement can establish the rights of all parties to content usage information, including, for example, the nature of information to be received by each party and the pathway of handling of content usage information and related procedures. A sixth agreement in this example, may involve all parties to the agreement and establishes certain general assumptions, such as security techniques and degree of trustedness (for example, commercial integrity of the system may require each VDE installation secure subsystem to electronically warrant that their VDE node meets certain interoperability requirements). In the above example, these six agreements could comprise agreements of an extended agreement for this commercial value chain instance.

VDE agreements support evolving ("living") electronic agreement arrangements that can be modified by current and/or new participants through very simple to sophisticated "negotiations" between newly proposed content control information interacting with control information already in place and/or by negotiation between concurrently proposed content control information submitted by a plurality of parties. A given model may be asynchronously and progressively modified over time in accordance with existing senior rules and such modification may be applied to all, to classes of, and/or to specific content, and/or to classes and/or specific users and/or user nodes. A given piece of content may be subject to different control information at different times or places of handling, depending on the evolution of its content control information (and/or on differing, applicable VDE installation content control information). The evolution of control information can occur during the passing along of one or more VDE control information containing objects, that is control information may be modified at one or more points along a chain of control information handling, so long as such modification is allowed. As a result, VDE managed content may have different control information applied at both different "locations" in a chain of content handling and at similar locations in differing chains of the handling of such content. Such different application of control information may also result from content control information specifying that a certain party or group of parties shall be subject to content control information that differs from another party or group of parties. For example, content control information for a given piece of content may be stipulated as senior information and therefore not changeable, might be put in place by a content creator and might stipulate that national distributors of a given piece of their content may be permitted to make 100,000 copies per calendar quarter, so long as such copies are provided to boni fide end-users, but may pass only a single copy of such content to a local retailers and the control information limits such a retailer to making no more than 1,000 copies per month for retail sales to end-users. In addition, for example, an end-user of such content might be limited by the same content control information to making three copies of such content, one for each of three different computers he or she uses (one desktop computer at work, one for a desktop computer at home, and one for a portable computer).

Electronic agreements supported by the preferred embodiment of the present invention can vary from very simple to very elaborate. They can support widely diverse information management models that provide for electronic information security, usage administration, and communication and may support:

(a) secure electronic distribution of information, for example commercial literary properties,

(b) secure electronic information usage monitoring and reporting,

(c) secure financial transaction capabilities related to both electronic information and/or appliance usage and other electronic credit and/or currency usage and administration capabilities,

(d) privacy protection for usage information a user does not wish to release, and

(e) "living" electronic information content dissemination models that flexibly accommodate:

(1) a breadth of participants,

(2) one or more pathways (chains) for: the handling of content, content and/or appliance control information, reporting of content and/or appliance usage related information, and/or payment,

(3) supporting an evolution of terms and conditions incorporated into content control information, including use of electronic negotiation capabilities,

(4) support the combination of multiple pieces of content to form new content aggregations, and

(5) multiple concurrent models.

Secure Processing Units

An important part of VDE provided by the present invention is the core secure transaction control arrangement,

herein called an SPU (or SPUs), that typically must be present in each user's computer, other electronic appliance, or network. SPUs provide a trusted environment for generating decryption keys, encrypting and decrypting information, managing the secure communication of keys and other information between electronic appliances (i.e. between VDE installations and/or between plural VDE instances within a single VDE installation), securely accumulating and managing audit trail, reporting, and budget information in secure and/or non-secure non-volatile memory, maintaining a secure database of control information management instructions, and providing a secure environment for performing certain other control and administrative functions.

A hardware SPU (rather than a software emulation) within a VDE node is necessary if a highly trusted environment for performing certain VDE activities is required. Such a trusted environment may be created through the use of certain control software, one or more tamper resistant hardware modules such as a semiconductor or semiconductor chipset (including, for example, a tamper resistant hardware electronic appliance peripheral device), for use within, and/or operatively connected to, an electronic appliance. With the present invention, the trustedness of a hardware SPU can be enhanced by enclosing some or all of its hardware elements within tamper resistant packaging and/or by employing other tamper resisting techniques (e.g. microfusing and/or thin wire detection techniques). A trusted environment of the present invention implemented, in part, through the use of tamper resistant semiconductor design, contains control logic, such as a microprocessor, that securely executes VDE processes.

A VDE node's hardware SPU is a core component of a VDE secure subsystem and may employ some or all of an electronic appliance's primary control logic, such as a microcontroller, microcomputer or other CPU arrangement. This primary control logic may be otherwise employed for non VDE purposes such as the control of some or all of an electronic appliance's non-VDE functions. When operating in a hardware SPU mode, said primary control logic must be sufficiently secure so as to protect and conceal important VDE processes. For example, a hardware SPU may employ a host electronic appliance microcomputer operating in protected mode while performing VDE related activities, thus allowing portions of VDE processes to execute with a certain degree of security. This alternate embodiment is in contrast to the preferred embodiment wherein a trusted environment is created using a combination of one or more tamper resistant semiconductors that are not part of said primary control logic. In either embodiment, certain control information (software and parameter data) must be securely maintained within the SPU, and further control information can be stored externally and securely (e.g. in encrypted and tagged form) and loaded into said hardware SPU when needed. In many cases, and in particular with microcomputers, the preferred embodiment approach of employing special purpose secure hardware for executing said VDE processes, rather than using said primary control logic, may be more secure and efficient. The level of security and tamper resistance required for trusted SPU hardware processes depends on the commercial requirements of particular markets or market niches, and may vary widely.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features and advantages provided by the present invention(s) may be better and more completely understood by referring to the following detailed description of presently preferred example embodiments in connection with the drawings, of which:

FIG. **1** illustrates an example of a "Virtual Distribution Environment" provided in accordance with a preferred example/embodiment of this invention;

FIG. **1A** is a more detailed illustration of an example of the "Information Utility" shown in FIG. **1**;

FIG. **2** illustrates an example of a chain of handling and control;

FIG. **2A** illustrates one example of how rules and control information may persist from one participant to another in the FIG. **2** chain of handling and control;

FIG. **3** shows one example of different control information that may be provided;

FIG. **4** illustrates examples of some different types of rules and/or control information;

FIGS. **5A** and **5B** show an example of an "object";

FIG. **6** shows an example of a Secure Processing Unit ("SPU");

FIG. **7** shows an example of an electronic appliance;

FIG. **8** is a more detailed block diagram of an example of the electronic appliance shown in FIG. **7**;

FIG. **9** is a detailed view of an example of the Secure Processing Unit (SPU) shown in FIGS. **6** and **8**;

FIG. **10** shows an example of a "Rights Operating System" ("ROS") architecture provided by the Virtual Distribution Environment;

FIGS. **11A–11C** show examples of functional relationship (s) between applications and the Rights Operating System;

FIGS. **11D–11J** show examples of "components" and "component assemblies";

FIG. **12** is a more detailed diagram of an example of the Rights Operating System shown in FIG. **10**;

FIG. **12A** shows an example of how "objects" can be created;

FIG. **13** is a detailed block diagram of an example the software architecture for a "protected processing environment" shown in FIG. **12**;

FIGS. **14A–14C** are examples of SPU memory maps provided by the protected processing environment shown in FIG. **13**;

FIG. **15** illustrates an example of how the channel services manager and load module execution manager of FIG. **13** can support a channel;

FIG. **15A** is an example of a channel header and channel detail records shown in FIG. **15**;

FIG. **15B** is a flowchart of an example of program control steps that may be performed by the FIG. **13** protected processing environment to create a channel;

FIG. **16** is a block diagram of an example of a secure data base structure;

FIG. **17** is an illustration of an example of a logical object structure;

FIG. **18** shows an example of a stationary object structure;

FIG. **19** shows an example of a traveling object structure;

FIG. **20** shows an example of a content object structure;

FIG. **21** shows an example of an administrative object structure;

FIG. **22** shows an example of a method core structure;

FIG. **23** shows an example of a load module structure;

FIG. **24** shows an example of a User Data Element (UDE) and/or Method Data Element (MDE) structure;

FIGS. **25**A–**25**C show examples of "map meters";

FIG. **26** shows an example of a permissions record (PERC) structure;

FIGS. **26**A and **26**B together show a more detailed example of a permissions record structure;

FIG. **27** shows an example of a shipping table structure;

FIG. **28** shows an example of a receiving table structure;

FIG. **29** shows an example of an administrative event log structure;

FIG. **30** shows an example inter-relationship between and use of the object registration table, subject table and user rights table shown in the FIG. **16** secure database;

FIG. **31** is a more detailed example of an object registration table shown in FIG. **16**;

FIG. **32** is a more detailed example of subject table shown in FIG. **16**;

FIG. **33** is a more detailed example of a user rights table shown in FIG. **16**;

FIG. **34** shows a specific example of how a site record table and group record table may track portions of the secure database shown in FIG. **16**;

FIG. **34**A is an example of a FIG. **34** site record table structure;

FIG. **34**B is an example of a FIG. **34** group record table structure;

FIG. **35** shows an example of a process for updating the secure database;

FIG. **36** shows an example of how new elements may be inserted into the FIG. **16** secure data base;

FIG. **37** shows an example of how an element of the secure database may be accessed;

FIG. **38** is a flowchart example of how to protect a secure database element;

FIG. **39** is a flowchart example of how to back up a secure database;

FIG. **40** is a flowchart example of how to recover a secure database from a backup;

FIGS. **41**A–**41**D are a set of examples showing how a "chain of handling and control" may be enabled using "reciprocal methods";

FIGS. **42**A–**42**D show an example of a "reciprocal" BUDGET method;

FIGS. **43**A–**43**D show an example of a "reciprocal" REGISTER method;

FIGS. **44**A–**44**C show an example of a "reciprocal" AUDIT method;

FIGS. **45**–**48** show examples of several methods being used together to control release of content or other information;

FIGS. **49**, **49**A–**49**F show an example OPEN method;

FIGS. **50**, **50**A–**50**F show an example of a READ method;

FIGS. **51**, **51**A–**51**F show an example of a WRITE method;

FIG. **52** shows an example of a CLOSE method;

FIGS. **53**A–**53**B show an example of an EVENT method;

FIG. **53**C shows am example of a BILLING method;

FIG. **54** shows an example of an ACCESS method;

FIGS. **55**A–**55**B show examples of DECRYPT and ENCRYPT methods;

FIG. **56** shows an example of a CONTENT method;

FIGS. **57**A and **57**B show examples of EXTRACT and EMBED methods;

FIG. **58**A shows an example of an OBSCURE method;

FIGS. **58**B, **58**C show examples of a FINGERPRINT method;

FIG. **59** shows an example of a DESTROY method;

FIG. **60** shows an example of a PANIC method;

FIG. **61** shows an example of a METER method;

FIG. **62** shows an example of a key "convolution" process;

FIG. **63** shows an example of how different keys may be generated using a key convolution process to determine a "true" key;

FIGS. **64** and **65** show an example of how protected processing environment keys may be initialized;

FIGS. **66** and **67** show example processes for decrypting information contained within stationary and traveling objects, respectively;

FIG. **68** shows an example of how a protected processing environment may be initialized;

FIG. **69** shows an example of how firmware may be downloaded into a protected processing environment;

FIG. **70** shows an example of multiple VDE electronic appliances connected together with a network or other communications means;

FIG. **71** shows an example of a portable VDE electronic appliance;

FIGS. **72**A–**72**D show examples of "pop-up" displays that may be generated by the user notification and exception interface;

FIG. **73** shows an example of a "smart object";

FIG. **74** shows an example of a process using "smart objects";

FIGS. **75**A–**75**D show examples of data structures used for electronic negotiation;

FIGS. **75**E–**75**F show example structures relating to an electronic agreement;

FIGS. **76**A–**76**B show examples of electronic negotiation processes;

FIG. **77** shows a further example of a chain of handling and control;

FIG. **78** shows an example of a VDE "repository";

FIGS. **79**–**83** show an example illustrating a chain of handling and control to evolve and transform VDE managed content and control information;

FIG. **84** shows a further example of a chain of handling and control involving several categories of VDE participants;

FIG. **85** shows a further example of a chain of distribution and handling within an organization;

FIGS. **86** and **86**A show a further example of a chain of handling and control; and

FIG. **87** shows an example of a virtual silicon container model.

MORE DETAILED DESCRIPTION

FIGS. **1**–**7** and the discussion below provides an overview of some aspects of features provided by this invention. Following this overview is a more technical "detail description" of example embodiments in accordance with the invention.

Overview

FIG. **1** shows a "Virtual Distribution Environment" ("VDE") **100** that may be provided in accordance with this

invention. In FIG. **1**, an information utility **200** connects to communications means **202** such as telephone or cable TV lines for example. Telephone or cable TV lines **202** may be part of an "electronic highway" that carries electronic information from place to place. Lines **202** connect information utility **200** to other people such as for example a consumer **208**, an office **210**, a video production studio **204**, and a publishing house **214**. Each of the people connected to information utility **200** may be called a "VDE participant" because they can participate in transactions occurring within the virtual distribution environment **100**.

Almost any sort of transaction you can think of can be supported by virtual distribution environment **100**. A few of many examples of transactions that can be supported by virtual distribution environment **100** include:

- home banking and electronic payments;
- electronic legal contracts;
- distribution of "content" such as electronic printed matter, video, audio, images and computer programs; and
- secure communication of private information such as medical records and financial information.

Virtual distribution environment **100** is "virtual" because it does not require many of the physical "things" that used to be necessary to protect rights, ensure reliable and predictable distribution, and ensure proper compensation to content creators and distributors. For example, in the past, information was distributed on records or disks that were difficult to copy. In the past, private or secret content was distributed in sealed envelopes or locked briefcases delivered by courier. To ensure appropriate compensation, consumers received goods and services only after they handed cash over to a seller. Although information utility **200** may deliver information by transferring physical "things" such as electronic storage media, the virtual distribution environment **100** facilitates a completely electronic "chain of handling and control."

VDE Flexibility Supports Transactions

Information utility **200** flexibly supports many different kinds of information transactions. Different VDE participants may define and/or participate in different parts of a transaction. Information utility **200** may assist with delivering information about a transaction, or it may be one of the transaction participants.

For example, the video production studio **204** in the upper right-hand corner of FIG. **1** may create video/television programs. Video production studio **204** may send these programs over lines **202**, or may use other paths such as satellite link **205** and CD ROM delivery service **216**. Video production studio **204** can send the programs directly to consumers **206**, **206**, **210**, or it can send the programs to information utility **200** which may store and later send them to the consumers, for example. Consumers **206**, **208**, **210** are each capable of receiving and using the programs created by video production studio **204**—assuming, that is, that the video production studio or information utility **200** has arranged for these consumers to have appropriate "rules and controls" (control information) that give the consumers rights to use the programs.

Even if a consumer has a copy of a video program, she cannot watch or copy the program unless she has "rules and controls" that authorize use of the program. She can use the program only as permitted by the "rules and controls."

For example, video production studio **204** might release a half-hour exercise video in the hope that as many viewers as possible will view it. The video production studio **204** wishes to receive $2.00 per viewing. Video production studio **204** may, through information utility **200**, make the exercise video available in "protected" form to all consumers **206**, **208**, **210**. Video production studio **204** may also provide "rules and controls" for the video. These "rules and controls" may specify for example:

(1) any consumer who has good credit of at least $2.00 based on a credit account with independent financial provider **212** (such as Mastercard or VISA) may watch the video,

(2) virtual distribution environment **100** will "meter" each time a consumer watches the video, and report usage to video production studio **204** from time to time, and

(3) financial provider **212** may electronically collect payment ($2.00) from the credit account of each consumer who watches the video, and transfer these payments to the video production studio **204**.

Information utility **200** allows even a small video production studio to market videos to consumers and receive compensation for its efforts. Moreover, the videos can, with appropriate payment to the video production studio, be made available to other video publishers who may add value and/or act as repackagers or redistributors.

FIG. **1** also shows a publishing house **214**. Publishing house **214** may act as a distributor for an author **206**. The publishing house **214** may distribute rights to use "content" (such as computer software, electronic newspapers, the video produced by publishing house **214**, audio, or any other data) to consumers such as office **210**. The use rights may be defined by "rules and controls" distributed by publishing house **216**. Publishing house **216** may distribute these "rules and controls" with the content, but this is not necessary. Because the content can be used only by consumers that have the appropriate "rules and controls," content and its associated "rules and controls" may be distributed at different times, in different ways, by different VDE participants. The ability of VDE to securely distribute and enforce "rules and controls" separately from the content they apply to provides great advantages.

Use rights distributed by publishing house **214** may, for example, permit office **210** to make and distribute copies of the content to its employees. Office **210** may act as a redistributor by extending a "chain of handling and control" to its employees. The office **210** may add or modify "rules and controls" (consistent with the "rules and controls" it receives from publishing house **214**) to provide office-internal control information and mechanisms. For example, office **210** may set a maximum usage budget for each individual user and/or group within the office, or it may permit only specified employees and/or groups to access certain information.

FIG. **1** also shows an information delivery service **216** delivering electronic storage media such as "CD ROM" disks to consumers **206**. Even though the electronic storage media themselves are not delivered electronically by information utility **200** over lines **202**, they are still part of the virtual distribution environment **100**. The electronic storage media may be used to distribute content, "rules and controls," or other information.

Example of What's Inside Information Utility **200**

"Information utility" **200** in FIG. **1** can be a collection of participants that may act as distributors, financial clearinghouses, and administrators. FIG. **1A** shows an example of what may be inside one example of information utility **200**. Information utility participants **200***a*–**200***g* could each be an independent organization/business. There can be any number of each of participants **200***a*–**200***g*. In this example, electronic "switch" **200***a* connects internal parts of

information utility **200** to each other and to outside participants, and may also connect outside participants to one another.

Information utility **200** may include a "transaction processor" **200***b* that processes transactions (to transfer electronic funds, for example) based on requests from participants and/or report receiver **200***e*. It may also include a "usage analyst" **200***c* that analyzes reported usage information. A "report creator" **200***d* may create reports based on usage for example, and may provide these reports to outside participants and/or to participants within information utility **200**. A "report receiver" **200***e* may receive reports such as usage reports from content users. A "permissioning agent" **200***f* may distribute "rules and controls" granting usage or distribution permissions based on a profile of a consumer's credit worthiness, for example. An administrator **200***h* may provide information that keeps the virtual distribution environment **100** operating properly. A content and message storage **200***g* may store information for use by participants within or outside of information utility **200**.

Example of Distributing "Content" Using a "Chain of Handling and Control"

As explained above, virtual distribution environment **100** can be used to manage almost any sort of transaction. One type of important transaction that virtual distribution environment **100** may be used to manage is the distribution or communication of "content" or other important information. FIG. **2** more abstractly shows a "model" of how the FIG. **1** virtual distribution environment **100** may be used to provide a "chain of handling and control" for distributing content. Each of the blocks in FIG. **2** may correspond to one or more of the VDE participants shown in FIG. **1**.

In the FIG. **2** example, a VDE content creator **102** creates "content." The content creator **102** may also specify "rules and controls" for distributing the content. These distribution-related "rules and controls" can specify who has permission to distribute the rights to use content, and how many users are allowed to use the content.

Arrow **104** shows the content creator **102** sending the "rules and controls" associated with the content to a VDE rights distributor **106** ("distributor") over an electronic highway **108** (or by some other path such as an optical disk sent by a delivery service such as U.S. mail). The content can be distributed over the same or different path used to send the "rules and controls." The distributor **106** generates her own "rules and controls" that relate to usage of the content. The usage-related "rules and controls" may, for example, specify what a user can and can't do with the content and how much it costs to use the content. These usage-related "rules and controls" must be consistent with the "rules and controls" specified by content creator **102**.

Arrow **110** shows the distributor **106** distributing rights to use the content by sending the content's "rules and controls" to a content user **112** such as a consumer. The content user **112** uses the content in accordance with the usage-related "rules and controls."

In this FIG. **2** example, information relating to content use is, as shown by arrow **114**, reported to a financial clearinghouse **116**. Based on this "reporting," the financial clearinghouse **116** may generate a bill and send it to the content user **112** over a "reports and payments" network **118**. Arrow **120** shows the content user **112** providing payments for content usage to the financial clearinghouse **116**. Based on the reports and payments it receives, the financial clearinghouse **116** may provide reports and/or payments to the distributor **106**. The distributor **106** may, as shown by arrow **122**, provide reports and/or payments to the content creator **102**. The clearinghouse **116** may provide reports and payments directly to the creator **102**. Reporting and/or payments may be done differently. For example, clearinghouse **116** may directly or through an agent, provide reports and/or payments to each of VDE content creators **102**, and rights distributor **106**, as well as reports to content user **112**.

The distributor **106** and the content creator **102** may be the same person, or they may be different people. For example, a musical performing group may act as both content creator **102** and distributor **106** by creating and distributing its own musical recordings. As another example, a publishing house may act as a distributor **106** to distribute rights to use works created by an author content creator **102**. Content creators **102** may use a distributor **106** to efficiently manage the financial end of content distribution.

The "financial clearinghouse" **116** shown in FIG. **2** may also be a "VDE administrator." Financial clearinghouse **116** in its VDE administrator role sends "administrative" information to the VDE participants. This administrative information helps to keep the virtual distribution environment **100** operating properly. The "VDE administrator" and financial clearinghouse roles may be performed by different people or companies, and there can be more than one of each.

More about "Rules and Controls"

The virtual distribution environment **100** prevents use of protected information except as permitted by the "rules and controls" (control information). For example, the "rules and controls" shown in FIG. **2** may grant specific individuals or classes of content users **112** "permission" to use certain content. They may specify what kinds of content usage are permitted, and what kinds are not. They may specify how content usage is to be paid for and how much it costs. As another example, "rules and controls" may require content usage information to be reported back to the distributor **106** and/or content creator **102**.

Every VDE participant in "chain of handling and control" is normally subject to "rules and controls." "Rules and controls" define the respective rights and obligations of each of the various VDE participants. "Rules and controls" provide information and mechanisms that may establish interdependencies and relationships between the participants. "Rules and controls" are flexible, and permit "virtual distribution environment" **100** to support most "traditional" business transactions. For example:

- "Rules and controls" may specify which financial clearinghouse(s) **116** may process payments,
- "Rules and controls" may specify which participant(s) receive what kind of usage report, and
- "Rules and controls" may specify that certain information is revealed to certain participants, and that other information is kept secret from them.

"Rules and controls" may self limit if and how they may be changed. Often, "rules and controls" specified by one VDE participant cannot be changed by another VDE participant. For example, a content user **112** generally can't change "rules and controls" specified by a distributor **106** that require the user to pay for content usage at a certain rate. "Rules and controls" may "persist" as they pass through a "chain of handling and control," and may be "inherited" as they are passed down from one VDE participant to the next.

Depending upon their needs, VDE participants can specify that their "rules and controls" can be changed under conditions specified by the same or other "rules and controls." For example, "rules and controls" specified by the content creator **102** may permit the distributor **106** to "mark up" the usage price just as retail stores "mark up" the

wholesale price of goods. FIG. **2**A shows an example in which certain "rules and controls" persist unchanged from content creator **102** to content user **112**; other "rules and controls" are modified or deleted by distributor **106**; and still other "rules and controls" are added by the distributor.

"Rules and controls" can be used to protect the content user's privacy by limiting the information that is reported to other VDE participants. As one example, "rules and controls" can cause content usage information to be reported anonymously without revealing content user identity, or it can reveal only certain information to certain participants (for example, information derived from usage) with appropriate permission, if required. This ability to securely control what information is revealed and what VDE participants) it is revealed to allows the privacy rights of all VDE participants to be protected.

"Rules and Contents" Can Be Separately Delivered

As mentioned above, virtual distribution environment **100** "associates" content with corresponding "rules and controls," and prevents the content from being used or accessed unless a set of corresponding "rules and controls" is available. The distributor **106** doesn't need to deliver content to control the content's distribution. The preferred embodiment can securely protect content by protecting corresponding, usage enabling "rules and controls" against unauthorized distribution and use.

In some examples, "rules and controls" may travel with the content they apply to. Virtual distribution environment **100** also allows "rules and controls" to be delivered separately from content. Since no one can use or access protected content without "permission" from corresponding "rules and controls," the distributor **106** can control use of content that has already been (or will in the future be) delivered. "Rules and controls" may be delivered over a path different from the one used for content delivery. "Rules and controls" may also be delivered at some other time. The content creator **102** might deliver content to content user **112** over the electronic highway **108**, or could make the content available to anyone on the highway. Content may be used at the time it is delivered, or it may be stored for later use or reuse.

The virtual distribution environment **100** also allows payment and reporting means to be delivered separately. For example, the content user **112** may have a virtual "credit card" that extends credit (up to a certain limit) to pay for usage of any content. A "credit transaction" can take place at the user's site without requiring any "online" connection or further authorization. This invention can be used to help securely protect the virtual "credit card" against unauthorized use.

"Rules and Contents" Define Processes

FIG. **3** shows an example of an overall process based on "rules and controls." It includes an "events" process **402**, a meter process **404**, a billing process **406**, and a budget process **408**. Not all of the processes shown in FIG. **3** will be used for every set of "rules and controls."

The "events process" **402** detects things that happen ("events") and determines which of those "events" need action by the other "processes." The "events" may include, for example, a request to use content or generate a usage permission. Some events may need additional processing, and others may not. Whether an "event" needs more processing depends on the "rules and controls" corresponding to the content. For example, a user who lacks permission will not have her request satisfied ("No Go"). As another example, each user request to turn to a new page of an electronic book may be satisfied ("Go"), but it may not be necessary to meter, bill or budget those requests. A user who has purchased a copy of a novel may be permitted to open and read the novel as many times as she wants to without any further metering, billing or budgeting. In this simple example, the "event process" **402** may request metering, billing and/or budgeting processes the first time the user asks to open the protected novel (so the purchase price can be charged to the user), and treat all later requests to open the same novel as "insignificant events." Other content (for example, searching an electronic telephone directory) may require the user to pay a fee for each access.

"Meter" process **404** keeps track of events, and may report usage to distributor **106** and/or other appropriate VDE participant(s). FIG. **4** shows that process **404** can be based on a number of different factors such as:

(a) type of usage to charge for,

(b) what kind of unit to base charges on,

(c) how much to charge per unit,

(d) when to report, and

(e) how to pay.

These factors may be specified by the "rules and controls" that control the meter process.

Billing process **406** determines how much to charge for events. It records and reports payment information.

Budget process **408** limits how much content usage is permitted. For example, budget process **408** may limit the number of times content may be accessed or copied, or it may limit the number of pages or other amount of content that can be used based on, for example, the number of dollars available in a credit account. Budget process **408** records and reports financial and other transaction information associated with such limits.

Content may be supplied to the user once these processes have been successfully performed.

Containers and "Objects"

FIG. **5**A shows how the virtual distribution environment **100**, in a preferred embodiment, may package information elements (content) into a "container" **302** so the information can't be accessed except as provided by its "rules and controls." Normally, the container **302** is electronic rather than physical. Electronic container **302** in one example comprises "digital" information having a well defined structure. Container **302** and its contents can be called an "object **300**."

The FIG. **5**A example shows items "within" and enclosed by container **302**. However, container **302** may "contain" items without those items actually being stored within the container. For example, the container **302** may reference items that are available elsewhere such as in other containers at remote sites. Container **302** may reference items available at different times or only during limited times. Some items may be too large to store within container **302**. Items may, for example, be delivered to the user in the form of a "live feed" of video at a certain time. Even then, the container **302** "contains" the live feed (by reference) in this example.

Container **302** may contain information content **304** in electronic (such as "digital") form. Information content **304** could be the text of a novel, a picture, sound such as a musical performance or a reading, a movie or other video, computer software, or just about any other kind of electronic information you can think of. Other types of "objects" **300** (such as "administrative objects") may contain "administrative" or other information instead of or in addition to information content **304**.

In the FIG. **5**A example, container **302** may also contain "rules and controls" in the form of:

(a) a "permissions record" **808**;

(b) "budgets" **308**; and

(c) "other methods" **1000**.

FIG. **5**B gives some additional detail about permissions record **808**, budgets **308** and other methods **1000**. The "permissions record" **808** specifies the rights associated with the object **300** such as, for example, who can open the container **302**, who can use the object's contents, who can distribute the object, and what other control mechanisms must be active. For example, permissions record **808** may specify a user's rights to use, distribute and/or administer the container **302** and its content. Permissions record **808** may also specify requirements to be applied by the budgets **308** and "other methods" **1000**. Permissions record **808** may also contain security related information such as scrambling and descrambling "keys."

"Budgets" **308** shown in FIG. **5**B are a special type of "method" **1000** that may specify, among other things, limitations on usage of information content **304**, and how usage will be paid for. Budgets **308** can specify, for example, how much of the total information content **304** can be used and/or copied. The methods **310** may prevent use of more than the amount specified by a specific budget.

"Other methods" **1000** define basic operations used by "rules and controls." Such "methods" **1000** may include, for example, how usage is to be "metered," if and how content **304** and other information is to be scrambled and descrambled, and other processes associated with handling and controlling information content **304**. For example, methods **1000** may record the identity of anyone who opens the electronic container **302**, and can also control how information content is to be charged based on "metering." Methods **1000** may apply to one or several different information contents **304** and associated containers **302**, as well as to all or specific portions of information content **304**.

Secure Processing Unit (SPU)

The "VDE participants" may each have an "electronic appliance." The appliance may be or contain a computer. The appliances may communicate over the electronic highway **108**. FIG. **6** shows a secure processing unit ("SPU") **500** portion of the "electronic appliance" used in this example by each VDE participant. SPU **500** processes information in a secure processing environment **503**, and stores important information securely. SPU **500** may be emulated by software operating in a host electronic appliance.

SPU **500** is enclosed within and protected by a "tamper resistant security barrier" **502**. Security barrier **502** separates the secure environment **503** from the rest of the world. It prevents information and processes within the secure environment **503** from being observed, interfered with and leaving except under appropriate secure conditions. Barrier **502** also controls external access to secure resources, processes and information within SPU **500**. In one example, tamper resistant security barrier **502** is formed by security features such as "encryption," and hardware that detects tampering and/or destroys sensitive information within secure environment **503** when tampering is detected.

SPU **500** in this example is an integrated circuit ("IC") "chip" **504** including "hardware" **506** and "firmware" **508**. SPU **500** connects to the rest of the electronic appliance through an "appliance link" **510**. SPU "firmware" **508** in this example is "software" such as a "computer program(s)" "embedded" within chip **504**. Firmware **508** makes the hardware **506** work. Hardware **506** preferably contains a processor to perform instructions specified by firmware **508**. "Hardware" **506** also contains long-term and short-term memories to store information securely so it can't be tampered with. SPU **500** may also have a protected clock/calendar used for timing events. The SPU hardware **506** in this example may include special purpose electronic circuits that are specially designed to perform certain processes (such as "encryption" and "decryption") rapidly and efficiently.

The particular context in which SPU **500** is being used will determine how much processing capabilities SPU **500** should have. SPU hardware **506**, in this example, provides at least enough processing capabilities to support the secure parts of processes shown in FIG. **3**. In some contexts, the functions of SPU **500** may be increased so the SPU can perform all the electronic appliance processing, and can be incorporated into a general purpose processor. In other contexts, SPU **500** may work alongside a general purpose processor, and therefore only needs to have enough processing capabilities to handle secure processes.

VDE Electronic Appliance and "rights Operating System"

FIG. **7** shows an example of an electronic appliance **600** including SPU **500**. Electronic appliance **600** may be practically any kind of electrical or electronic device, such as:

a computer

a T.V. "set top" control box

a pager

a telephone

a sound system

a video reproduction system

a video game player

a "smart" credit card

Electronic appliance **600** in this example may include a keyboard or keypad **612**, a voice recognizer **613**, and a display **614**. A human user can input commands through keyboard **612** and/or voice recognizer **613**, and may view information on display **614**. Appliance **600** may communicate with the outside world through any of the connections/devices normally used within an electronic appliance. The connections/devices shown along the bottom of the drawing are examples:

a "modem" **618** or other telecommunications link;

a CD ROM disk **620** or other storage medium or device;

a printer **622**;

broadcast reception **624**;

a document scanner **626**; and

a "cable" **628** connecting the appliance with a "network."

Virtual distribution environment **100** provides a "rights operating system" **602** that manages appliance **600** and SPU **500** by controlling their hardware resources. The operating system **602** may also support at least one "application" **608**. Generally, "application" **608** is hardware and/or software specific to the context of appliance **600**. For example, if appliance **600** is a personal computer, then "application" **608** could be a program loaded by the user, for instance, a word processor, a communications system or a sound recorder. If appliance **600** is a television controller box, then application **608** might be hardware or software that allows a user to order videos on demand and perform other functions such as fast forward and rewind. In this example, operating system **602** provides a standardized, well defined, generalized "interface" that could support and work with many different "applications" **608**.

Operating system **602** in this example provides "rights and auditing operating system functions" **604** and "other operating system functions" **606**. The "rights and auditing operating system functions" **604** securely handle tasks that relate to virtual distribution environment **100**. SPU **500**

provides or supports many of the security functions of the "rights and auditing operating system functions" **402**. The "other operating system functions" **606** handle general appliance functions. Overall operating system **602** may be designed from the beginning to include the "rights and auditing operating system functions" **604** plus the "other operating system functions" **606**, or the "rights and auditing operating system functions" may be an add-on to a preexisting operating system providing the "other operating system functions."

"Rights operating system" **602** in this example can work with many different types of appliances **600**. For example, it can work with large mainframe computers, "minicomputers" and "microcomputers" such as personal computers and portable computing devices. It can also work in control boxes on the top of television sets, small portable "pagers," desktop radios, stereo sound systems, telephones, telephone switches, or any other electronic appliance. This ability to work on big appliances as well as little appliances is called "scalable." A "scalable" operating system **602** means that there can be a standardized interface across many different appliances performing a wide variety of tasks.

The "rights operating system functions" **604** are "services-based" in this example. For example, "rights operating system functions" **604** handle summary requests from application **608** rather than requiring the application to always make more detailed "subrequests" or otherwise get involved with the underlying complexities involved in satisfying a summary request. For example, application **608** may simply ask to read specified information; "rights operating system functions" **604** can then decide whether the desired information is VDE-protected content and, if it is, perform processes needed to make the information available. This feature is called "transparency." "Transparency" makes tasks easy for the application **608**. "Rights operating system functions" **604** can support applications **608** that "know" nothing about virtual distribution environment **100**. Applications **608** that are "aware" of virtual distribution environment **100** may be able to make more detailed use of virtual distribution environment **100**.

In this example, "rights operating system functions" **604** are "event driven". Rather than repeatedly examining the state of electronic appliance **600** to determine whether a condition has arisen, the "rights operating system functions" **604** may respond directly to "events" or "happenings" within appliance **600**.

In this example, some of the services performed by "rights operating system functions" **604** may be extended based on additional "components" delivered to operating system **602**. "Rights operating system functions" **604** can collect together and use "components" sent by different participants at different times. The "components" help to make the operating system **602** "scalable." Some components can change how services work on little appliances versus how they work on big appliances (e.g., multi-user). Other components are designed to work with specific applications or classes of applications (e.g., some types of meters and some types of budgets).

Electronic Appliance **600**

An electronic appliance **600** provided by the preferred embodiment may, for example, be any electronic apparatus that contains one or more microprocessors and/or microcontrollers and/or other devices which perform logical and/or mathematical calculations. This may include computers; computer terminals; device controllers for use with computers; peripheral devices for use with computers; digital display devices; televisions; video and audio/video projection systems; channel selectors and/or decoders for use with broadcast and/or cable transmissions; remote control devices; video and/or audio recorders; media players including compact disc players, videodisc players and tape players; audio and/or video amplifiers; virtual reality machines; electronic game players; multimedia players; radios; telephones; videophones; facsimile machines; robots; numerically controlled machines including machine tools and the like; and other devices containing one or more microcomputers and/or microcontrollers and/or other CPUs, including those not yet in existence.

FIG. **8** shows an example of an electronic appliance **600**. This example of electronic appliance **600** includes a system bus **653**. In this example, one or more conventional general purpose central processing units ("CPUs") **654** are connected to bus **653**. Bus **653** connects CPU(s) **654** to RAM **656**, ROM **658**, and I/O controller **660**. One or more SPUs **500** may also be connected to system bus **653**. System bus **653** may permit SPU(s) **500** to communicate with CPU(s) **654**, and also may allow both the CPU(s) and the SPU(s) to communicate (e.g., over shared address and data lines) with RAM **656**, ROM **658** and I/O controller **660**. A power supply **659** may provide power to SPU **500**, CPU **654** and the other system components shown.

In the example shown, I/O controller **660** is connected to secondary storage device **652**, a keyboard/display **612**, **614**, a communications controller **666**, and a backup storage device **668**. Backup storage device **668** may, for example, store information on mass media such as a tape **670**, a floppy disk, a removable memory card, etc. Communications controller **666** may allow electronic appliance **600** to communicate with other electronic appliances via network **672** or other telecommunications links. Different electronic appliances **600** may interoperate even if they use different CPUs and different instances of ROS **602**, so long as they typically use compatible communication protocols and/or security methods. In this example, I/O controller **660** permits CPU **654** and SPU **500** to read from and write to secondary storage **662**, keyboard/display **612**, **614**, communications controller **666**, and backup storage device **668**.

Secondary storage **662** may comprise the same one or more non-secure secondary storage devices (such as a magnetic disk and a CD-ROM drive as one example) that electronic appliance **600** uses for general secondary storage functions. In some implementations, part or all of secondary storage **652** may comprise a secondary storage device(s) that is physically enclosed within a secure enclosure. However, since it may not be practical or cost-effective to physically secure secondary storage **652** in many implementations, secondary storage **652** may be used to store information in a secure manner by encrypting information before storing it in secondary storage **652**. If information is encrypted before it is stored, physical access to secondary storage **652** or its contents does not readily reveal or compromise the information.

Secondary storage **652** in this example stores code and data used by CPU **654** and/or SPU **500** to control the overall operation of electronic appliance **600**. For example, FIG. **8** shows that "Rights Operating System" ("ROS") **602** (including a portion **604** of ROS that provides VDE functions and a portion **606** that provides other OS functions) shown in FIG. **7** may be stored on secondary storage **652**. Secondary storage **652** may also store one or more VDE objects **300**. FIG. **8** also shows that the secure files **610** shown in FIG. **7** may be stored on secondary storage **652** in the form of a "secure database" or management file system **610**. This secure database **610** may store and organize

information used by ROS **602** to perform VDE functions **604**. Thus, the code that is executed to perform VDE and other OS functions **604**, **606**, and secure files **610** (as well as VDE objects **300**) associated with those functions may be stored in secondary storage **652**. Secondary storage **652** may also store "other information" **673** such as, for example, information used by other operating system functions **606** for task management, non-VDE files, etc. Portions of the elements indicated in secondary storage **652** may also be stored in ROM **658**, so long as those elements do not require changes (except when ROM **658** is replaced). Portions of ROS **602** in particular may desirably be included in ROM **658** (e.g., "bootstrap" routines, POST routines, etc. for use in establishing an operating environment for electronic appliance **600** when power is applied).

FIG. **8** shows that secondary storage **652** may also be used to store code ("application programs") providing user application(s) **608** shown in FIG. **7**. FIG. **8** shows that there may be two general types of application programs **608**: "VDE aware" applications **608***a*, and Non-VDE aware applications **608***b*. VDE aware applications **608***a* may have been at least in part designed specifically with VDE **100** in mind to access and take detailed advantage of VDE functions **604**. Because of the "transparency" features of ROS **602**, non-VDE aware applications **608***b* (e.g., applications not specifically designed for VDE **100**) can also access and take advantage of VDE functions **604**.

Secure Processing Unit **500**

Each VDE node or other electronic appliance **600** in the preferred embodiment may include one or more SPUs **500**. SPUs **500** may be used to perform all secure processing for VDE **100**. For example, SPU **500** is used for decrypting (or otherwise unsecuring) VDE protected objects **300**. It is also used for managing encrypted and/or otherwise secured communication (such as by employing authentication and/or error-correction validation of information). SPU **500** may also perform secure data management processes including governing usage of, auditing of, and where appropriate, payment for VDE objects **300** (through the use of prepayments, credits, real-time electronic debits from bank accounts and/or VDE node currency token deposit accounts). SPU **500** may perform other transactions related to such VDE objects **300**.

SPU Physical Packaging and Security Barrier **502**

As shown FIG. **6**, in the preferred embodiment, an SPU **500** may be implemented as a single integrated circuit "chip" **505** to provide a secure processing environment in which confidential and/or commercially valuable information can be safely processed, encrypted and/or decrypted. IC chip **505** may, for example, comprise a small semiconductor "die" about the size of a thumbnail. This semiconductor die may include semiconductor and metal conductive pathways. These pathways define the circuitry, and thus the functionality, of SPU **500**. Some of these pathways are electrically connected to the external "pins" **504** of the chip **505**.

As shown in FIGS. **6** and **9**, SPU **500** may be surrounded by a tamper-resistant hardware security barrier **502**. Part of this security barrier **502** is formed by a plastic or other package in which an SPU "die" is encased. Because the processing occurring within, and information stored by, SPU **500** are not easily accessible to the outside world, they are relatively secure from unauthorized access and tampering. All signals cross barrier **502** through a secure, controlled path provided by BIU **530** that restricts the outside world's access to the internal components within SPU **500**. This secure, controlled path resists attempts from the outside world to access secret information and resources within SPU **500**.

It is possible to remove the plastic package of an IC chip and gain access to the "die." It is also possible to analyze and "reverse engineer" the "die" itself (e.g., using various types of logic analyzers and microprobes to collect and analyze signals on the die while the circuitry is operating, using acid etching or other techniques to remove semiconductor layers to expose other layers, viewing and photographing the die using an electron microscope, etc.) Although no system or circuit is absolutely impervious to such attacks, SPU barrier **502** may include additional hardware protections that make successful attacks exceedingly costly and time consuming. For example, ion implantation and/or other fabrication techniques may be used to make it very difficult to visually discern SPU die conductive pathways, and SPU internal circuitry may be fabricated in such a way that it "self-destructs" when exposed to air and/or light. SPU **500** may store secret information in internal memory that loses its contents when power is lost. Circuitry may be incorporated within SPU **500** that detects microprobing or other tampering, and self-destructs (or destroys other parts of the SPU) when tampering is detected. These and other hardware-based physical security techniques contribute to tamper-resistant hardware security barrier **502**.

To increase the security of security barrier **502** even further, it is possible to encase or include SPU **500** in one or more further physical enclosures such as, for example: epoxy or other "potting compound"; further module enclosures including additional self-destruct, self-disabling or other features activated when tampering is detected; further modules providing additional security protections such as requiring password or other authentication to operate; and the like. In addition, further layers of metal may be added to the die to complicate acid etching, micro probing, and the like; circuitry designed to "zeroize" memory may be included as an aspect of self-destruct processes; the plastic package itself may be designed to resist chemical as well as physical "attacks"; and memories internal to SPU **500** may have specialized addressing and refresh circuitry that "shuffles" the location of bits to complicate efforts to electrically determine the value of memory locations. These and other techniques may contribute to the security of barrier **502**.

In some electronic appliances **600**, SPU **500** may be integrated together with the device microcontroller or equivalent or with a device I/O or communications microcontroller into a common chip (or chip set) **505**. For example, in one preferred embodiment, SPU **500** may be integrated together with one or more other CPU(s) (e.g., a CPU **654** of an electronic appliance) in a single component or package. The other CPU(s) **654** may be any centrally controlling logic arrangement, such as for example, a microprocessor, other microcontroller, and/or array or other parallel processor. This integrated configuration may result in lower overall cost, smaller overall size, and potentially faster interaction between an SPU **500** and a CPU **654**. Integration may also provide wider distribution if an integrated SPU/CPU component is a standard feature of a widely distributed microprocessor line. Merging an SPU **500** into a main CPU **654** of an electronic appliance **600** (or into another appliance or appliance peripheral microcomputer or other microcontroller) may substantially reduce the overhead cost of implementing VDE **100**. Integration considerations may include cost of implementation, cost of manufacture, desired degree of security, and value of compactness.

SPU **500** may also be integrated with devices other than CPUs. For example, for video and multimedia applications,

some performance and/or security advantages (depending on overall design) could result from integrating an SPU **500** into a video controller chip or chipset. SPU **500** can also be integrated directly into a network communications chip or chipset or the like. Certain performance advantages in high speed communications applications may also result from integrating an SPU **500** with a modem chip or chipset. This may facilitate incorporation of an SPU **500** into communication appliances such as stand-alone fax machines. SPU **500** may also be integrated into other peripheral devices, such as CD-ROM devices, set-top cable devices, game devices, and a wide variety of other electronic appliances that use, allow access to, perform transactions related to, or consume, distributed information.

SPU **500** Internal Architecture

FIG. **9** is a detailed diagram of the internal structure within an example of SPU **500**. SPU **500** in this example includes a single microprocessor **520** and a limited amount of memory configured as ROM **532** and RAM **534**. In more detail, this example of SPU **500** includes microprocessor **520**, an encrypt/decrypt engine **522**, a DMA controller **526**, a real-time clock **528**, a bus interface unit ("BIU") **530**, a read only memory (ROM) **532**, a random access memory (RAM) **534**, and a memory management unit ("MMU") **540**. DMA controller **526** and MMU **540** are optional, but the performance of SPU **500** may suffer if they are not present. SPU **500** may also include an optional pattern matching engine **524**, an optional random number generator **542**, an optional arithmetic accelerator circuit **544**, and optional compression/decompression circuit **546**. A shared address/data bus arrangement **536** may transfer information between these various components under control of microprocessor **520** and/or DMA controller **526**. Additional or alternate dedicated paths **538** may connect microprocessor **520** to the other components (e.g., encrypt/decrypt engine **522** via line **538***a*, real-time clock **528** via line **538***b*, bus interface unit **530** via line **538***c*, DMA controller via line **538***d*, and memory management unit (MMU) **540** via line **538***e*).

The following section discusses each of these SPU components in more detail.

Microprocessor **520**

Microprocessor **520** is the "brain" of SPU **500**. In this example, it executes a sequence of steps specified by code stored (at least temporarily) within ROM **532** and/or RAM **534**. Microprocessor **520** in the preferred embodiment comprises a dedicated central processing arrangement (e.g., a RISC and/or CISC processor unit, a microcontroller, and/or other central processing means or, less desirably in most applications, process specific dedicated control logic) for executing instructions stored in the ROM **532** and/or other memory. Microprocessor **520** may be separate elements of a circuitry layout, or may be separate packages within a secure SPU **500**.

In the preferred embodiment, microprocessor **520** normally handles the most security sensitive aspects of the operation of electronic appliance **600**. For example, microprocessor **520** may manage VDE decrypting, encrypting, certain content and/or appliance usage control information, keeping track of usage of VDE secured content, and other VDE usage control related functions.

Stored in each SPU **500** and/or electronic appliance secondary memory **652** may be, for example, an instance of ROS **602** software, application programs **608**, objects **300** containing VDE controlled property content and related information, and management database **610** that stores both information associated with objects and VDE control information. ROS **602** includes software intended for execution by SPU microprocessor **520** for, in part, controlling usage of VDE related objects **300** by electronic appliance **600**. As will be explained, these SPU programs include "load modules" for performing basic control functions. These various programs and associated data are executed and manipulated primarily by microprocessor **520**.

Real Time Clock (RTC) **528**

In the preferred embodiment, SPU **500** includes a real time clock circuit ("RTC") **528** that serves as a reliable, tamper resistant time base for the SPU. RTC **528** keeps track of time of day and date (e.g., month, day and year) in the preferred embodiment, and thus may comprise a combination calendar and clock. A reliable time base is important for implementing time based usage metering methods, "time aged decryption keys," and other time based SPU functions.

The RTC **528** must receive power in order to operate. Optimally, the RTC **528** power source could comprise a small battery located within SPU **500** or other secure enclosure. However, the RTC **528** may employ a power source such as an externally located battery that is external to the SPU **500**. Such an externally located battery may provide relatively uninterrupted power to RTC **528**, and may also maintain as non-volatile at least a portion of the otherwise volatile RAM **534** within SPU **500**.

In one implementation, electronic appliance power supply **659** is also used to power SPU **500**. Using any external power supply as the only power source for RTC **528** may significantly reduce the usefulness of time based security techniques unless, at minimum, SPU **500** recognizes any interruption (or any material interruption) of the supply of external power, records such interruption, and responds as may be appropriate such as disabling the ability of the SPU **500** to perform certain or all VDE processes. Recognizing a power interruption may, for example, be accomplished by employing a circuit which is activated by power failure. The power failure sensing circuit may power another circuit that includes associated logic for recording one or more power fail events. Capacitor discharge circuitry may provide the necessary temporary power to operate this logic. In addition or alternatively, SPU **500** may from time to time compare an output of RTC **528** to a clock output of a host electronic appliance **600**, if available. In the event a discrepancy is detected, SPU **500** may respond as appropriate, including recording the discrepancy and/or disabling at least some portion of processes performed by SPU **500** under at least some circumstances.

If a power failure and/or RTC **528** discrepancy and/or other event indicates the possibility of tampering, SPU **500** may automatically destroy, or render inaccessible without privileged intervention, one or more portions of sensitive information it stores, such as execution related information and/or encryption key related information. To provide further SPU operation, such destroyed information would have to be replaced by a VDE clearinghouse, administrator and/or distributor, as may be appropriate. This may be achieved by remotely downloading update and/or replacement data and/or code. In the event of a disabling and/or destruction of processes and/or information as described above, the electronic appliance **600** may require a secure VDE communication with an administrator, clearinghouse, and/or distributor as appropriate in order to reinitialize the RTC **528**. Some or all secure SPU **500** processes may not operate until then.

It may be desirable to provide a mechanism for setting and/or synchronizing RTC **528**. In the preferred embodiment, when communication occurs between VDE electronic appliance **600** and another VDE appliance, an output of RTC **528** may be compared to a controlled RTC

528 output time under control of the party authorized to be "senior" and controlling. In the event of a discrepancy, appropriate action may be taken, including resetting the RTC 528 of the "junior" controlled participant in the communication.

SPU Encrypt/Decrypt Engine 522

In the preferred embodiment, SPU encrypt/decrypt engine 522 provides special purpose hardware (e.g., a hardware state machine) for rapidly and efficiently encrypting and/or decrypting data. In some implementations, the encrypt/decrypt functions may be performed instead by microprocessor 520 under software control, but providing special purpose encrypt/decrypt hardware engine 522 will, in general, provide increased performance. Microprocessor 520 may, if desired, comprise a combination of processor circuitry and dedicated encryption/decryption logic that may be integrated together in the same circuitry layout so as to, for example, optimally share one or more circuit elements.

Generally, it is preferable that a computationally efficient but highly secure "bulk" encryption/decryption technique should be used to protect most of the data and objects handled by SPU 500. It is preferable that an extremely secure encryption/decryption technique be used as an aspect of authenticating the identity of electronic appliances 600 that are establishing a communication channel and securing any transferred permission, method, and administrative information. In the preferred embodiment, the encrypt/decrypt engine 522 includes both a symmetric key encryption/decryption circuit (e.g. DES, Skipjack/Clipper, IDEA, RC-2, RC-4, etc.) and an antisymmetric (asymmetric) or Public Key ("PK") encryption/decryption circuit. The public/private key encryption/decryption circuit is used principally as an aspect of secure communications between an SPU 500 and VDE administrators, or other electronic appliances 600, that is between VDE secure subsystems. A symmetric encryption/decryption circuit may be used for "bulk" encrypting and decrypting most data stored in secondary storage 662 of electronic appliance 600 in which SPU 500 resides. The symmetric key encryption/decryption circuit may also be used for encrypting and decrypting content stored within VDE objects 300.

DES or public/private key methods may be used for all encryption functions. In alternate embodiments, encryption and decryption methods other than the DES and public/private key methods could be used for the various encryption related functions. For instance, other types of symmetric encryption/decryption techniques in which the same key is used for encryption and decryption could be used in place of DES encryption and decryption. The preferred embodiment can support a plurality of decryption/encryption techniques using multiple dedicated circuits within encrypt/decrypt engine 522 and/or the processing arrangement within SPU 500.

Pattern Matching Engine 524

Optional pattern matching engine 524 may provide special purpose hardware for performing pattern matching functions. One of the functions SPU 500 may perform is to validate/authenticate VDE objects 300 and other items. Validation/authentication often involves comparing long data strings to determine whether they compare in a predetermined way. In addition, certain forms of usage (such as logical and/or physical (contiguous) relatedness of accessed elements) may require searching potentially long strings of data for certain bit patterns or other significant pattern related metrics. Although pattern matching can be performed by SPU microprocessor 520 under software control, providing special purpose hardware pattern matching engine 524 may speed up the pattern matching process.

Compression/Decompression Engine 546

An optional compression/decompression engine 546 may be provided within an SPU 500 to, for example, compress and/or decompress content stored in, or released from, VDE objects 300. Compression/decompression engine 546 may implement one or more compression algorithms using hardware circuitry to improve the performance of compression/decompression operations that would otherwise be performed by software operating on microprocessor 520, or outside SPU 500. Decompression is important in the release of data such as video and audio that is usually compressed before distribution and whose decompression speed is important. In some cases, information that is useful for usage monitoring purposes (such as record separators or other delimiters) is "hidden" under a compression layer that must be removed before this information can be detected and used inside SPU 500.

Random Number Generator 542

Optional random number generator 542 may provide specialized hardware circuitry for generating random values (e.g., from inherently unpredictable physical processes such as quantum noise). Such random values are particularly useful for constructing encryption keys or unique identifiers, and for initializing the generation of pseudo-random sequences. Random number generator 542 may produce values of any convenient length, including as small as a single bit per use. A random number of arbitrary size may be constructed by concatenating values produced by random number generator 542. A cryptographically strong pseudo-random sequence may be generated from a random key and seed generated with random number generator 542 and repeated encryption either with the encrypt/decrypt engine 522 or cryptographic algorithms in SPU 500. Such sequences may be used, for example, in private headers to frustrate efforts to determine an encryption key through cryptoanalysis.

Arithmetic Accelerator 544

An optional arithmetic accelerator 544 may be provided within an SPU 500 in the form of hardware circuitry that can rapidly perform mathematical calculations such as multiplication and exponentiation involving large numbers. These calculations can, for example, be requested by microprocessor 520 or encrypt/decrypt engine 522, to assist in the computations required for certain asymmetric encryption/decryption operations. Such arithmetic accelerators are well-known to those skilled in the art. In some implementations, a separate arithmetic accelerator 544 may be omitted and any necessary calculations may be performed by microprocessor 520 under software control.

DMA Controller 526

DMA controller 526 controls information transfers over address/data bus 536 without requiring microprocessor 520 to process each individual data transfer. Typically, microprocessor 520 may write to DMA controller 526 target and destination addresses and the number of bytes to transfer, and DMA controller 526 may then automatically transfer a block of data between components of SPU 500 (e.g., from ROM 532 to RAM 534, between encrypt/decrypt engine 522 and RAM 534, between bus interface unit 530 and RAM 534, etc.). DMA controller 526 may have multiple channels to handle multiple transfers simultaneously. In some implementations, a separate DMA controller 526 may be omitted, and any necessary data movements may be performed by microprocessor 520 under software control.

Bus Interface Unit (BIU) 530

Bus interface unit (BIU) 530 communicates information between SPU 500 and the outside world across the security

barrier **502**. BIU **530** shown in FIG. **9** plus appropriate driver software may comprise the "appliance link" **510** shown in FIG. **6**. Bus interface unit **530** may be modelled after a USART or PCI bus interface in the preferred embodiment. In this example, BIU **530** connects SPU **500** to electronic appliance system bus **653** shown in FIG. **8**. BIU **530** is designed to prevent unauthorized access to internal components within SPU **500** and their contents. It does this by only allowing signals associated with an SPU **500** to be processed by control programs running on microprocessor **520** and not supporting direct access to the internal elements of an SPU **500**.

Memory Management Unit **540**

Memory Management Unit (MMU) **540**, if present, provides hardware support for memory management and virtual memory management functions. It may also provide heightened security by enforcing hardware compartmentalization of the secure execution space (e.g., to prevent a less trusted task from modifying a more trusted task). More details are provided below in connection with a discussion of the architecture of a Secure Processing Environment ("SPE") **503** supported by SPU **500**.

MMU **540** may also provide hardware-level support functions related to memory management such as, for example, address mapping.

SPU Memory Architecture

In the preferred embodiment, SPU **500** uses three general kinds of memory:

(1) internal ROM **532**;

(2) internal RAM **534**; and

(3) external memory (typically RAM and/or disk supplied by a host electronic appliance).

The internal ROM **532** and RAM **534** within SPU **500** provide a secure operating environment and execution space. Because of cost limitations, chip fabrication size, complexity and other limitations, it may not be possible to provide sufficient memory within SPU **500** to store all information that an SPU needs to process in a secure manner. Due to the practical limits on the amount of ROM **532** and RAM **534** that may be included within SPU **500**, SPU **500** may store information in memory external to it, and move this information into and out of its secure internal memory space on an as needed basis. In these cases, secure processing steps performed by an SPU typically must be segmented into small, securely packaged elements that may be "paged in" and "paged out" of the limited available internal memory space. Memory external to an SPU **500** may not be secure. Since the external memory may not be secure, SPU **500** may encrypt and cryptographically seal code and other information before storing it in external memory. Similarly, SPU **500** must typically decrypt code and other information obtained from external memory in encrypted form before processing (e.g., executing) based on it. In the preferred embodiment, there are two general approaches used to address potential memory limitations in a SPU **500**. In the first case, the small, securely packaged elements represent information contained in secure database **610**. In the second case, such elements may represent protected (e.g., encrypted) virtual memory pages. Although virtual memory pages may correspond to information elements stored in secure database **610**, this is not required in this example of a SPU memory architecture.

The following is a more detailed discussion of each of these three SPU memory resources.

SPU Internal ROM

SPU **500** read only memory (ROM) **532** or comparable purpose device provides secure internal non-volatile storage for certain programs and other information. For example, ROM **532** may store "kernel" programs such as SPU control firmware **508** and, if desired, encryption key information and certain fundamental "load modules." The "kernel" programs, load module information, and encryption key information enable the control of certain basic functions of the SPU **500**. Those components that are at least in part dependent on device configuration (e.g., POST, memory allocation, and a dispatcher) may be loaded in ROM **532** along with additional load modules that have been determined to be required for specific installations or applications.

In the preferred embodiment, ROM **532** may comprise a combination of a masked ROM **532***a* and an EEPROM and/or equivalent "flash" memory **532***b*. EEPROM or flash memory **532***b* is used to store items that need to be updated and/or initialized, such as for example, certain encryption keys. An additional benefit of providing EEPROM and/or flash memory **532***b* is the ability to optimize any load modules and library functions persistently stored within SPU **500** based on typical usage at a specific site. Although these items could also be stored in NVRAM **534***b*, EEPROM and/or flash memory **532***b* may be more cost effective.

Masked ROM **532***a* may cost less than flash and/or EEPROM **532***b*, and can be used to store permanent portions of SPU software/firmware. Such permanent portions may include, for example, code that interfaces to hardware elements such as the RTC **528**, encryption/decryption engine **522**, interrupt handlers, key generators, etc. Some of the operating system, library calls, libraries, and many of the core services provided by SPU **500** may also be in masked ROM **532***a*. In addition, some of the more commonly used executables are also good candidates for inclusion in masked ROM **532***a*. Items that need to be updated or that need to disappear when power is removed from SPU **500** should not be stored in masked ROM **532***a*.

Under some circumstances, RAM **534***a* and/or NVRAM **534***b* (NVRAM **534***b* may, for example, be constantly powered conventional RAM) may perform at least part of the role of ROM **532**.

SPU Internal RAM

SPU **500** general purpose RAM **534** provides, among other things, secure execution space for secure processes. In the preferred embodiment, RAM **534** is comprised of different types of RAM such as a combination of high-speed RAM **534***a* and an NVRAM ("non-volatile RAM") **534***b*. RAM **534***a* may be volatile, while NVRAM **534***b* is preferably battery backed or otherwise arranged so as to be non-volatile (i.e., it does not lose its contents when power is turned off).

High-speed RAM **534***a* stores active code to be executed and associated data structures.

NVRAM **534***b* preferably contains certain keys and summary values that are preloaded as part of an initialization process in which SPU **500** communicates with a VDE administrator, and may also store changeable or changing information associated with the operation of SPU **500**. For security reasons, certain highly sensitive information (e.g., certain load modules and certain encryption key related information such as internally generated private keys) needs to be loaded into or generated internally by SPU **500** from time to time but, once loaded or generated internally, should never leave the SPU. In this preferred embodiment, the SPU **500** non-volatile random access memory (NVRAM) **534***b* may be used for securely storing such highly sensitive information. **534***b* is also used by SPU **500** to store data that

may change frequently but which preferably should not be lost in a power down or power fail mode.

NVRAM **534***b* is preferably a flash memory array, but may in addition or alternatively be electrically erasable programmable read only memory (EEPROM), static RAM (SRAM), bubble memory, three dimensional holographic or other electro-optical memory, or the like, or any other writable (e.g., randomly accessible) non-volatile memory of sufficient speed and cost-effectiveness.

SPU External Memory

The SPU **500** can store certain information on memory devices external to the SPU. If available, electronic appliance **600** memory can also be used to support any device external portions of SPU **500** software. Certain advantages may be gained by allowing the SPU **500** to use external memory. As one example, memory internal to SPU **500** may be reduced in size by using non-volatile read/write memory in the host electronic appliance **600** such as a non-volatile portion of RAM **656** and/or ROM **658**.

Such external memory may be used to store SPU programs, data and/or other information. For example, a VDE control program may be, at least in part, loaded into the memory and communicated to and decrypted within SPU **500** prior to execution. Such control programs may be re-encrypted and communicated back to external memory where they may be stored for later execution by SPU **500**. "Kernel" programs and/or some or all of the non-kernel "load modules" may be stored by SPU **500** in memory external to it. Since a secure database **610** may be relatively large, SPU **500** can store some or all of secure database **610** in external memory and call portions into the SPU **500** as needed.

As mentioned above, memory external to SPU **500** may not be secure. Therefore, when security is required, SPU **500** must encrypt secure information before writing it to external memory, and decrypt secure information read from external memory before using it. Inasmuch as the encryption layer relies on secure processes and information (e.g., encryption algorithms and keys) present within SPU **500**, the encryption layer effectively "extends" the SPU security barrier **502** to protect information the SPU **500** stores in memory external to it.

SPU **500** can use a wide variety of different types of external memory. For example, external memory may comprise electronic appliance secondary storage **652** such as a disk; external EEPROM or flash memory **658**; and/or external RAM **656**. External RAM **656** may comprise an external nonvolatile (e.g. constantly powered) RAM and/or cache RAM.

Using external RAM local to SPU **500** can significantly improve access times to information stored externally to an SPU. For example, external RAM may be used:

- to buffer memory image pages and data structures prior to their storage in flash memory or on an external hard disk (assuming transfer to flash or hard disk can occur in significant power or system failure cases);
- provide encryption and decryption buffers for data being released from VDE objects **300**.
- to cache "swap blocks" and VDE data structures currently in use as an aspect of providing a secure virtual memory environment for SPU **500**.
- to cache other information in order to, for example, reduce frequency of access by an SPU to secondary storage **652** and/or for other reasons.

Dual ported external RAM can be particularly effective in improving SPU **500** performance, since it can decrease the data movement overhead of the SPU bus interface unit **530** and SPU microprocessor **520**.

Using external flash memory local to SPU **500** can be used to significantly improve access times to virtually all data structures. Since most available flash storage devices have limited write lifetimes, flash storage needs to take into account the number of writes that will occur during the lifetime of the flash memory. Hence, flash storage of frequently written temporary items is not recommended. If external is non-volatile, then transfer to flash (or hard disk) may not be necessary.

External memory used by SPU **500** may include two categories:

- external memory dedicated to SPU **500**, and
- memory shared with electronic appliance **600**.

For some VDE implementations, sharing memory (e.g., electronic appliance **656**, ROM **658** and/or secondary storage **652**) with CPU **654** or other elements of an electronic appliance **600** may be the most cost elective way to store VDE secure database management files **610** and information that needs to be stored external to SPU **500**. A host system hard disk secondary memory **652** used for general purpose file storage can, for example, also be used to store VDE management files **610**. SPU **500** may be given exclusive access to the external memory (e.g., over a local bus high speed connection provided by BIU **530**). Both dedicated and shared external memory may be provided.

The hardware configuration of an example of electronic appliance **600** has been described above. The following section describes an example of the software architecture of electronic appliance **600** provided by the preferred embodiment, including the structure and operation of preferred embodiment "Rights Operating System" ("ROS") **602**.

Rights Operating System **602**

Rights Operating System ("ROS") **602** in the preferred embodiment is a compact, secure, event-driven, services-based, "component" oriented, distributed multiprocessing operating system environment that integrates VDE information security control information, components and protocols with traditional operating system concepts. Like traditional operating systems, ROS **602** provided by the preferred embodiment is a piece of software that manages hardware resources of a computer system and extends management functions to input and/or output devices, including communications devices. Also like traditional operating systems, preferred embodiment ROS **602** provides a coherent set of basic functions and abstraction layers for hiding the differences between, and many of the detailed complexities of, particular hardware implementations. In addition to these characteristics found in many or most operating systems, ROS **602** provides secure VDE transaction management and other advantageous features not found in other operating systems. The following is a non-exhaustive list of some of the advantageous features provided by ROS **602** in the preferred embodiment:

Standardized interface provides coherent set of basic functions

- simplifies programming
- the same application can run on many different platforms
- Event driven
- eases functional decomposition
- extendible
- accommodates state transition and/or process oriented events
- simplifies task management
- simplifies inter-process communications

Services based
allows simplified and transparent scalability
simplifies multiprocessor support
hides machine dependencies
eases network management and support
Component Based Architecture
processing based on independently deliverable secure components
component model of processing control allows different sequential steps that are reconfigurable based on requirements
components can be added, deleted or modified (subject to permissioning)
full control information over pre-defined and user-defined application events
events can be individually controlled with independent executables
Secure
secure communications
secure control functions
secure virtual memory management
information control structures protected from exposure
data elements are validated, correlated and access controlled
components are encrypted and validated independently
components are tightly correlated to prevent unauthorized use of elements
control structures and secured executables are validated prior to use to protect against tampering
integrates security considerations at the I/O level
provides on-the-fly decryption of information at release time
enables a secure commercial transaction network
flexible key management features
Scalaeble
highly scalaeble across many different platforms
supports concurrent processing in a multiprocessor environment
supports multiple cooperating processors
any number of host or security processors can be supported
control structures and kernel are easily portable to various host platforms and to different processors within a target platform without recompilation
supports remote processing
Remote Procedure Calls may be used for internal OS communications
Highly Integratable
can be highly integrated with host platforms as an additional operating system layer
permits non-secure storage of secured components and information using an OS layer "on top of" traditional OS platforms
can be seamlessly integrated with a host operating system to provide a common usage paradigm for transaction management and content access
integration may take many forms: operating system layers for desktops (e.g., DOS, Windows, Macintosh); device drivers and operating system interfaces for network services (e.g, Unix and Netware); and dedicated component drivers for "low end" set tops are a few of many examples
can be integrated in traditional and real time operating systems
Distributed
provides distribution of control information and reciprocal control information and mechanisms
supports conditional execution of controlled processes within any VDE node in a distributed, asynchronous arrangement
controlled delegation of rights in a distributed environment
supports chains of handling and control
management environment for distributed, occasionally connected but otherwise asynchronous networked database
real time and time independent data management
supports "agent" processes
Transparent
can be seamlessly integrated into existing operating systems
can support applications not specifically written to use it
Network friendly
internal OS structures may use RPCs to distribute processing
subnets may seamlessly operate as a single node or independently

General Background Regarding Operating Systems

An "operating system" provides a control mechanism for organizing computer system resources that allows programmers to create applications for computer systems more easily. An operating system does this by providing commonly used functions, and by helping to ensure compatibility between different computer hardware and architectures (which may, for example, be manufactured by different vendors). Operating systems also enable computer "peripheral device" manufacturers to far more easily supply compatible equipment to computer manufacturers and users.

Computer systems are usually made up of several different hardware components. These hardware components include, for example:

- a central processing unit (CPU) for executing instructions;
- an array of main memory cells (e.g., "RAM" or "ROM") for storing instructions for execution and data acted upon or parameterizing those instructions; and
- one or more secondary storage devices (e.g., hard disk drive, floppy disk drive, CD-ROM drive, tape reader, card reader, or "flash" memory) organized to reflect named elements (a "file system") for storing images of main memory cells.

Most computer systems also include input/output devices such as keyboards, mice, video systems, printers, scanners and communications devices.

To organize the CPU's execution capabilities with available RAM, ROM and secondary storage devices, and to provide commonly used functions for use by programmers, a piece of software called an "operating system" is usually included with the other components. Typically, this piece of software is designed to begin executing after power is applied to the computer system and hardware diagnostics are completed. Thereafter, all use of the CPU, main memory and secondary memory devices is normally managed by this "operating system" software. Most computer operating systems also typically include a mechanism for extending their management functions to I/O and other peripheral devices, including commonly used functions associated with these devices.

By managing the CPU, memory and peripheral devices through the operating system, a coherent set of basic functions and abstraction layers for hiding hardware details allows programmers to more easily create sophisticated applications. In addition, managing the computer's hardware resources with an operating system allows many differences in design and equipment requirements between different manufacturers to be hidden. Furthermore, applications can be more easily shared with other computer users who have the same operating system, with significantly less work to support different manufacturers' base hardware and peripheral devices.

ROS **602** is an Operating System Providing Significant Advantages

ROS **602** is an "operating system." It manages the resources of electronic appliance **600**, and provides a commonly used set of functions for programmers writing applications **608** for the electronic appliance. ROS **602** in the preferred embodiment manages the hardware (e.g., CPU(s), memory(ies), secure RTC(s), and encrypt/decrypt engines) within SPU **500**. ROS may also manage the hardware (e.g., CPU(s) and memory(ies)) within one or more general purpose processors within electronic appliance **600**. ROS **602** also manages other electronic appliance hardware resources, such as peripheral devices attached to an electronic appliance. For example, referring to FIG. **7**, ROS **602** may manage keyboard **612**, display **614**, modem **618**, disk drive **620**, printer **622**, scanner **624**. ROS **602** may also manage secure database **610** and a storage device (e.g., "secondary storage" **652**) used to store secure database **610**.

ROS **602** supports multiple Processors. ROS **602** in the preferred embodiment supports any number of local and/or remote processors. Supported processors may include at least two types: one or more electronic appliance processors **654**, and/or one or more SPUs **500**. A host processor CPU **654** may provide storage, database, and communications services. SPU **500** may provide cryptographic and secured process execution services. Diverse control and execution structures supported by ROS **602** may require that processing of control information occur within a controllable execution space—this controllable execution space may be provided by SPU **500**. Additional host and/or SPU processors may increase efficiencies and/or capabilities. ROS **602** may access, coordinate and/or manage further processors remote to an electronic appliance **600** (e.g., via network or other communications link) to provide additional processor resources and/or capabilities.

ROS **602** is services based. The ROS services provided using a host processor **654** and/or a secure processor (SPU **500**) are linked in the preferred embodiment using a "Remote Procedure Call" ("RPC") internal processing request structure. Cooperating processors may request interprocess services using a RPC mechanism, which is minimally time dependent and can be distributed over cooperating processors on a network of hosts. The multi-processor architecture provided by ROS **602** is easily extensible to support any number of host or security processors. This extensibility supports high levels of scalability. Services also allow functions to be implemented differently on different equipment. For example, a small appliance that typically has low levels of usage by one user may implement a database service using very different techniques than a very large appliance with high levels of usage by many users. This is another aspect of scalability.

ROS **602** provides a distributed processing environment. For example, it permits information and control structures to automatically, securely pass between sites as required to fulfill a user's requests. Communications between VDE modes under the distributed processing features of ROS **602** may include interprocess service requests as discussed above. ROS **602** supports conditional and/or state dependent execution of controlled processors within any VDE node. The location that the process executes and the control structures used may be locally resident, remotely accessible, or carried along by the process to support execution on a remote system.

ROS **602** provides distribution of control information, including for example the distribution of control structures required to permit "agents" to operate in remote environments. Thus, ROS **602** provides facilities for passing execution and/or information control as part of emerging requirements for "agent" processes.

If desired, ROS **602** may independently distribute control information over very low bandwidth connections that may or may not be "real time" connections. ROS **602** provided by the preferred embodiment is "network friendly," and can be implemented with any level of networking protocol. Some examples include e-mail and direct connection at approximately "Layer **5**" of the ISO model.

The ROS **602** distribution process (and the associated auditing of distributed information) is a controlled event that itself uses such control structures. This "reflective" distributed processing mechanism permits ROS **602** to securely distribute rights and permissions in a controlled manner, and effectively restrict the characteristics of use of information content. The controlled delegation of rights in a distributed environment and the secure processing techniques used by ROS **602** to support this approach provide significant advantages.

Certain control mechanisms within ROS **602** are "reciprocal." Reciprocal control mechanisms place one or more control components at one or more locations that interact with one or more components at the same or other locations in a controlled way. For example, a usage control associated with object content at a user's location may have a reciprocal control at a distributor's location that governs distribution of the usage control, auditing of the usage control, and logic to process user requests associated with the usage control. A usage control at a user's location (in addition to controlling one or more aspects of usage) may prepare audits for a distributor and format requests associated with the usage control for processing by a distributor. Processes at either end of a reciprocal control may be further controlled by other processes (e.g., a distributor may be limited by a budget for the number of usage control mechanisms they may produce). Reciprocal control mechanisms may extend over many sites and many levels (e.g., a creator to a distributor to a user) and may take any relationship into account (e.g., creator/distributor, distributor/user, user/user, user/creator, user/creator/distributor, etc.) Reciprocal control mechanisms have many uses in VDE **100** in representing relationships and agreements in a distributed environment.

ROS **602** is scalable. Many portions of ROS **602** control structures and kernel(s) are easily portable to various host platforms without recompilation. Any control structure may be distributed (or redistributed) if a granting authority permits this type of activity. The executable references within ROS **602** are portable within a target platform. Different instances of ROS **602** may execute the references using different resources. For example, one instance of ROS **602** may perform a task using an SPU **500**, while another instance of ROS **602** might perform the same task using a host processing environment running in protected memory that is emulating an SPU in software. ROS **602** control

informationis similarly portable; in many cases the event processing structures may be passed between machines and host platforms as easily as between cooperative processors in a single computer. Appliances with different levels of usage and/or resources available for ROS **602** functions may implement those functions in very different ways. Some services may be omitted entirely if insufficient resources exist. As described elsewhere, ROS **602** "knows" what services are available, and how to proceed based on any given event. Not all events may be processable if resources are missing or inadequate.

ROS **602** is component based. Much of the functionality provided by ROS **602** in the preferred embodiment may be based on "components" that can be securely, independently deliverable, replaceable and capable of being modified (e.g., under appropriately secure conditions and authorizations). Moreover, the "components" may themselves be made of independently deliverable elements. ROS **602** may assemble these elements together (using a construct provided by the preferred embodiment called a "channel") at execution time. For example, a "load module" for execution by SPU **500** may reference one or more "method cores," method parameters and other associated data structures that ROS **602** may collect and assemble together to perform a task such as billing or metering. Different users may have different combinations of elements, and some of the elements may be customizable by users with appropriate authorization. This increases flexibility, allows elements to be reused, and has other advantages.

ROS **602** is highly secure. ROS **602** provides mechanisms to protect information control structures from exposure by end users and conduit hosts. ROS **602** can protect information, VDE control structures and control executables using strong encryption and validation mechanisms. These encryption and validation mechanisms are designed to make them highly resistant to undetected tampering. ROS **602** encrypts information stored on secondary storage device(s) **652** to inhibit tampering. ROS **602** also separately encrypts and validates its various components. ROS **602** correlates control and data structure components to prevent unauthorized use of elements. These features permit ROS **602** to independently distribute elements, and also allows integration of VDE functions **604** with non-secure "other" OS functions **606**.

ROS **602** provided by the preferred embodiment extends conventional capabilities such as, for example, Access Control List (ACL) structures, to user and process defined events, including state transitions. ROS **602** may provide full control information over pre-defined and user-defined application events. These control mechanisms include "go/no-go" permissions, and also include optional event-specific executables that permit complete flexibility in the processing and/or controlling of events. This structure permits events to be individually controlled so that, for example, metering and budgeting may be provided using independent executables. For example, ROS **602** extends ACL structures to control arbitrary granularity of information. Traditional operating systems provide static "go-no go" control mechanisms at a file or resource level; ROS **602** extends the control concept in a general way from the largest to the smallest sub-element using a flexible control structure. ROS **602** can, for example, control the printing of a single paragraph out of a document file.

ROS **602** provided by the preferred embodiment permits secure modification and update of control information governing each component. The control information may be provided in a template format such as method options to an end-user. An end-user may then customize the actual control information used within guidelines provided by a distributor or content creator. Modification and update of existing control structures is preferably also a controllable event subject to auditing and control information.

ROS **602** provided by the preferred embodiment validates control structures and secured executables prior to use. This validation provides assurance that control structures and executables have not been tampered with by end-users. The validation also permits ROS **602** to securely implement components that include fragments of files and other operating system structures. ROS **602** provided by the preferred embodiment integrates security considerations at the operating system I/O level (which is below the access level), and provides "on-the-fly" decryption of information at release time. These features permit non-secure storage of ROS **602** secured components and information using an OS layer "on top of" traditional operating system platforms.

ROS **602** is highly integratable with host platforms as an additional operating system layer. Thus, ROS **602** may be created by "adding on" to existing operating systems. This involves hooking VDE "add ons" to the host operating system at the device driver and network interface levels. Alternatively, ROS **602** may comprise a wholly new operating system that integrates both VDE functions and other operating system functions.

Indeed, there are at least three general approaches to integrating VDE functions into a new operating system, potentially based on an existing operating system, to create a Rights Operating System **602** including:

(1) Redesign the operating system based on VDE transaction management requirements;

(2) Compile VDE API functions into an existing operating systems; and

(3) Integrate a VDE Interpreter into an existing operating system.

The first approach could be most effectively applied when a new operating system is being designed, or if a significant upgrade to an existing operating system is planned. The transaction management and security requirements provided by the VDE functions could be added to the design requirements list for the design of a new operating system that provides, in an optimally efficient manner, an integration of "traditional" operating system capabilities and VDE capabilities. For example, the engineers responsible for the design of the new version or instance of an operating system would include the requirements of VDE metering/transaction management in addition to other requirements (if any) that they use to form their design approach, specifications, and actual implementations. This approach could lead to a "seamless" integration of VDE functions and capabilities by threading metering/transaction management functionality throughout the system design and implementation.

The second approach would involve taking an existing set of API (Application Programmer Interface) functions, and incorporating references in the operating system code to VDE function calls. This is similar to the way that the current Windows operating system is integrated with DOS, wherein DOS serves as both the launch point and as a significant portion of the kernel underpinning of the Windows operating system. This approach would be also provide a high degree of "seamless" integration (although not quite as "seamless" as the first approach). The benefits of this approach include the possibility that the incorporation of metering/transaction management functionality into the new version or instance of an operating system may be accom-

plished with lower cost (by making use of the existing code embodied in an API, and also using the design implications of the API functional approach to influence the design of the elements into which the metering/transaction management functionality is incorporated).

The third approach is distinct from the first two in that it does not incorporate VDE functionality associated with metering/transaction management and data security directly into the operating system code, but instead adds a new generalized capability to the operating system for executing metering/transaction management functionality. In this case, an interpreter including metering/transaction management functions would be integrated with other operating system code in a "stand alone" mode. This interpreter might take scripts or other inputs to determine what metering/transaction management functions should be performed, and in what order and under which circumstances or conditions they should be performed.

Instead of (or in addition to) integrating VDE functions into/with an electronic appliance operating system, it would be possible to provide certain VDE functionality available as an application running on a conventional operating system.

ROS Software Architecture

FIG. **10** is a block diagram of one example of a software structure/architecture for Rights Operating System ("ROS") **602** provided by the preferred embodiment. In this example, ROS **602** includes an operating system ("OS") "core" **679**, a user Application Program Interface ("API") **682**, a "redirector" **684**, an "intercept" **692**, a User Notification/Exception Interface **686**, and a file system **687**. ROS **602** in this example also includes one or more Host Event Processing Environments ("HPEs") **655** and/or one or more Secure Event Processing Environments ("SPEs") **503** (these environments may be generically referred to as "Protected Processing Environments" **650**).

HPE(s) **655** and SPE(s) **503** are self-contained computing and processing environments that may include their own operating system kernel **688** including code and data processing resources. A given electronic appliance **600** may include any number of SPE(s) **503** and/or any number of HPE(s) **655**. HPE(s) **655** and SPE(s) **503** may process information in a secure way, and provide secure processing support for ROS **602**. For example, they may each perform secure processing based on one or more VDE component assemblies **690**, and they may each offer secure processing services to OS kernel **680**.

In the preferred embodiment, SPE **503** is a secure processing environment provided at least in part by an SPU **500**. Thus, SPU **500** provides the hardware tamper-resistant barrier **503** surrounding SPE **503**. SPE **503** provided by the preferred embodiment is preferably:

- a small and compact
- loadable into resource constrained environments such as for example minimally configured SPUs **500**
- dynamically updatable
- extensible by authorized users
- integratable into object or procedural environments
- secure.

In the preferred embodiment, HPE **655** is a secure processing environment supported by a processor other than an SPU, such as for example an electronic appliance CPU **654** general-purpose microprocessor or other processing system or device. In the preferred embodiment, HPE **655** may be considered to "emulate" an SPU **500** in the sense that it may use software to provide some or all of the processing resources provided in hardware and/or firmware by an SPU. HPE **655** in one preferred embodiment of the present invention is full-featured and fully compatible with SPE **503**—that is, HPE **655** can handle each and every service call SPE **503** can handle such that the SPE and the HPE are "plug compatible" from an outside interface standpoint (with the exception that the HPE may not provide as much security as the SPE).

HPEs **655** may be provided in two types: secure and not secure. For example, it may be desirable to provide non-secure versions of HPE **655** to allow electronic appliance **600** to efficiently run non-sensitive VDE tasks using the full resources of a fast general purpose processor or computer. Such non-secure versions of HPE **655** may run under supervision of an instance of ROS **602** that also includes an SPE **503**. In this way, ROS **602** may run all secure processes within SPE **503**, and only use HPE **655** for processes that do not require security but that may require (or run more efficiently) under potentially greater resources provided by a general purpose computer or processor supporting HPE **655**. Non-secure and secure HPE **655** may operate together with a secure SPE **503**.

HPEs **655** may (as shown in FIG. **10**) be provided with a software-based tamper resistant barrier **674** that makes them more secure. Such a software-based tamper resistant barrier **674** may be created by software executing on general-purpose CPU **654**. Such a "secure" HPE **655** can be used by ROS **602** to execute processes that, while still needing security, may not require the degree of security provided by SPU **500**. This can be especially beneficial in architectures providing both an SPE **503** and an HPE **655**. The SPU **502** may be used to perform all truly secure processing, whereas one or more HPEs **655** may be used to provide additional secure (albeit possibly less secure than the SPE) processing using host processor or other general purpose resources that may be available within an electronic appliance **600**. Any service may be provided by such a secure HPE **655**. In the preferred embodiment, certain aspects of "channel processing" appears to be a candidate that could be readily exported from SPE **503** to HPE **655**.

The software-based tamper resistant barrier **674** provided by HPE **655** may be provided, for example, by: introducing time checks and/or code modifications to complicate the process of stepping through code comprising a portion of kernel **688***a* and/or a portion of component assemblies **690** using a debugger; using a map of defects on a storage device (e.g., a hard disk, memory card, etc.) to form internal test values to impede moving and/or copying HPE **655** to other electronic appliances **600**; using kernel code that contains false branches and other complications in flow of control to disguise internal processes to some degree from disassembly or other efforts to discover details of processes; using "self-generating" code (based on the output of a co-sine transform, for example) such that detailed and/or complete instruction sequences are not stored explicitly on storage devices and/or in active memory but rather are generated as needed; using code that "shuffles" memory locations used for data values based on operational parameters to complicate efforts to manipulate such values; using any software and/or hardware memory management resources of electronic appliance **600** to "protect" the operation of HPE **655** from other processes, functions, etc. Although such a software-based tamper resistant barrier **674** may provide a fair degree of security, it typically will not be as secure as the hardware-based tamper resistant barrier **502** provided (at least in part) by SPU **500**. Because security may be better/more effectively enforced with the assistance of hardware security features such as those provided by SPU **500** (and

because of other factors such as increased performance provided by special purpose circuitry within SPU **500**), at least one SPE **503** is preferred for many or most higher security applications. However, in applications where lesser security can be tolerated and/or the cost of an SPU **500** cannot be tolerated, the SPE **503** may be omitted and all secure processing may instead be performed by one or more secure HPEs **655** executing on general-purpose CPUs **654**. Some VDE processes may not be allowed to proceed on reduced-security electronic appliances of this type if insufficient security is provided for the particular process involved.

Only those processes that execute completely within SPEs **503** (and in some cases, HPEs **655**) may be considered to be truly secure. Memory and other resources external to SPE **503** and HPEs **655** used to store and/or process code and/or data to be used in secure processes should only receive and handle that information in encrypted form unless SPE **503**/HPE **655** can protect secure process code and/or data from non-secure processes.

OS "core" **679** in the preferred embodiment includes a kernel **680**, an RPC manager **732**, and an "object switch" **734**. API **682**, HPE **655** and SPE **503** may communicate "event" messages with one another via OS "core" **679**. They may also communicate messages directly with one another without messages going through OS "core" **679**.

Kernel **680** may manage the hardware of an electronic appliance **600**. For example, it may provide appropriate drivers and hardware managers for interacting with input/output and/or peripheral devices such as keyboard **612**, display **614**, other devices such as a "mouse" pointing device and speech recognizer **613**, modem **618**, printer **622**, and an adapter for network **672**. Kernel **680** may also be responsible for initially loading the remainder of ROS **602**, and may manage the various ROS tasks (and associated underlying hardware resources) during execution. OS kernel **680** may also manage and access secure database **610** and file system **687**. OS kernel **680** also provides execution services for applications **608***a*(**1**), **608***a*(**2**), etc. and other applications.

RPC manager **732** performs messaging routing and resource management/integration for ROS **680**. It receives and routes "calls" from/to API **682**, HPE **655** and SPE **503**, for example.

Object switch **734** may manage construction, deconstruction and other manipulation of VDE objects **300**.

User Notification/Exception Interface **686** in the preferred embodiment (which may be considered part of API **682** or another application coupled to the API) provides "pop up" windows/displays on display **614**. This allows ROS **602** to communicate directly with a user without having to pass information to be communicated through applications **608**. For applications that are not "VDE aware," user notification/exception interface **686** may provide communications between ROS **602** and the user.

API **682** in the preferred embodiment provides a standardized, documented software interface to applications **608**. In part, API **682** may translate operating system "calls" generated by applications **608** into Remote Procedure Calls ("RPCs") specifying "events." RPC manager **732** may route these RPCs to kernel **680** or elsewhere (e.g., to HPE(s) **655** and/or SPE(s) **503**, or to remote electronic appliances **600**, processors, or VDE participants) for processing. The API **682** may also service RPC requests by passing them to applications **608** that register to receive and process specific requests.

API **682** provides an "Applications Programming Interface" that is preferably standardized and documented. It provides a concise set of function calls an application program can use to access services provided by ROS **602**. In at least one preferred example, API **682** will include two parts: an application program interface to VDE functions **604**; and an application program interface to other OS functions **606**. These parts may be interwoven into the same software, or they may be provided as two or more discrete pieces of software (for example).

Some applications, such as application **608***a*(**1**) shown in FIG. **11**, may be "VDE aware" and may therefore directly access both of these parts of API **682**. FIG. **11**A shows an example of this. A "VDE aware" application may, for example, include explicit calls to ROS **602** requesting the creation of new VDE objects **300**, metering usage of VDE objects, storing information in VDE-protected form, etc. Thus, a "VDE aware" application can initiate (and, in some examples, enhance and/or extend) VDE functionality provided by ROS **602**. In addition, "VDE aware" applications may provide a more direct interface between a user and ROS **602** (e.g., by suppressing or otherwise dispensing with "pop up" displays otherwise provided by user notification/exception interface **686** and instead providing a more "seamless" interface that integrates application and ROS messages).

Other applications, such as application **608***b* shown in FIG. **11**B, may not be "VDE Aware" and therefore may not "know" how to directly access an interface to VDE functions **604** provided by API **682**. To provide for this, ROS **602** may include a "redirector" **684** that allows such "non-VDE aware" applications **608**(*b*) to access VDE objects **300** and functions **604**. Redirector **684**, in the preferred embodiment, translates OS calls directed to the "other OS functions" **606** into calls to the "VDE functions" **604**. As one simple example, redirector **684** may intercept a "file open" call from application **608**(*b*), determine whether the file to be opened is contained within a VDE container **300**, and if it is, generate appropriate VDE function call(s) to file system **687** to open the VDE container (and potentially generate events to HPE **655** and/or SPE **503** to determine the name(s) of file(s) that may be stored in a VDE object **300**, establish a control structure associated with a VDE object **300**, perform a registration for a VDE object **300**, etc.). Without redirector **684** in this example, a non-VDE aware application such as **608***b* could access only the part of API **682** that provides an interface to other OS functions **606**, and therefore could not access any VDE functions.

This "translation" feature of redirector **684** provides "transparency." It allows VDE functions to be provided to the application **608**(*b*) in a "transparent" way without requiring the application to become involved in the complexity and details associated with generating the one or more calls to VDE functions **604**. This aspect of the "transparency" features of ROS **602** has at least two important advantages:

(a) it allows applications not written specifically for VDE functions **604** ("non-VDE aware applications") to nevertheless access critical VDE functions; and

(b) it reduces the complexity of the interface between an application and ROS **602**.

Since the second advantage (reducing complexity) makes it easier for an application creator to produce applications, even "VDE aware" applications **608***a*(**2**) may be designed so that some call invoking VDE functions **604** are requested at the level of an "other OS functions" call and then "translated" by redirector **684** into a VDE function call (in this sense, redirector **684** may be considered a part of API **682**). FIG. **11**C shows an example of this. Other calls invoking VDE functions **604** may be passed directly without translation by redirector **684**.

Referring again to FIG. **10**, ROS **620** may also include an "interceptor" **692** that transmits and/or receives one or more real time data feeds **694** (this may be provided over cable(s) **628** for example), and routes one or more such data feeds appropriately while providing "translation" functions for real time data sent and/or received by electronic appliance **600** to allow "transparency" for this type of information analogous to the transparency provided by redirector **684** (and/or it may generate one or more real time data feeds).

Secure ROS Components and Component Assemblies

As discussed above, ROS **602** in the preferred embodiment is a component-based architecture. ROS VDE functions **604** may be based on segmented, independently loadable executable "component assemblies" **690**. These component assemblies **690** are independently securely deliverable. The component assemblies **690** provided by the preferred embodiment comprise code and data elements that are themselves independently deliverable. Thus, each component assembly **690** provided by the preferred embodiment is comprised of independently securely deliverable elements which may be communicated using VDE secure communication techniques, between VDE secure subsystems.

These component assemblies **690** are the basic functional unit provided by ROS **602**. The component assemblies **690** are executed to perform operating system or application tasks. Thus, some component assemblies **690** may be considered to be part of the ROS operating system **602**, while other component assemblies may be considered to be "applications" that run under the support of the operating system. As with any system incorporating "applications" and "operating systems," the boundary between these aspects of an overall system can be ambiguous. For example, commonly used "application" functions (such as determining the structure and/or other attributes of a content container) may be incorporated into an operating system. Furthermore, "operating system" functions (such as task management, or memory allocation) may be modified and/or replaced by an application. A common thread in the preferred embodiment's ROS **602** is that component assemblies **690** provide functions needed for a user to fulfill her intended activities, some of which may be "application-like" and some of which may be "operating system-like."

Components **690** are preferably designed to be easily separable and individually loadable. ROS **602** assembles these elements together into an executable component assembly **690** prior to loading and executing the component assembly (e.g., in a secure operating environment such as SPE **503** and/or HPE **655**). ROS **602** provides an element identification and referencing mechanism that includes information necessary to automatically assemble elements into a component assembly **690** in a secure manner prior to, and/or during, execution.

ROS **602** application structures and control parameters used to form component assemblies **690** can be provided by different parties. Because the components forming component assemblies **690** are independently securely deliverable, they may be delivered at different times and/or by different parties ("delivery" may take place within a local VDE secure subsystem, that is submission through the use of such a secure subsystem of control information by a chain of content control information handling participant for the preparation of a modified control information set constitutes independent, secure delivery). For example, a content creator can produce a ROS **602** application that defines the circumstances required for licensing content contained within a VDE object **300**. This application may reference structures provided by other parties. Such references might, for example, take the form of a control path that uses content creator structures to meter user activities; and structures created/owned by a financial provider to handle financial parts of a content distribution transaction (e.g., defining a credit budget that must be present in a control structure to establish creditworthiness, audit processes which must be performed by the licensee, etc.). As another example, a distributor may give one user more favorable pricing than another user by delivering different data elements defining pricing to different users. This attribute of supporting multiple party securely, independently deliverable control information is fundamental to enabling electronic commerce, that is, defining of a content and/or appliance control information set that represents the requirements of a collection of independent parties such as content creators, other content providers, financial service providers, and/or users.

In the preferred embodiment, ROS **602** assembles securely independently deliverable elements into a component assembly **690** based in part on context parameters (e.g., object, user). Thus, for example, ROS **602** may securely assemble different elements together to form different component assemblies **690** for different users performing the same task on the same VDE object **300**. Similarly, ROS **602** may assemble differing element sets which may include, that is reuse, one or more of the same components to form different component assemblies **690** for the same user performing the same task on different VDE objects **300**.

The component assembly organization provided by ROS **602** is "recursive" in that a component assembly **690** may comprise one or more component "subassemblies" that are themselves independently loadable and executable component assemblies **690**. These component "subassemblies" may, in turn, be made of one or more component "sub-subassemblies." In the general case, a component assembly **690** may include N levels of component subassemblies.

Thus, for example, a component assembly **690**(*k*) that may includes a component subassembly **690**(k+l). Component subassembly **690**(k+l), in turn, may include a component sub-subassembly **690**(**3**), . . . and so on to N-level subassembly **690**(k+N). The ability of ROS **602** to build component assemblies **690** out of other component assemblies provides great advantages in terms of, for example, code/data reusability, and the ability to allow different parties to manage different parts of an overall component.

Each component assembly **690** in the preferred embodiment is made of distinct components. FIGS. **11**D–**11**H are abstract depictions of various distinct components that may be assembled to form a component assembly **690**(*k*) showing FIG. **11**I. These same components can be combined in different ways (e.g., with more or less components) to form different component assemblies **690** providing completely different functional behavior. FIG. **11**J is an abstract depiction of the same components being put together in a different way (e.g., with additional components) to form a different component assembly **690**(*j*). The component assemblies **690**(*k*) and **690**(*j*) each include a common feature **691** that interlocks with a "channel" **594** defined by ROS **602**. This "channel" **594** assembles component assemblies **690** and interfaces them with the (rest of) ROS **602**.

ROS **602** generates component assemblies **690** in a secure manner. As shown graphically in FIGS. **11**I and **11**J, the different elements comprising a component assembly **690** may be "interlocking" in the sense that they can only go

together in ways that are intended by the VDE participants who created the elements and/or specified the component assemblies. ROS **602** includes security protections that can prevent an unauthorized person from modifying elements, and also prevent an unauthorized person from substituting elements. One can picture an unauthorized person making a new element having the same "shape" as the one of the elements shown in FIGS. **11D–11H**, and then attempting to substitute the new element in place of the original element. Suppose one of the elements shown in FIG. **11H** establishes the price for using content within a VDE object **300**. If an unauthorized person could substitute her own "price" element for the price element intended by the VDE content distributor, then the person could establish a price of zero instead of the price the content distributor intended to charge. Similarly, if the element establishes an electronic credit card, then an ability to substitute a different element could have disastrous consequences in terms of allowing a person to charge her usage to someone else's (or a non-existent) credit card. These are merely a few simple examples demonstrating the importance of ROS **602** ensuring that certain component assemblies **690** are formed in a secure manner. ROS **602** provides a wide range of protections against a wide range of "threats" to the secure handling and execution of component assemblies **690**.

In the preferred embodiment, ROS **602** assembles component assemblies **690** based on the following types of elements:

- Permissions Records ("PERC"s) **808**;
- Method "Cores" **1000**;
- Load Modules **1100**;
- Data Elements (e.g., User Data Elements ("UDEs") **1200** and Method Data Elements ("MDEs") **1202**); and
- Other component assemblies **690**.

Briefly, a PERC **808** provided by the preferred embodiment is a record corresponding to a VDE object **300** that identifies to ROS **602**, among other things, the elements ROS is to assemble together to form a component assembly **690**. Thus PERC **808** in effect contains a "list of assembly instructions" or a "plan" specifying what elements ROS **602** is to assemble together into a component assembly and how the elements are to be connected together. PERC **808** may itself contain data or other elements that are to become part of the component assembly **690**.

The PERC **808** may reference one or more method "cores" **1000**'. A method core **1000**' may define a basic "method" **1000** (e.g., "control," "billing," "metering," etc.)

In the preferred embodiment, a "method" **1000** is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements, and/or relationships for use in performing, and/or preparing to perform, basic instructions in relation to the operation of one or more electronic appliances **600**. Basic instructions may be comprised of, for example:

- machine code of the type commonly used in the programming of computers; pseudo-code for use by an interpreter or other instruction processing program operating on a computer;
- a sequence of electronically represented logical operations for use with an electronic appliance **600**;
- or other electronic representations of instructions, source code, object code, and/or pseudo code as those terms are commonly understood in the arts.

Information relating to said basic instructions may comprise, for example, data associated intrinsically with basic instructions such as for example, an identifier for the combined basic instructions and intrinsic data, addresses, constants, and/or the like. The information may also, for example, include one or more of the following:

- information that identifies associated basic instructions and said intrinsic data for access, correlation and/or validation purposes;
- required and/or optional parameters for use with basic instructions and said intrinsic data;
- information defining relationships to other methods;
- data elements that may comprise data values, fields of information, and/or the like;
- information specifying and/or defining relationships among data elements, basic instructions and/or intrinsic data;
- information specifying relationships to external data elements;
- information specifying relationships between and among internal and external data elements, methods, and/or the like, if any exist; and
- additional information required in the operation of basic instructions and intrinsic data to complete, or attempt to complete, a purpose intended by a user of a method, where required, including additional instructions and/or intrinsic data.

Such information associated with a method may be stored, in part or whole, separately from basic instructions and intrinsic data. When these components are stored separately, a method may nevertheless include and encompass the other information and one or more sets of basic instructions and intrinsic data (the latter being included because of said other information's reference to one or more sets of basic instructions and intrinsic data), whether or not said one or more sets of basic instructions and intrinsic data are accessible at any given point in time.

Method core **1000**' may be parameterized by an "event code" to permit it to respond to different events in different ways. For example, a METER method may respond to a "use" event by storing usage information in a meter data structure. The same METER method may respond to an "administrative" event by reporting the meter data structure to a VDE clearinghouse or other VDE participant.

In the preferred embodiment, method core **1000**' may "contain," either explicitly or by reference, one or more "load modules" **1100** and one or more data elements (UDEs **1200**, MDEs **1202**). In the preferred embodiment, a "load module" **1100** is a portion of a method that reflects basic instructions and intrinsic data. Load modules **1100** in the preferred embodiment contain executable code, and may also contain data elements ("DTDs" **1108**) associated with the executable code. In the preferred embodiment, load modules **1100** supply the program instructions that are actually "executed" by hardware to perform the process defined by the method. Load modules **1100** may contain or reference other load modules.

Load modules **1100** in the preferred embodiment are modular and "code pure" so that individual load modules may be reenterable and reusable. In order for components **690** to be dynamically updatable, they may be individually addressable within a global public name space. In view of these design goals, load modules **1100** are preferably small, code (and code-like) pure modules that are individually named and addressable. A single method may provide different load modules **1100** that perform the same or similar functions on different platforms, thereby making the method scalable and/or portable across a wide range of different electronic appliances.

UDEs **1200** and MDEs **1202** may store data for input to or output from executable component assembly **690** (or data describing such inputs and/or outputs). In the preferred embodiment, UDEs **1200** may be user dependent, whereas MDEs **1202** may be user independent.

The component assembly example **690**(*k*) shown in FIG. **11**E comprises a method core **1000**', UDEs **1200***a* & **1200***b*, an MDE **1202**, load modules **1100***a*–**1100***d*, and a further component assembly **690**(k+1). As mentioned above, a PERC **808**(*k*) defines, among other things, the "assembly instructions" for component assembly **690**(*k*), and may contain or reference parts of some or all of the components that are to be assembled to create a component assembly.

One of the load modules **1100***b* shown in this example is itself comprised of plural load modules **1100***c*, **1100***d*. Some of the load modules (e.g., **1100***a*, **1100***d*) in this example include one or more "DTD" data elements **1108** (e.g., **1108***a*, **1108***b*). "DTD" data elements **1108** may be used, for example, to inform load module **1100***a* of the data elements included in MDE **1202** and/or UDEs **1200***a*, **1200***b*. Furthermore, DTDs **1108** may be used as an aspect of forming a portion of an application used to inform a user as to the information required and/or manipulated by one or more load modules **1100**, or other component elements. Such an application program may also include functions for creating and/or manipulating UDE(s) **1200**, MDE(s) **1202**, or other component elements, subassemblies, etc.

Components within component assemblies **690** may be "reused" to form different component assemblies. As mentioned above, FIG. **11**F is an abstract depiction of one example of the same components used for assembling component assembly **690**(*k*) to be reused (e.g., with some additional components specified by a different set of "assembly instructions" provided in a different PERC **808**(*l*)) to form a different component assembly **690**(*l*). Even though component assembly **690**(*l*) is formed from some of the same components used to form component assembly **690**(*k*), these two component assemblies may perform completely different processes in complete different ways.

As mentioned above, ROS **602** provides several layers of security to ensure the security of component assemblies **690**. One important security layer involves ensuring that certain component assemblies **690** are formed, loaded and executed only in secure execution space such as provided within an SPU **500**. Components **690** and/or elements comprising them may be stored on external media encrypted using local SPU **500** generated and/or distributor provided keys.

ROS **602** also provides a tagging and sequencing scheme that may be used within the loadable component assemblies **690** to detect tampering by substitution. Each element comprising a component assembly **690** may be loaded into an SPU **500**, decrypted using encrypt/decrypt engine **522**, and then tested/compared to ensure that the proper element has been loaded. Several independent comparisons may be used to ensure there has been no unauthorized substitution. For example, the public and private copies of the element ID may be compared to ensure that they are the same, thereby preventing gross substitution of elements. In addition, a validation/correlation tag stored under the encrypted layer of the loadable element may be compared to make sure it matches one or more tags provided by a requesting process. This prevents unauthorized use of information. As a third protection, a device assigned tag (e.g., a sequence number) stored under an encryption layer of a loadable element may be checked to make sure it matches a corresponding tag value expected by SPU **500**. This prevents substitution of older elements. Validation/correlation tags are typically passed only in secure wrappers to prevent plaintext exposure of this information outside of SPU **500**.

The secure component based architecture of ROS **602** has important advantages. For example, it accommodates limited resource execution environments such as provided by a lower cost SPU **500**. It also provides an extremely high level of configurability. In fact, ROS **602** will accommodate an almost unlimited diversity of content types, content provider objectives, transaction types and client requirements. In addition, the ability to dynamically assemble independently deliverable components at execution time based on particular objects and users provides a high degree of flexibility, and facilitates or enables a distributed database, processing, and execution environment.

One aspect of an advantage of the component-based architecture provided by ROS **602** relates to the ability to "stage" functionality and capabilities over time. As designed, implementation of ROS **602** is a finite task. Aspects of its wealth of functionality can remain unexploited until market realities dictate the implementation of corresponding VDE application functionality. As a result, initial product implementation investment and complexity may be limited. The process of "surfacing" the full range of capabilities provided by ROS **602** in terms of authoring, administrative, and artificial intelligence applications may take place over time. Moreover, already-designed functionality of ROS **602** may be changed or enhanced at any time to adapt to changing needs or requirements.

More Detailed Discussion of Rights Operating System **602** Architecture

FIG. **12** shows an example of a detailed architecture of ROS **602** shown in FIG. **10**. ROS **602** may include a file system **687** that includes a commercial database manager **730** and external object repositories **728**. Commercial database manager **730** may maintain secure database **610**. Object repository **728** may store, provide access to, and/or maintain VDE objects **300**.

FIG. **12** also shows that ROS **602** may provide one or more SPEs **503** and/or one or more HPEs **655**. As discussed above, HPE **655** may "emulate" an SPU **500** device, and such HPEs **655** may be integrated in lieu of (or in addition to) physical SPUs **500** for systems that need higher throughput. Some security may be lost since HPEs **655** are typically protected by operating system security and may not provide truly secure processing. Thus, in the preferred embodiment, for high security applications at least, all secure processing should take place within an SPE **503** having an execution space within a physical SPU **500** rather than a HPE **655** using software operating elsewhere in electronic appliance **600**.

As mentioned above, three basic components of ROS **602** are a kernel **680**, a Remote Procedure Call (RPC) manager **732** and an object switch **734**. These components, and the way they interact with other portions of ROS **602**, will be discussed below.

Kernel **680**

Kernel **680** manages the basic hardware resources of electronic appliance **600**, and controls the basic tasking provided by ROS **602**. Kernel **680** in the preferred embodiment may include a memory manager **680***a*, a task manager **680***b*, and an I/O manager **680***c*. Task manager **680***b* may initiate and/or manage initiation of executable tasks and schedule them to be executed by a processor on which ROS **602** runs (e.g., CPU **654** shown in FIG. **8**). For example, Task manager **680***b* may include or be associated with a "bootstrap loader" that loads other parts of ROS **602**. Task

manager **680***b* may manage all tasking related to ROS **602**, including tasks associated with application program(s) **608**. Memory manager **680***a* may manage allocation, deallocation, sharing and/or use of memory (e.g., RAM **656** shown in FIG. **8**) of electronic appliance **600**, and may for example provide virtual memory capabilities as required by an electronic appliance and/or associated application(s). I/O manager **680***c* may manage all input to and output from ROS **602**, and may interact with drivers and other hardware managers that provide communications and interactivity with physical devices.

RPC Manager **732**

ROS **602** in a preferred embodiment is designed around a "services based" Remote Procedure Call architecture/interface. All functions performed by ROS **602** may use this common interface to request services and share information. For example, SPE(s) **503** provide processing for one or more RPC based services. In addition to supporting SPUs **500**, the RPC interface permits the dynamic integration of external services and provides an array of configuration options using existing operating system components. ROS **602** also communicates with external services through the RPC interface to seamlessly provide distributed and/or remote processing. In smaller scale instances of ROS **602**, a simpler message passing IPC protocol may be used to conserve resources. Tis may limit the configurability of ROS **602** services, but this possible limitation may be acceptable in some electronic appliances.

The RPC structure allows services to be called/requested without the calling process having to know or specify where the service is physically provided, what system or device will service the request, or how the service request will be fulfilled. This feature supports families of services that may be scaled and/or customized for specific applications. Service requests can be forwarded and serviced by different processors and/or different sites as easily as they can be forwarded and serviced by a local service system. Since the same RPC interface is used by ROS **602** in the preferred embodiment to request services within and outside of the operating system, a request for distributed and/or remote processing incurs substantially no additional operating system overhead. Remote processing is easily and simply integrated as part of the same service calls used by ROS **602** for requesting local-based services. In addition, the use of a standard RPC interface ("RSI") allows ROS **602** to be modularized, with the different modules presenting a standardized interface to the remainder of the operating system. Such modularization and standardized interfacing permits different vendors/operating system programmers to create different portions of the operating system independently, and also allows the functionality of ROS **602** to be flexibly updated and/or changed based on different requirements and/or platforms.

RPC manager **732** manages the RPC interface. It receives service requests in the form of one or more "Remote Procedure Calls" (RPCs) from a service requestor, and routes the service requests to a service provider(s) that can service the request. For example, when rights operating system **602** receives a request from a user application via user API **682**, RPC manager **732** may route the service request to an appropriate service through the "RPC service interface" ("RSI"). The RSI is an interface between RPC manager **732**, service requestors, and a resource that will accept and service requests.

The RPC interface (RSI) is used for several major ROS **602** subsystems in the preferred embodiment.

RPC services provided by ROS **602** in the preferred embodiment are divided into subservices, i.e., individual instances of a specific service each of which may be tracked individually by the RPC manager **732**. This mechanism permits multiple instances of a specific service on higher throughput systems while maintaining a common interface across a spectrum of implementations. The subservice concept extends to supporting multiple processors, multiple SPEs **503**, multiple HPEs **655**, and multiple communications services.

The preferred embodiment ROS **602** provides the following RPC based service providers/requestors (each of which have an RPC interface or "RSI" that communicates with RPC manager **732**):

- SPE device driver **736** (this SPE device driver is connected to an SPE **503** in the preferred embodiment);
- HPE Device Driver **738** (this HPE device driver is connected to an HPE **738** in the preferred embodiment);
- Notification Service **740** (this notification service is connected to user notification interface **686** in the preferred embodiment);
- API Service **742** (this API service is connected to user API **682** in the preferred embodiment;
- Redirector **684**;
- Secure Database (File) Manager **744** (this secure database or file manager **744** may connect to and interact with commercial database manager **730** and secure files **610** through a cache manager **746**, a database interface **748**, and a database driver **750**);
- Name Services Manager **752**;
- Outgoing Administrative Objects Manager **754**;
- Incoming Administrative Objects Manager **756**;
- a Gateway **734** to object switch **734** (this is a path used to allow direct communication between RPC manager **732** and Object Switch **734**); and
- Communications Manager **776**.

The types of services provided by HPE **655**, SPE **503**, User Notification **686**, API **742** and Redirector **684** have already been described above. Here is a brief description of the type(s) of services provided by OS resources **744**, **752**, **754**, **756** and **776**:

- Secure Database Manager **744** services requests for access to secure database **610**;
- Name Services Manager **752** services requests relating to user, host, or service identification;
- Outgoing Administrative Objects Manager **754** services requests relating to outgoing administrative objects;
- Incoming Administrative Objects Manager **756** services requests relating to incoming administrative objects; and
- Communications Manager **776** services requests relating to communications between electronic appliance **600** and the outside world.

Object Switch **734**

Object switch **734** handles, controls and communicates (both locally and remotely) VDE objects **300**. In the preferred embodiment, the object switch may include the following elements:

- a stream router **758**;
- a real time stream interface(s) **760** (which may be connected to real time data feed(s) **694**);
- a time dependent stream interface(s) **762**;
- a intercept **692**;

a container manager **764**;

one or more routing tables **766**; and

buffering/storage **768**.

Stream router **758** routes to/from "real time" and "time independent" data streams handled respectively by real time stream interface(s) **760** and time dependent stream interface (s) **762**. Intercept **692** intercepts I/O requests that involve real-time information streams such as, for example, real time feed **694**. The routing performed by stream router **758** may be determined by routing tables **766**. Buffering/storage **768** provides temporary store-and-forward, buffering and related services. Container manager **764** may (typically in conjunction with SPE **503**) perform processes on VDE objects **300** such as constructing, deconstructing, and locating portions of objects.

Object switch **734** communicates through an Object Switch Interface ("OSI") with other parts of ROS **602**. The Object Switch Interface may resemble, for example, the interface for a Unix socket in the preferred embodiment. Each of the "OSI" interfaces shown in FIG. **12** have the ability to communicate with object switch **734**.

ROS **602** includes the following object switch service providers/resources (each of which can communicate with the object switch **734** through an "OSI"):

Outgoing Administrative Objects Manager **754**;

Incoming Administrative Objects Manager **756**;

Gateway **734** (which may translate RPC calls into object switch calls and vice versa so RPC manager **732** may communicate with object switch **734** or any other element having an OSI to, for example, provide and/or request services);

External Services Manager **772**;

Object Submittal Manager **774**; and

Communications Manager **776**.

Briefly,

Object Repository Manager **770** provides services relating to access to object repository **728**;

External Services Manager **772** provides services relating to requesting and receiving services externally, such as from a network resource or another site;

Object Submittal Manager **774** provides services relating to how a user application may interact with object switch **734** (since the object submittal manager provides an interface to an application program **608**, it could be considered part of user API **682**); and

Communications Manager **776** provides services relating to communicating with the outside world.

In the preferred embodiment, communications manager **776** may include a network manager **780** and a mail gateway (manager) **782**. Mail gateway **782** may include one or more mail filters **784** to, for example, automatically route VDE related electronic mail between object switch **734** and the outside world electronic mail services. External Services Manager **772** may interface to communications manager **776** through a Service Transport Layer **786**. Service Transport Layer **786***a* may enable External Services Manager **772** to communicate with external computers and systems using various protocols managed using the service transport layer **786**.

The characteristics of and interfaces to the various subsystems of ROS **680** shown in FIG. **12** are described in more detail below.

RPC Manager **732** and Its RPC Services Interface

As discussed above, the basic system services provided by ROS **602** are invoked by using an RPC service interface (RSI). This RPC service interface provides a generic, standardized interface for different services systems and subsystems provided by ROS **602**.

RPC Manager **732** routes RPCs requesting services to an appropriate RPC service interface. In the preferred embodiment, upon receiving an RPC call, RPC manager **732** determines one or more service managers that are to service the request. RPC manager **732** then routes a service request to the appropriate service(s) (via a RSI associated with a service) for action by the appropriate service manager(s).

For example, if a SPE **503** is to service a request, the RPC Manager **732** routes the request to RSI **736***a,* which passes the request on to SPE device driver **736** for forwarding to the SPE. Similarly, if HPE **655** is to service the request, RPC Manager **732** routes the request to RSI **738***a* for forwarding to a HPE. In one preferred embodiment, SPE **503** and HPE **655** may perform essentially the same services so that RSIs **736***a,* **738***a* are different instances of the same RSI. Once a service request has been received by SPE **503** (or HPE **655**), the SPE (or HPE) typically dispatches the request internally using its own internal RPC manager (as will be discussed shortly). Processes within SPEs **503** and HPEs **655** can also generate RPC requests. These requests may be processed internally by a SPE/HPE, or if not internally serviceable, passed out of the SPE/HPE for dispatch by RPC Manager **732**.

Remote (and local) procedure calls may be dispatched by a RPC Manager **732** using an "RPC Services Table." An RPC Services Table describes where requests for specific services are to be routed for processing. Each row of an RPC Services Table in the preferred embodiment contains a services ID, the location of the service, and an address to which control will be passed to service a request. An RPC Services Table may also include control information that indicates which instance of the RPC dispatcher controls the service. Both RPC Manager **732** and any attached SPEs **503** and HPEs **655** may have symmetric copies of the RPC Services Table. If an RPC service is not found in the RPC services tables, it is either rejected or passed to external services manager **772** for remote servicing.

Assuming RPC manager **732** finds a row corresponding to the request in an RPC Services Table, it may dispatch the request to an appropriate RSI. The receiving RSI accepts a request from the RPC manager **732** (which may have looked up the request in an RPC service table), and processes that request in accordance with internal priorities associated with the specific service.

In the preferred embodiment, RPC Service Interface(s) supported by RPC Manager **732** may be standardized and published to support add-on service modules developed by third party vendors, and to facilitate scalability by making it easier to program ROS **602**. The preferred embodiment RSI closely follows the DOS and Unix device driver models for block devices so that common code may be developed for many platforms with minimum effort. An example of one possible set of common entry points are listed below in the table.

Interface call	Description
SVC_LOAD	Load a service manager and return its status.
SVC_UNLOAD	Unload a service manager.
SVC_MOUNT	Mount (load) a dynamically loaded subservice and return its status.
SVC_UNMOUNT	Unmount (unload) a dynamically loaded subservice.

-continued

Interface call	Description
SVC_OPEN	Open a mounted subservice.
SVC_CLOSE	Close a mounted subservice.
SVC_READ	Read a block from an opened subservice.
SVC_WRITE	Write a block to an opened subservice.
SVC_IOCTL	Control a subservice or a service manager.

Load

In the preferred embodiment, services (and the associated RSIs they present to RPC manager **732**) may be activated during boot by an installation boot process that issues an RPC LOAD. This process reads an RPC Services Table from a configuration file, loads the service module if it is run time loadable (as opposed to being a kernel linked device driver), and then calls the LOAD entry point for the service. A successful return from the LOAD entry point will indicate that the service has properly loaded and is ready to accept requests.

RPC LOAD Call Example: SVC_LOAD (long service_id)

This LOAD interface call is called by the RPC manager **732** during rights operating system **602** initialization. It permits a service manager to load any dynamically loadable components and to initialize any device and memory required by the service. The service number that the service is loaded as is passed in as service_id parameter. In the preferred embodiment, the service returns 0 is the initialization process was completed successfully or an error number if some error occurred.

Mount

Once a service has been loaded, it may not be fully functional for all subservices. Some subservices (e.g., communications based services) may require the establishment of additional connections, or they may require additional modules to be loaded. If the service is defined as "mountable," a RPC manager **732** will call the MOUNT subservice entry point with the requested subservice ID prior to opening an instance of a subservice.

RPC MOUNT Call Example:

SVC_MOUNT (long service_id, long subservice_id, BYTE *buffer)

This MOUNT interface call instructs a service to make a specific subservice ready. This may include services related to networking, communications, other system services, or external resources. The service_id and subservice_id parameters may be specific to the specific service being requested. The buffer parameter is a memory address that references a control structure appropriate to a specific service.

Open

Once a service is loaded and "mounted," specific instances of a service may be "opened" for use. "Opening" an instance of a service may allocate memory to store control and status information. For example, in a BSD socket based network connection, a LOAD call will initialize the software and protocol control tables, a MOUNT call will specify networks and hardware resources, and an OPEN will actually open a socket to a remote installation.

Some services, such as commercial database manager **730** that underlies the secure database service, may not be "mountable." In this case, a LOAD call will make a connection to a database manager **730** and ensure that records are readable. An OPEN call may create instances of internal cache manager **746** for various classes of records.

RPC OPEN Call Example:

SVC_OPEN (long service_id, long subservice_id, BYTE *buffer, int (*receive) (long request_id))

This OPEN interface call instructs a service to open a specific subservice. The service_id and subservice_id parameters are specific to the specific service being requested, and the buffer parameter is a memory address that references a control structure appropriate to a specific service.

The optional receive parameter is the address of a notification callback function that is called by a service whenever a message is ready for the service to retrieve it. One call to this address is made for each incoming message received. If the caller passes a NULL to the interface, the software will not generate a callback for each message.

Close, Unmount and Unload

The converse of the OPEN, MOUNT, and LOAD calls are CLOSE, UNMOUNT, and UNLOAD. These interface calls release any allocated resources back to ROS **602** (e.g., memory manager **680***a*).

RPC CLOSE Call Example: SVC_CLOSE (long svc_handle)

This LOAD interface call closes an open service "handle." A service "handle" describes a service and subservice that a user wants to close. The call returns 0 if the CLOSE request succeeds (and the handle is no longer valid) or an error number.

RPC UNLOAD Call Example: SVC_UNLOAD (void)

This UNLOAD interface call is called by a RPC manager **732** during shutdown or resource reallocation of rights operating system **602**. It permits a service to close any open connections, flush buffers, and to release any operating system resources that it may have allocated. The service returns 0.

RPC UNMOUNT Call Example: SVC_UNMOUNT (long service_id, long subservice_id)

This UNMOUNT interface call instructs a service to deactivate a specific subservice. The service_id and subservice_id parameters are specific to the specific service being requested, and must have been previously mounted using the SVC_MOUNT() request. The call releases all system resources associated with the subservice before it returns.

Read and Write

The READ and WRITE calls provide a basic mechanism for sending information to and receiving responses from a mounted and opened service. For example, a service has requests written to it in the form of an RPC request, and makes its response available to be read by RPC Manager **732** as they become available.

RPC READ Call Example:

SVC_READ (long svc_handle, long request_id, BYTE *buffer, long size)

This READ call reads a message response from a service. The svc_handle and request_id parameters uniquely identify a request. The results of a request will be stored in the user specified buffer up to size bytes. If the buffer is too small, the first size bytes of the message will be stored in the buffer and an error will be returned.

If a message response was returned to the caller's buffer correctly, the function will return 0. Otherwise, an error message will be returned.

RPC WRITE Call Example:

SVC_write (long service_id, long subservice_id, BYTE *buffer, long size, int (*receive) (long request_id)

This WRITE call writes a message to a service and subservice specified by the service_id/subservice_id parameter pair. The message is stored in buffer (and usually conforms to the VDE RPC message format) and is size bytes long. The function returns the request id for the message (if

it was accepted for sending) or an error number. If a user specifies the receive callback functions, all messages regarding a request will be sent to the request specific callback routine instead of the generalized message callback.

Input/Output Control

The IOCTL ("Input/Output ConTroL") call provides a mechanism for querying the status of and controlling a loaded service. Each service type will respond to specific general IOCTL requests, all required class IOCTL requests, and service specific IOCTL requests.

RPC IOCTL Call Example: ROI_SVC_IOCTL (long service_id, long subservice_id,

int command, BYTE *buffer)

This IOCTL function provides a generalized control interface for a RSI. A user specifies the service_id parameter and an optional subservice_id parameter that they wish to control. They specify the control command parameter(s), and a buffer into/from which the command parameters may be written/read. An example of a list of commands and the appropriate buffer structures are given below.

Command	Structure	Description
GET_INFO	SVC_INFO	Returns information about a service/subservice.
GET_STATS	SVC_STATS	Returns current statistics about a service/subservice.
CLR_STATS	None	Clears the statistics about a service/subservice.

Now that a generic RPC Service Interface provided by the preferred embodiment has been described, the following description relates to particular examples of services provided by ROS **602**.

SPE Device Driver **736**

SPE device driver **736** provides an interface between ROS **602** and SPE **503**. Since SPE **503** in the preferred embodiment runs within the confines of an SPU **500**, one aspect of this device driver **736** is to provide low level communications services with the SPU **500** hardware. Another aspect of SPE device driver **736** is to provide an RPC service interface (RSI) **736***a* particular to SPE **503** (this same RSI may be used to communicate with HPE **655** through HPE device driver **738**).

SPE RSI **736***a* and driver **736** isolates calling processes within ROS **602** (or external to the ROS) from the detailed service provided by the SPE **503** by providing a set of basic interface points providing a concise function set. This has several advantages. For example, it permits a full line of scaled SPUs **500** that all provide common functionality to the outside world but which may differ in detailed internal structure and architecture. SPU **500** characteristics such as the amount of memory resident in the device, processor speed, and the number of services supported within SPU **500** may be the decision of the specific SPU manufacturer, and in any event may differ from one SPU configuration to another. To maintain compatibility, SPE device driver **736** and the RSI **736***a* it provides conform to a basic common RPC interface standard that "hides" differences between detailed configurations of SPUs **500** and/or the SPEs **503** they may support.

To provide for such compatibility, SPE RSI **736***a* in the preferred embodiment follows a simple block based standard. In the preferred embodiment, an SPE RSI **736***a* may be modeled after the packet interfaces for network Ethernet cards. This standard closely models the block mode interface characteristics of SPUs **500** in the preferred embodiment.

An SPE RSI **736***a* allows RPC calls from RPC manager **732** to access specific services provided by an SPE **736**. To do this, SPE RSI **736***a* provides a set of "service notification address interfaces." These provide interfaces to individual services provided by SPE **503** to the outside world. Any calling process within ROS **602** may access these SPE-provided services by directing an RPC call to SPE RSI **736***a* and specifying a corresponding "service notification address" in an RPC call. The specified "service notification address" causes SPE **503** to internally route an RPC call to a particular service within an SPE. The following is a listing of one example of a SPE service breakdown for which individual service notification addresses may be provided:

Channel Services Manager

Authentication Manager/Secure Communications Manager

Secure Database Manager

The Channel Services Manager is the principal service provider and access point to SPE **503** for the rest of ROS **602**. Event processing, as will be discussed later, is primarily managed (from the point of view of processes outside SPE **503**) by this service. The Authentication Manager/Secure Communications Manager may provide login/logout services for users of ROS **602**, and provide a direct service for managing communications (typically encrypted or otherwise protected) related to component assemblies **690**, VDE objects **300**, etc. Requests for display of information (e.g., value remaining in a financial budget) may be provided by a direct service request to a Secure Database Manager inside SPE **503**. The instances of Authentication Manager/Secure Communications Manager and Secure Database Manager, if available at all, may provide only a subset of the information and/or capabilities available to processes operating inside SPE **503**. As stated above, most (potentially all) service requests entering SPE are routed to a Channel Services Manager for processing. As will be discussed in more detail later on, most control structures and event processing logic is associated with component assemblies **690** under the management of a Channel Services Manager.

The SPE **503** must be accessed through its associated SPE driver **736** in this example. Generally, calls to SPE driver **736** are made in response to RPC calls. In this example, SPE driver RSI **736***a* may translate RPC calls directed to control or ascertain information about SPE driver **736** into driver calls. SPE driver RSI **736***a* in conjunction with driver **736** may pass RPC calls directed to SPE **503** through to the SPE.

The following table shows one example of SPE device driver **736** calls:

Entry Point	Description
SPE_info()	Returns summary information about the SPE driver 736 (and SPE 503)
SPE_initialize_interface()	Initializes SPE driver 736, and sets the default notification address for received packets.
SPE_terminate_interface()	Terminates SPE driver 736 and resets SPU 500 and the driver 736.
SPE_reset_interface()	Resets driver 736 without resetting SPU 500.
SPE_get_stats()	Return statistics for notification addresses and/or an entire driver 736.
SPE_clear_stats()	Clears statistics for a specific notification address and/or an entire driver 736.
SPE_set_notify()	Sets a notification address for a specific service ID.
SPE_get_notify()	Returns a notification address for a specific service ID.