COPY

FILED
2016 APR 20 PM 2:49
CLERK U.S. DISTRICT COURT
DISTRICT OF DELAWARE

Dr. Lakshmi Arunachalam
222 Stanford Avenue
Menlo Park, CA 94025
Telephone: (650) 690-0995
Facsimile: (650) 854-3393
Email: laks22002@yahoo.com
Pro Se Plaintiff
Dr. Lakshmi Arunachalam

Paid
Rec #
25542

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

DR. LAKSHMI ARUNACHALAM,

Plaintiff,

v.

INTERNATIONAL BUSINESS MACHINES CORPORATION AND DOES 1-100,

Defendant(s).

C.A. No. 16 - 281

COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS;

Date Filed: April 18, 2016
JURY TRIAL DEMANDED

18 U. S. C. 1961 et seq.;
18 U. S. C. 1964
(Civil RICO Remedies);

COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS

INTRODUCTION

Pro Se Plaintiff Dr. Lakshmi Arunachalam (hereafter "Dr. Arunachalam") hereby files

this complaint for patent infringement of Plaintiff's U.S. 7,340,506 Patent/US 7,340,506 C1 ("the '506 patent") against the Defendant(s) and a verified complaint for declaratory and injunctive relief and damages from racketeering, conspiracy to engage in a pattern of racketeering activity and related claims. This is a complex civil action for RICO remedies authorized by the federal statutes at 18 U. S. C. 1961 et seq.; for declaratory and injunctive relief; for actual, consequential and exemplary damages; and for all other relief which this Court deems just and proper under all circumstances which have occasioned this Initial COMPLAINT. See 18 U. S. C. §§ 1964 (a) and (c) ("Civil RICO").

The primary cause of this action is a widespread enterprise engaged in a *pattern of racketeering activity* across State lines, and a conspiracy to engage in *racketeering activity* involving numerous RICO predicate acts during at least the past ten (10) calendar years. The predicate acts alleged here cluster around patent infringement, trafficking in certain goods bearing counterfeit marks, tampering with a Federal Witness, interstate transportation of stolen property and obstruction of justice. See 18 U. S. C. §§ 2319, 2320, 1512, 1513, 2315, 1503, 1510, 1511 and 1581-1588 respectively.

Other RICO predicate acts, although *appearing* to be isolated events, were actually part of the overall conspiracy and *pattern of racketeering activity* alleged herein. See 18 U.S.C. §§ 1341 and 1344, respectively.

The primary objective of the racketeering *enterprise* has been to inflict severe and sustained economic hardship upon Plaintiff, with the intent of impairing, obstructing, preventing and discouraging Plaintiff from writing, publishing, investigating and conducting judicial activism as the inventor of valid patents and inventions of Web applications on a Web browser..

Dr. Arunachalam alleges upon information and belief as follows:

PARTIES

1. Plaintiff Dr. Arunachalam, residing at 222 Stanford Avenue, Menlo Park, California 94025, is the inventor and assignee of the patent asserted here.

2. Having a priority date of 1995, the '506 patent discloses the fundamental technology underlying Web applications displayed on a Web browser, that are reflected in the Defendant(s)' accused systems.

3. Having a priority date of 1995, the '506 patent discloses the fundamental technology underlying Web commerce and other Web applications displayed on a Web browser. The examples of the pioneering technology in the patent were directed to Web banking, payroll processing and other financial services on the Web which are the same as in the Defendant's accused systems. The patent pioneered interactive Web applications. The priority application, Provisional Patent Application with S/N, 60/006,634, was the first to disclose a Web application displayed on a Web browser/web page and providing a value-added network service over the Web for connecting a Web client to a provider's (e.g. Web merchant) services, as opposed to the then state-of-the-art's reliance on CGI scripting and hyperlinks. Thus, the patent discloses the fundamental technology underlying Web commerce and other online services by use of Web applications displayed on a Web page/Web browser.

4. Upon information and belief, defendant International Business Machines Corporation ("IBM") is a corporation organized and existing under the laws of the State of New York, with its principal place of business at 1 New Orchard Road, Armonk, New York 10504. IBM is registered to do business in Delaware and has a registered agent for service located at The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801. IBM resides in this judicial district and transacts business throughout the State

of Delaware, including this judicial district.

5. Defendant IBM is using Plaintiff's patented Web applications on a Web browser. Plaintiff's patented technology has created the millennial generation and transformed the way we live, work and play and is mission critical to how the Defendant conducts its business and operations today on the Web.

6. Plaintiff is ignorant of the true names and capacities of the Defendants sued herein as DOES 1 through 100, inclusive, and therefore sue these Defendants by such fictitious names. Plaintiff will amend this Complaint to allege the true names and capacities of DOES 1 through 100, inclusive, when Plaintiff ascertains the identity of such Defendants. Plaintiff is informed and believes, and thereon alleges, that each of these Defendants is responsible in some manner for the acts and omissions which damaged Plaintiff, and that Plaintiff's damages as alleged herein were proximately caused by their actions or omissions.

JURISDICTION AND VENUE

7. This is an action for patent infringement of Plaintiff's '506 patent" under the patent laws of the United States, Title 35 of the United States Code. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a). Also, this Court has original jurisdiction pursuant to the civil RICO remedies at 18 U.S.C. 1964.

8. This Court has personal jurisdiction over the Defendant because IBM has established minimum contacts with the forum and because of its presence and business activities within this judicial district. IBM has transacted business and committed acts of infringement within the State of Delaware and within this District, and is subject to the personal jurisdiction of this Court. IBM is a corporation organized and existing under the laws of the State of New York. IBM is registered to do business in Delaware and has a registered agent for service located at The

Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801. IBM resides in this judicial district and transacts business throughout the State of Delaware, including this judicial district. The Court has personal jurisdiction over IBM, which has purposefully availed itself of the privileges of conducting business in the State of Delaware and has sought the protection and benefits of the laws of the State; and regularly conducts business within the State of Delaware; and Plaintiff's cause of action arise directly from IBM's business contacts and other activities in the State of Delaware. IBM has placed and continues to place products used to practice Dr. Arunachalam's patented methods and systems (identified below) into the stream of commerce, which stream is directed at this district, and knows or should know that such products are used throughout the United States, including in this district.

9. Upon information and belief, IBM is subject to the personal jurisdiction of this Court and is amenable to service of process pursuant to the Delaware long-arm statute, 10 Del. C. § 3104 and Fed. R. Civ. P. 4 (e).

10. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b), 1391(c) and 1400(b).

BACKGROUND

11. On March 4, 2008, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 7,340,506 ("the '506 Patent"), entitled "Value- Added Network Switching and Object Routing A Network," to Dr. Lakshmi Arunachalam's company, WebXchange, Inc, in which she is the majority shareholder with 100% voting rights. Dr. Arunachalam is the assignee of all rights, title, and interest in the '506 Patent, including the right to recover damages for past infringement. A copy of the '506 Patent is attached to the Complaint as Exhibit A.

12. Patent 7,340,506 underwent a pre-AIA inter-partes re-examination Control No.

95/001,129 by the United States Patent and Trademark Office, in which the Third Party Requester was Microsoft. Two claims emerged successfully out of the re-examination and the inter-partes reexamination certificate US 7,340,506 C1 was issued under 35 U.S.C. 316 on October 15, 2014. A copy of the inter-partes reexamination certificate issued is attached to the Complaint as Exhibit B. The '506 patent is presumed to be, and is valid and enforceable. The defendant IBM is not licensed under the '506 patent.

13. Upon information and belief, IBM has infringed and is continuing to infringe and contributorily infringes and/or induces others to infringe, one or more claims of the '506 patent by engaging in acts constituting infringement under 35 U.S.C. § 271, included but not limited to practicing one or more claims of the re-examined and allowed claims, inducing others to practice one or more of the said claims, and/or contributing to another's practice of one or more of the said claims in this District and elsewhere in the United States, by means of at least IBM's WebSphere and other web application/web application development platform and tools, products and services.

14. Defendant IBM provides web application development platform, tools, web applications, products and services, value-added network services, for example, online financial services via electronic means accessible through several web sites, which include, but are not limited to the following websites: http://www.ibm.com. Each of the Defendant's products and services enable Web applications, for example, Web banking applications and other Web financial transactional features, which are exemplified, in part, by screenshots of their opening screen which displays the various value-added network services over the Web of the inventions of the patent-in-suit, such as paying bills, transfer funds between accounts, and many, many more.

15. As reflected in the screenshots, each of the Defendant's and its customers' on-line (for

example, financial system) provides a plurality of value added network services over the Web, applications displayed on a Web browser, for rendering value-added network services, for example, financial services, practicing the claimed inventions. For example, a user of each of the Defendant's system may choose to transfer assets between checking and savings accounts, or transfer assets to third-parties by using the application displayed on a Web browser/web page.

16. Defendant IBM makes, uses and sells, inter alia, at least WebSphere and its associated programs, which comprise the claimed inventions and operates without authority one or more apparatus, reflected in at least the websites cited above, wherein the first computer system offering the value-added network service comprising access to employee payroll information over the Web.

17. Defendant IBM makes and uses value-added network services, which are practiced using the claimed inventions. Hereafter, the word "Service" refers to applications offered as value-added network services provided by online service portals, including at least those listed above. These sites and Services can be accessed from stationary personal computers or from mobile devices such as laptop computers, smartphones and tablets. Upon accessing these sites, Defendant's clients or customers and their customers can, for example, view and service accounts; make transfers; pay and manage bills online using Bill Pay ("Bill Pay") which allows users to schedule bill payments through the Service; initiate and monitor Wire Transfer service; and make and manage investments through, for example, through the brokerage services, including trading securities. Through IBM's customers' Mobile Banking websites and mobile apps, the customers or clients of IBM's customers can access their accounts, transfer funds, pay bills, place and track brokerage trades, and locate ATMs via mobile devices.

COUNT I: INFRINGEMENT OF THE '506 PATENT BY IBM

18. Dr. Arunachalam incorporates and re-alleges paragraphs 1-16.

19. Upon information and belief, Defendant IBM has directly infringed and is continuing to infringe one or more claims of the '506 Patent by operating without authority one or more apparatus, reflected in the websites cited above, wherein the transaction is handed over to an exchange, wherein the exchange manages the connection between the user and the online service operating across the digital network, which offers value-added network services atop the Web. Defendant IBM operates without authority one or more apparatus, reflected in at least the websites cited above, wherein the first computer system offering the value-added network service comprising access to employee payroll information over the Web. Specifically, Defendant IBM infringed and infringes, because (i) it operated and continues to operate applications and software including, but not limited to, those maintained on servers located in and/or accessible from the United States under the United States/IBM's' control that, as reflected in the website, *inter alia*, provide an apparatus for providing a service over a digital network, the apparatus comprising:

a processor;

a machine-readable storage device including one or more instructions executable by the processor for sending first display information from a first computer system to a user device, wherein the first display information includes a control associated with a commercial service;

accepting a first signal in response to a user input to activate the control; and

initiating, in response to the first signal, communication between the user device and a second computer system, wherein the second computer system acts to send second display information to the user device, wherein the second display information includes a list of at least one commercial service; wherein the second computer system further acts to accept a second signal

in response to a user input to select a commercial service from the list; and to complete a commercial transaction relating to the selected commercial service;

associating an object identity with information entries and attributes, wherein the object identity represents a networked object;

storing said information entries and said attributes in a virtual information store; and

assigning a unique network address to said object identity,

wherein (a) the transaction is handed over to an exchange, wherein the exchange manages the connection between the user and the commercial service, wherein the commercial service is an online service operating across the digital network, wherein the digital network is a value-added service network atop the Web, (ii) the first computer system offering the commercial service comprising access to employee payroll information on a value-added service network atop the Web, and (iii) utilized and is utilizing computer equipment, including, without limitation, computer equipment that stores, serves, and/or runs the foregoing.

20. IBM's infringement is by making, using and selling without authority WebSphere and other web application development platforms, tools, web applications, products and services, and by making and using IBM Cloud Services. Defendant's infringement has injured Plaintiff. Accordingly, Plaintiff is entitled to recover damages adequate to compensate it for such infringement, but in no event less than a reasonable royalty, and an injunction to prohibit further infringement of the '506 Patent or future compensation for use of the inventions.

21. IBM has directly infringed and is continuing to infringe one or more claims of the '506 Patent by operating without authority one or more online and mobile banking systems providing Services which utilize the patented inventions.

22. Upon information and belief, IBM has infringed and is continuing to infringe one or more

claims of the '506 patent in this District and elsewhere in the United States by practicing one or more of the claims of the '506 patent, by means of at least the IBM WebSphere and other web application development tools, platforms and web application products and services.

23. Defendant IBM's online practices of the patented inventions are reflected in, but not limited to, the websites http://www.ibm.com and the websites of Defendant IBM's customers. Defendant's servers providing the claimed apparatus are located in the United States under IBM's control.

24. Upon information and belief, IBM is contributing to the infringement of the '506 patent by others in this District and elsewhere in the United States by contributing to another's practice of one or more of the claims of the '506 patent. The direct infringement occurs by activities of the end users of at least IBM's web application products and services.

25. Upon information and belief, IBM is inducing the infringement of the '506 patent by others in this District and elsewhere in the United States by inducing others to practice one or more of the claims of the '506 patent. The direct infringement occurs by activities of the end users of at least IBM's web application products and services.

26. Upon information and belief, IBM, in its practicing one or more claims of the '506 patent, its inducing others to practice one or more claims of the '506 patent, and/or its contributing to another's practice of one or more claims of the '506 patent, is acting despite an objectively high likelihood that its actions constitute infringement of the '506 patent. Thus, at least IBM's ongoing infringement of the '506 patent after notice of this Complaint is willful.

27. Upon information and belief, IBM's infringement of the '506 patent will continue unless enjoined by this Court.

28. As a direct and proximate consequence of IBM's infringement of the '506 patent, Dr.

Arunachalam has suffered and will continue to suffer irreparable injury and damages, in an amount to be determined at trial, for which Dr. Arunachalam is entitled to relief.

29. Upon information and belief, IBM's infringement of the '506 patent is exceptional and entitles Dr. Arunachalam to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT II: CIVIL RACKETEERING BY IBM

PARTIAL LIST OF RICO PREDICATE ACTS

30. Particular attention of this Court is now drawn to Exhibits A2, C1, D1 and D2 and to the legislative history of the Anticounterfeiting Consumer Protection Act of 1996 ("ACPA"), available from the House Congressional Record dated June 4, 1996, 110 Stat. 1386, July 2, 1996.

31. The ACPA is particularly relevant to the instant case, because it elevated copyright and trademark infringement to the status of RICO predicate acts, and cited superb reasons for doing so. An excellent discussion of the legal implications of the ACPA, in the context of other applicable federal laws, are available at LETTER TO JON MUMMOLO, *Washington Square News*, Nov. 9, 2002.

32. Exhibit A2 provides a partial list of RICO Predicate Acts by IBM, SAP, JPMorgan and additional background. Exhibit C1 is a partial list of Documented Retaliations which Plaintiff had suffered *prior* to the date on which this federal case was first filed (April 18, 2016.) Exhibit D1 is a subset of those Documented Retaliations which also qualify as one or more of the RICO Predicate Acts that are itemized at 18 U. S. C. §§ 1961(1)(B), (1)(D), and (5). Exhibit D2 is a true copy of the CPL Agreement of Eclipse code, which shows IBM-SAP collusion from the Eclipse website.

33. **Plaintiff now testifies that the partial list of acts and events now documented in Exhibits A2, C1, D1 and D2 constitutes probable cause for granting all relief requested *infra* in the instant COMPLAINT.**

34. Moreover, further acts and events occurred between April 1995 and April 2016 by IBM, which also qualify as RICO predicate acts that constitute *further* probable causes for all the relief requested *infra*.

35. For example, Plaintiff herein alleges that obstruction of justice did in fact occur *whenever* Plaintiff was deprived of specific relief from the federal district courts in Wilmington, Delaware and in San Francisco, California, in the Third Circuit, the Federal Circuit and the U.S. Supreme Court.

Acquisition and Maintenance of an Interest in and Control of an *Enterprise* Engaged in a *Pattern of Racketeering Activity*: 18 U.S.C. §§ 1961(5), 1962(b)

36. Plaintiff now re-alleges each and every allegation as set forth above, and hereby incorporates same by reference, as if all were set forth fully herein. Substance prevails over form.

THE IBM ECLIPSE FOUNDATION

37. At various times and places partially enumerated in Plaintiff's *documentary material*, Defendant and DOES 1-100 did acquire and/or maintain, directly or indirectly, an interest in or control of a RICO *enterprise* of individuals who were associated in fact and who did engage in, and whose activities did affect, interstate and foreign commerce, all in violation of 18 U.S.C. §§ 1961(4), (5), (9), and 1962(b).

38. During the ten (10) calendar years preceding January 31, 2016, Defendant and DOES 1-100 did cooperate jointly and severally in the commission of two (2) or more of the RICO

predicate acts that are itemized in the RICO laws at 18 U.S.C. §§ 1961(1)(A) and (B), and did so in violation of the RICO law at 18 U. S. C. 1962(b) (Prohibited activities).

39. Plaintiff further alleges that Defendant and DOES 1-100 did commit two (2) or more of the offenses itemized above in a manner which they calculated and premeditated intentionally to threaten continuity, *i.e.* a continuing threat of their respective *racketeering activities*, also in violation of the RICO law at 18 U. S. C. 1962(b) *supra*.

40. Pursuant to the original Statutes at Large, the RICO laws itemized above are to be *liberally* construed by this Court. Said construction rule was never codified in Title 18 of the United States Code, however. See 84 Stat. 947, Sec. 904, Oct. 15, 1970.

41. *Respondeat superior* (principal is liable for agents' misconduct: knowledge of, participation in, and benefit from a RICO enterprise).

COUNT III:

Conduct and Participation in a RICO *Enterprise* through a *Pattern of Racketeering Activity*: 18 U. S. C. §§ 1961(5), 1962(c)

42. Plaintiff now re-alleges each and every allegation as set forth above, and hereby incorporates same by reference, as if all were set forth fully herein. Substance prevails over form.

THE IBM ECLIPSE FOUNDATION

43. At various times and places partially enumerated in Plaintiff's *documentary material*, Defendant and DOES 1-100 did associate with a RICO *enterprise* of individuals who were associated in fact and who engaged in, and whose activities did affect, interstate and foreign commerce.

44. Likewise, Defendant and DOES 1-100 did conduct and/or participate, either directly or indirectly, in the conduct of the affairs of said RICO *enterprise* through a *pattern of racketeering activity*, all in violation of 18 U.S.C. §§ 1961(4), (5), (9), and 1962(c).

45. During the ten (10) calendar years preceding January 31, 2016, Defendant and DOES 1-100 did cooperate jointly and severally in the commission of two (2) or more of the RICO predicate acts that are itemized in the RICO laws at 18 U.S.C. §§ 1961(1)(A) and (B), and did so in violation of the RICO law at 18 U. S. C. 1962(c) (Prohibited activities).

46. Plaintiff further alleges that Defendant and DOES 1-100 did commit two (2) or more of the offenses itemized above in a manner which they calculated and premeditated intentionally to threaten continuity, *i.e.* a continuing threat of their respective *racketeering activities*, also in violation of the RICO law at 18 U. S. C. 1962(c) *supra*.

47. Pursuant to 84 Stat. 947, Sec. 904, Oct. 15, 1970, the original Statutes at Large, the RICO laws itemized above are to be *liberally* construed by this Court. Said construction rule was never codified in Title 18 of the United States Code, however. See 84 Stat. 947, Sec. 904, Oct. 15, 1970.

48. *Respondeat superior* (principal is liable for agents' misconduct: knowledge of, participation in, and benefit from a RICO enterprise).

COUNT IV:

Conspiracy to Engage in a *Pattern of Racketeering Activity*: 18 U.S.C. §§ 1961(5), 1962(d)

49. Plaintiff now re-alleges each and every allegation as set forth above, and hereby incorporates same by reference, as if all were set forth fully herein. Substance prevails over form.

THE IBM ECLIPSE FOUNDATION

50. At various times and places partially enumerated in Plaintiff's *documentary material*, Defendant and DOES 1-100 did conspire to acquire and maintain an interest in a RICO *enterprise* engaged in a *pattern of racketeering activity*, in violation of 18 U.S.C. §§ 1962(b) and (d).

51. At various times and places partially enumerated in Plaintiff's *documentary material*, Defendant and DOES 1-100 did also conspire to conduct and participate in said RICO *enterprise* through a *pattern of racketeering activity*, in violation of 18 U.S.C. §§ 1962(c) and (d). See also 18 U.S.C. §§ 1961(4), (5) and (9).

52. During the ten (10) calendar years preceding January 31, 2016, Defendant and DOES 1-100 did cooperate jointly and severally in the commission of two (2) or more of the predicate acts that are itemized at 18 U.S.C. §§ 1961(1)(A) and (B), in violation of 18 U. S. C. 1962(d).

53. Plaintiff further alleges that Defendant and DOES 1-100 did commit two (2) or more of the offenses itemized above in a manner which they calculated and premeditated intentionally to threaten continuity, *i.e.* a continuing threat of their respective *racketeering activities*, also in violation of 18 U.S. C. 1962(d) (Prohibited activities *supra*).

54. Pursuant to 84 Stat. 947, Sec. 904, Oct. 15, 1970, the RICO laws itemized above are to be *liberally* construed by this honorable Court. Said construction rule was never codified in Title 18 of the United States Code, however. *Respondeat superior* (as explained above).

PRAYER FOR RELIEF

54. WHEREFORE, Plaintiff asks this Court to enter judgment against Defendant IBM and against Defendant's subsidiaries, affiliates, agents, servants, employees and all persons in active concert or participation with them, in the amount of one billion dollars, based on the number of Web transactions per application displayed on a Web browser, as each of Defendant's and its

customers' web sites has an infinite number of applications displayed on a Web browser offered as an online service on the Web and an infinite number of transactions from said application(s), granting the following relief:

A. Enter judgment that IBM has infringed and continues to infringe the '506 patent;

B. Enter judgment that the '506 patent is valid and enforceable;

C. Enter a preliminary and permanent injunction restraining and enjoining IBM and its officers, agents, servants, employees, attorneys, and any persons in active concert or participation with them who receive actual notice of the order by personal service or otherwise, from any further manufacture, use, sales, offers to sell, or importations of any and all of the products identified above;

D. An award of damages adequate to compensate Plaintiff for the infringement that has occurred, together with prejudgment interest from the date infringement of the '506 Patent began, based on the number of Web transactions per application displayed on a Web browser per each of Defendant's website(s), as each web site has an infinite number of applications displayed on a Web browser offered as an online service on the Web and an infinite number of transactions, totaling to at least $1 billion;

E. An award to Plaintiff of all remedies available under 35 U.S.C. § 284, up to treble damages, pre-judgment and post-judgment interest and costs and all other remedies available under 35 U.S.C. § 284;

F. An award to Plaintiff of all remedies available under 35 U.S.C. § 285;

G. A permanent injunction under 35 U.S.C. § 283 prohibiting further infringement of the '506 Patent, and, in the alternative, in the event injunctive relief is not granted as requested by Plaintiff, an award of a compulsory future royalty, based on the number of Web transactions

per application displayed on a Web browser per each of Defendant's web sites, as each of the Defendant's web sites has an infinite number of applications displayed on a Web browser offered as an online service on the Web and an infinite number of transactions, totaling to at least $1 billion; and

H. Such other and further relief as this Court or a jury may deem proper and just; and

55. **And *Wherefore***, pursuant to the statutes at 18 U. S. C. 1964(a) and (c), Plaintiff requests judgment against Defendant and DOES 1-100 as follows:

ON COUNT II:

56. That this Court liberally construe the RICO laws and thereby find that Defendant and DOES 1-100, both jointly and severally, have acquired and maintained, both directly and indirectly, an interest in and/or control of a RICO *enterprise* of *persons* and of other individuals who were associated in fact, all of whom engaged in, and whose activities did affect, interstate and foreign commerce in violation of 18 U. S. C. 1962(b) (Prohibited activities).

57. That Defendant and DOES 1-100 and all their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from acquiring or maintaining, whether directly or indirectly, any interest in or control of any RICO *enterprise* of *persons*, or of other individuals associated in fact, who are engaged in, or whose activities do affect, interstate or foreign commerce.

58. That Defendant and DOES 1-100 and all of their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from committing any more predicate acts in furtherance of the RICO *enterprise* alleged in COUNT II *supra*.

59. That all Defendants be required to account for all gains, profits, and advantages derived from their several acts of *racketeering activity* in violation of 18 U. S. C. 1962(b) and from all other violation(s) of applicable State and federal law(s).

60. That judgment be entered for Plaintiff and against all Defendants for Plaintiff's actual damages, and for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(b), according to the best available proof.

61. That all Defendants pay to Plaintiff treble (triple) damages, under authority of 18 U. S. C. 1964(c), for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(b), according to the best available proof.

62. That all Defendants pay to Plaintiff all damages sustained by Plaintiff in consequence of Defendants' several violations of 18 U. S. C. 1962(b), according to the best available proof.

63. That all Defendants pay to Plaintiff her costs of the lawsuit incurred herein including, but not limited to, all necessary research, all non-judicial enforcement and all reasonable counsel's fees, at a minimum of $690.00 per hour worked (Plaintiff's standard professional rate at start of this action).

64. That all damages caused by all Defendants, and all gains, profits, and advantages derived by all Defendants, from their several acts of racketeering in violation of 18 U. S. C. 1962(b) and from all other violation(s) of applicable Federal, State and federal law(s), be deemed to be held in constructive trust, legally foreign with respect to the federal zone [*sic*], for the benefit of Plaintiff, Her heirs and assigns.

65. That Plaintiff have such other and further relief as this Court deems just and proper, under the circumstances of this action.

ON COUNT III:

66. That this Court liberally construe the RICO laws and thereby find that all Defendants have associated with a RICO *enterprise* of *persons* and of other individuals who were associated in fact, all of whom engaged in, and whose activities did affect, interstate and foreign commerce in violation of the RICO law18 U. S. C. 1962(c) (Prohibited activities).

67. That this Court liberally construe the RICO laws and thereby find that all Defendants have conducted and/or participated, directly or indirectly, in the affairs of said RICO *enterprise* through a *pattern of racketeering activity* in violation of the RICO laws at 18 U.S.C. §§ 1961(5) ("pattern" defined) and 1962(c) *supra*.

68. That all Defendants and all of their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from associating with any RICO *enterprise* of *persons*, or of other individuals associated in fact, who do engage in, or whose activities do affect, interstate and foreign commerce.

69. That all Defendants and all of their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from conducting or participating, either directly or indirectly, in the conduct of the affairs of any RICO *enterprise* through a *pattern of racketeering activity* in violation of the RICO laws at 18 U.S.C. §§ 1961(5) and 1962(c) *supra*.

70. That all Defendants and all of their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from committing any more predicate acts in furtherance of the RICO *enterprise* alleged in COUNT III *supra*.

71. That all Defendants be required to account for all gains, profits, and advantages derived from their several acts of racketeering in violation of 18 U. S. C. 1962(c) *supra* and from all other violation(s) of applicable Federal, State and federal law(s).

72. That judgment be entered for Plaintiff and against all Defendants for Plaintiff's actual damages, and for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(c) *supra*, according to the best available proof.

73. That all Defendants pay to Plaintiff treble (triple) damages, under authority of 18 U. S. C. 1964(c), for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(c) *supra*, according to the best available proof.

74. That all Defendants pay to Plaintiff all damages sustained by Plaintiff in consequence of Defendants' several violations of 18 U. S. C. 1962(c) *supra*, according to the best available proof.

75. That all Defendants pay to Plaintiff His costs of the lawsuit incurred herein including, but not limited to, all necessary research, all non-judicial enforcement and all reasonable counsel's fees, at a minimum of $690.00 per hour worked (Plaintiff's standard professional rate at start of this action).

76. That all damages caused by all Defendants, and all gains, profits, and advantages derived by all Defendants, from their several acts of racketeering in violation of 18 U. S. C. 1962(c) *supra* and from all other violation(s) of applicable Federal, State and federal law(s), be deemed to be held in constructive trust, legally foreign with respect to the federal zone [*sic*], for the benefit of Plaintiff, Her heirs and assigns.

77. That Plaintiff have such other and further relief as this Court deems just and proper, under the full range of relevant circumstances which have occasioned the instant action.

ON COUNT IV:

78. That this Court liberally construe the RICO laws and thereby find that all Defendants have conspired to acquire and maintain an interest in, and/or conspired to acquire and maintain control of, a RICO *enterprise* engaged in a *pattern of racketeering activity* in violation of 18 U.S.C. §§ 1961(5), 1962(b) and (d) *supra*.

79. That this Court liberally construe the RICO laws and thereby find that all Defendants have conspired to conduct and participate in said RICO *enterprise* through a *pattern of racketeering activity* in violation of 18 U.S.C. §§ 1961(5), 1962(c) and (d) *supra*.

80. That all Defendants and all their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from conspiring to acquire or maintain an interest in, or control of, any RICO *enterprise* that engages in a *pattern of racketeering activity* in violation of 18 U.S.C. §§ 1961(5), 1962 (b) and (d) *supra*.

81. That all Defendants and all their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from conspiring to conduct, participate in, or benefit in any manner from any RICO *enterprise* through a *pattern of racketeering activity* in violation of 18 U.S.C. §§ 1961(5), 1962(c) and (d) *supra*.

82. That all Defendants and all their directors, officers, employees, agents, servants and all other *persons* in active concert or in participation with them, be enjoined *temporarily* during pendency of this action, and *permanently* thereafter, from committing any more predicate acts in furtherance of the RICO *enterprise* alleged in COUNT IV *supra*.

83. That all Defendants be required to account for all gains, profits, and advantages derived from their several acts of racketeering in violation of 18 U. S. C. 1962(d) *supra* and from all other violation(s) of applicable State and federal law(s).

84. That judgment be entered for Plaintiff and against all Defendants for Plaintiff's actual damages, and for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(d) *supra*, according to the best available proof.

85. That all Defendants pay to Plaintiff treble (triple) damages, under authority of 18 U. S. C. 1964(c), for any gains, profits, or advantages attributable to all violations of 18 U. S. C. 1962(d) *supra*, according to the best available proof.

86. That all Defendants pay to Plaintiff all damages sustained by Plaintiff in consequence of Defendants' several violations of 18 U. S. C. 1962(d) *supra*, according to the best available proof.

87. That all Defendants pay to Plaintiff her costs of the lawsuit incurred herein including, but not limited to, all necessary research, all non-judicial enforcement, and all reasonable counsel's fees, at a minimum of $690.00 per hour worked (Plaintiff's standard professional rate at start of this action).

88. That all damages caused by all Defendants, and all gains, profits, and advantages derived by all Defendants, from their several acts of racketeering in violation of 18 U.S.C. 1962(d) *supra* and from all other violation(s) of applicable State and federal law(s), be deemed to be held in constructive trust, legally foreign with respect to the federal zone [*sic*], for the benefit of Plaintiff, Her heirs and assigns.

89. That Plaintiff have such other and further relief as this Court deems just and proper, under the full range of relevant circumstances which have occasioned the instant action.

JURY DEMAND

90. Plaintiff demands a trial by jury on all issues so triable.

LIST OF EXHIBITS

91. Pursuant to 18 U.S.C. 1961(9), Plaintiff now formally incorporates Her *documentary material* by reference to all of the following Exhibits, as if set forth fully here, to wit: Exhibits A1, B1, A2, C1, D1, D2, A–K and the Eclipse code version 2.0.1, which is available for download at www.eclipse.org, which incorporates the inventions of Dr. Arunachalam and inventions of other inventors, demonstrating a pattern of racketeering activity by Defendant IBM.

VERIFICATION

92. I, Dr. Lakshmi Arunachalam, Plaintiff in the above entitled action, hereby verify under penalty of perjury, under the laws of the United States of America, without the "United States" (federal government), that the above statement of facts and laws is true and correct, according to the best of My current information, knowledge, and belief, so help me God, pursuant to 28 U.S.C. 1746(1). See the Supremacy Clause in the Constitution for the United States of America, as lawfully amended (hereinafter "U. S. Constitution").

Dated: April 18, 2016

Signed: Lakshmi Arunachalam

Printed: Dr. Lakshmi Arunachalam

93. A certificate of mailing by Express Mail via the U.S. Post Office to the Clerk of the Court, United States Federal District Court for the District of Delaware, is attached, along with a money order for the filing fees and a cover sheet.

Dated: April 18, 2016

Respectfully submitted,

Lakshmi Arunachalam

Tel: 650 690 0995
laks22002@yahoo.com

Dr. Lakshmi Arunachalam
222 Stanford Ave, Menlo Park, CA 94025

Pro Se Plaintiff

List of Exhibits

Exhibit A1: U.S. Patent No. 7,340,506

Exhibit B1: US 7,340,506 C1, Inter partes Re-examination Certificate

Exhibit A2: A partial list of RICO Predicate Acts by IBM, SAP, JPMorgan and additional background.

Exhibit C1: A partial list of Documented Retaliations which Plaintiff had suffered *prior* to the date on which this federal case was first filed (April 18, 2016.)

Exhibit D1: A subset of those Documented Retaliations which also qualify as one or more of the RICO Predicate Acts that are itemized at 18 U. S. C. §§ 1961(1)(B), (1)(D), and (5).

Exhibit D2: CPL Agreement of Eclipse code, which shows IBM-SAP collusion from the Eclipse website. The documents in the Exhibit are true and accurate copies of files downloaded from www.eclipse.org on April 18, 2016: 2002-08-29 Common Public License (CPL) Version 0.5 http://www.eclipse.org/legal/cpl-v05.html ; 2004-09-02 Tentative IP Log for eclipse.platform, eclipse.jdt and eclipse.pde http://www.eclipse.org/projects/ip_log.php?projectid=eclipse.platform,eclipse.jdt,eclipse.pde ; and 2004-09-02 Eclipse CPL to EPL Transition Plan http://www.eclipse.org/legal/cpl2epl/

Exhibit A: Judge William Alsup's Order in Case No. C 08-05149 WHA (N. Dt. CA) on February 17, 2009.

Exhibit B: April 5, 2016 Federal Circuit ("CAFC") Ruling in Case 14-1562, *Cardpool, Inc. v. Plastic Jungle, Inc.*

Exhibit C: Mandate issued on July 24, 2015 in CAFC Case No. 14-1495, *JPMorgan v. Dr. Arunachalam and Pi-Net International, Inc.*

Exhibit D: CAFC's Order denying *en banc* rehearing issued in June 2015 in CAFC Case No. 14-1495, *JPMorgan v. Dr. Arunachalam and Pi-Net International, Inc.*

Exhibit E: U.S. Supreme Court's Letter to CAFC on Order denying rehearing of Dr. Arunachalam's Petition for Writ of Certiorari in Case No. 15-691.

Exhibit F: Claims 14, 20 and 21 in U.S. Patent No. 7,340,506/US 7,340,506 C1.

Exhibit G: excerpts pp. 175-181, 189-191 of the prosecution history of the related U.S. Patent No. 6,212,556, the ('556) patent in the same priority chain as the '506 patent.

Exhibit H: excerpts pp 1-5 of the parent provisional patent application with S/N 60/006,634 filed November 13, 1995.

Exhibit **I:** excerpts pp 82-93 from the prosecution history of the parent U.S. Patent No. 5,778,178, the ('178) patent in the same priority chain as the '506 patent.

Exhibit J: is a true and correct copy of the web page for eclipse.org where Eclipse code is available for download including Plaintiff's inventions; list of members showing SAP, JPMorgan, IBM as members; board of directors showing SAP as a Board member; board meeting minutes of Dec 8, 2004 showing SAP's lead role; Eclipse awarded JPMorgan "Best Deployment of Eclipse Technology in an enterprise" at EclipseCon March 6, 2007; article entitled "JPMorgan raises the Bar for Banking Applications;" Amendment No. 8 to Form S-1 Registration statement for Facebook, Inc. showing JPMorgan, BofA, Barclays, Citigroup, Wells Fargo; and list of tutorials, sample code on Eclipse SOAP, REST, OData services from SAP.

Exhibit K: letter from SAP's counsel Greg Lanier to Dr. Arunachalam, terrorizing her on April 8, 2016.

Eclipse code version 2.0.1 is available for download at www.eclipse.org.

Exhibit A2: A partial list of RICO Predicate Acts by IBM, SAP, JPMorgan and additional background

Plaintiff Dr. Lakshmi Arunachalam's U.S. Patent No. 7,340,506/US 7,340,506 C1 ("'506 patent"), with a priority date of November 13, 1995, re-emerged successfully against Microsoft from an *inter-partes* re-examination by the USPTO. Judge Alsup ruled (**Exh. A**) against Microsoft, in Dr. Arunachalam's favor in Case No. C 08-05149 WHA (N. Dt. CA) on 2/17/09: "Microsoft is using counterfeit logic to manufacture a controversy where none exists."

35 U.S.C § 282 of the Patent Act allows the presumption of validity of her '506 patent. Defendant JPMorgan Chase & Co. did not provide clear and convincing evidence of invalidity of her patents, U.S. patent No. 5,987,500 ('500 patent), 8,037,158 ('158 patent) and 8,108,492 ('492 patent) in Case 1:12-cv-282 (D. Del) with completely different claims and specifications, different from the specification and claims of the '506 patent. SAP, Citizen's Financial Group, CitiBank, Wells Fargo Bank, JPMorgan Chase and Company and Kronos have not provided clear and convincing evidence of invalidity of the '506 patent.

A. SAP, Citizen's Financial Group, CitiBank, Wells Fargo Bank, JPMorgan Chase and Company and Kronos' (collectively "Delaware Defendants") arguments in 1:12-cv-355 (D. Del) are irrelevant to the facts of the '506 patent and contrary to April 5, 2016 Federal Circuit ("CAFC") Ruling (Exh. B) in Case 14-1562, *Cardpool, Inc. v. Plastic Jungle, Inc.* that "axed patent claims do not doom amended ones."

CAFC held that the validity of the new claims was not examined by any Court and that the district court's prior invalidity decision on *Cardpool Inc.*'s patent 7,494,048 was based on the prior set of claims and had **no effect on the new claims** granted upon reexamination:

> "district court's final judgment as to an original group of claims does not automatically render that judgment res judicata as to new claims granted upon reexamination."
>
> *"...Fresenius USA, Inc. v. Baxter Int'l, Inc.,* 721 F.3d 1330, 1346 (Fed. Cir. 2013)... CAFC held..."the statute requires that a final PTO decision affirmed by this court be given effect in pending infringement cases that are not yet final, and is not affected by a subsequent final court ruling contrary to the PTO ruling. *Cardpool* Dist. Dk. 93 at 1–2 (May 29, 2014)."... PTO's issuance of the Reexamination Certificate was an

> interpretation or application of federal law, and must be given retroactive effect because the infringement suit was still pending on appeal. Cardpool argues that the district court erred in law, because "the controlling interpretation of federal law must be given full retroactive effect in all cases still open on direct review and as to all events, regardless of whether such events predate or postdate our announcement of the rule." *Id.* (quoting *Harper v. Va. Dep't of Taxation,* 509 U.S. 86, 97 (1993)). CAFC "requires that this principle "applies with equal force where the change is made by an administrative agency acting pursuant to legislative authorization." *Thorpe v. Hous. Auth.,* 393 U.S. 268, 282 (1969)." "Cardpool also criticizes the district court for "fail[ing] to consider the case under the reexamined claims." Cardpool Br. 21. Cardpool states that the district court "committed legal error in not giving full effect to the reexamined amended claims...and by denying the motion to vacate without reconsideration of the basis in view of the amended reexamined claims." *Id.* at 22." "Cardpool...stated that "if the Court is inclined to apply its prior invalidity decision to the amended reexamined claims..., such a determination must not be done in a cursory manner but with a full opportunity of the parties to provide briefing and argument." *Cardpool* Dist. Dk. 93 at 5–6 (May 29, 2014)."

The validity and infringement of the re-examined claims in the '506 patent have not been evaluated by any court. The Delaware district court's initial unpatentability ruling in another case involving a completely different set of patents and different claims do not apply on the facts involving the '506 patent with new amended claims, because the claims that were the subject of the prior ruling on a different set of patents were different and do not exist "in the same form."

The "final PTO judgment" on reexamination of the '506 patent was issued before "the appellate mandate **(Exh. C)** that would have finalized the interim district court decision" on the '492, '500 and '158 patents, issued on July 24, 2015 in CAFC Case No. 14-1495, *JPMorgan*, a year after the PTO issued the '506 reexamination certificate. See **Exhs. D** and **E** - CAFC and U.S. Supreme Court Denial of Rehearing. There is no inequitable conduct or any non-disclosure, as alleged by Delaware Defendants, on the part of Dr. Arunachalam or her attorney Lawrence Goodwin, who is a highly experienced patent lawyer. This Court must grant Dr. Arunachalam her due process right to demonstrate that new and amended claims differ substantially from the claims already rejected by the Court in another case involving ***different patents***. The district court's decision was not final nor was it affirmed on appeal before the PTO's reexamination

decision. The District Court's decision was not affirmed by the CAFC, which dismissed the case ***without adjudicating on the merits of the case***. The district court's original decision is limited to the claims and grounds that existed in that case related to the '500, '492 and '158 patents-in-suit, not on the '506 patent. CAFC cites *"Allard v. DeLorean*, 884 F.2d464, 466 (9th Cir. 1989)."

B. Courts must examine changed factual circumstances: On 4/5/16, CAFC stated in 14-1562:

> "Dismissal "with prejudice" operates as res judicata as to the same cause of action. 747 Am. Jur. 2d Judgments § 547. How this rule of finality would apply to changed circumstances depends on the factual circumstances of the specific situation. *See Lawlor v. Nat'l Screen Serv. Corp.*, 349 U.S. 322, 327–328 (1955) ("That both suits involved 'essentially the same course of wrongful conduct' is not decisive" of the applicability of the doctrine of res judicata and courts must examine factual circumstances, such as, for example, whether "new causes of action" or "substantial changes in scope" of wrongful conduct exist, in determining its applicability.) Res judicata does not automatically arise against unknown future situations. In *Aspex*, the court applied these principles to the facts of that case, recognizing…If the claim did not exist at the time of the earlier action, it could not have been asserted in that action and is not barred by res judicata." 672 F.3d at 1342; *see also Lawlor*, 349 U.S. at 328 (a prior judgment "cannot be given the effect of extinguishing claims which did not even then exist and which could not possibly have been sued upon in the previous case"). On the facts and procedures of this case, the issue of validity of the reexamined claims remains to be addressed in any future proceeding. In the initial proceeding the original claims were adjudicated only on grounds of subject matter eligibility under section 101. As in *Aspex*, the effect of a prior judgment rendered on specific issues as applied to the original claims, depends on the facts and issues of the reexamination, and invokes equity as well as law. 672 F.3d at 1341–1346."

> "A district court's denial of a motion to vacate its judgment, Fed. R. Civ. P. 60(b), is reviewed on the procedural standards of the regional circuit, while any aspects of the motion that are unique to patent law are reviewed in accordance with Federal Circuit law.*Univ. of W. Va. Bd. of Trs. v. VanVoorhies*, 342 F.3d 1290, 1294 (Fed. Cir. 2003); *Lazare Kaplan International, Inc. v. Photoscribe Technologies, Inc.*, 714 F.3d 1289 (Fed. Cir. 2013)…a district court's denial of a Rule 60(b) motion is reviewed for abuse of discretion. *United States v. Asarco Inc.*, 430 F.3d 972, 978 (9th Cir. 2005). In reviewing discretionary rulings, the Ninth Circuit determines whether the district court applied an incorrect legal rule or whether the district court's application of the law to the facts was "illogical, implausible, or without support in inferences that may be drawn from the facts in the record." *United States v. Hinkson*, 585 F.3d 1247, 1262 (9th Cir. 2009) (*en banc*)…(quoting *Anderson v. City of Bessemer City, N.C.*, 470 U.S. 564, 577 (1985)). The Supreme Court counsels that "vacatur must be decreed for those judgments whose review is . . . 'prevented through happenstance'—that is to say, where a controversy presented for review has 'become moot due to circumstances

> unattributable to any of the parties.'" *U.S. Bancorp Mortg. Co. v. Bonner Mall P'ship*, 513 U.S. 18, 23 (1994) (quoting *United States v. Munsingwear, Inc.*, 340 U.S. 36, 40 (1950)). ...remand so the district court can decide whether to vacate its judgment in light of 'the consequences and attendant hardships of dismissal or refusal to dismiss' and 'the competing values of finality of judgment and right to relitigation of unreviewed disputes.'" *Dilley v. Gunn*, 64 F.3d 1365, 1370–71 (9th Cir. 1995) (quoting *Ringsby Truck Lines, Inc. v. W. Conference of Teamsters*, 686 F.2d 720, 722 (9th Cir. 1982), and stating that "*Ringsby* is wholly consistent with the 'equitable tradition of vacatur' reflected in *U.S. Bancorp*.")." (Emphasis added)

C. The '506 patent is a completely different patent with a completely different specification and totally different claims (Exh. F) from the patents-in-suit previously asserted. A claim term cannot be construed stripped from the context of the total claim.

The claim term in the '506 patent, "value-added service network," is definite because the boundaries of the patent protection sought are clear. Older cases should be applied with care, according to the facts of each case. Prosecution history estoppel and disclaimer prevent the Court from ruling several terms indefinite, such as "value-added service network," "service network," "value-added network switch." The District Courts' and CAFC's errors were prejudicial. The Court must analyze claim terms in view of the specification from the perspective of those skilled in the relevant art since a particular term used in one patent or application may not have the same meaning when used in a different application. Medrad, Inc. v. MRI Devices Corp., 401 F.3d 1313, 1318, 74 USPQ2d 1184, 1188 (Fed. Cir. 2005). **"Value-added service network" is a term coined by the inventor, Dr. Arunachalam and can only take on that meaning ascribed to it by the inventor**. The PTAB interpreted this claim term. Definiteness of claim language must be analyzed, not in a vacuum, but in light of the content of the particular application disclosure; the teachings of the prior art; and the claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made. In reviewing a claim for compliance with 35 U.S.C. 112(b), **the Court must consider the claim as a whole** to determine whether the claim apprises one of ordinary skill in the art of its scope and,

therefore, serves the notice function required by 35 U.S.C. 112(b), by providing clear warning to others as to what constitutes infringement of the patent. See Solomon v. Kimberly-Clark Corp., 216 F.3d 1372, 1379, 55 USPQ2d 1279, 1283 (Fed. Cir. 2000); In re Larsen, No. 01-1092 (Fed. Cir. May 9, 2001) (unpublished) (The preamble of the Larsen claim recited only a hanger and a loop but the body of the claim positively recited a linear member. The court observed that the totality of all the limitations of the claim and their interaction with each other must be considered to ascertain the inventor's contribution to the art. Upon review of the claim in its entirety, the court concluded that the claim at issue apprises one of ordinary skill in the art of its scope and, therefore, serves the notice function required by 35 U.S.C. 112.) Examples of claim language which have been held to be indefinite set forth in MPEP § 2173.05(d) are fact specific and should not be applied as per se rules. CAFC provides guidance (emphasis added):

> "The Federal Circuit's decision in *Powell v. Home Depot,* App. No. 2010-1309 (Fed. Cir. Nov 14, 2011)... reminds one "the prior art cited in the prosecution history of a patent forms part of the intrinsic evidence for claim construction purposes," *Kumar v. Ovonic Battery Co.,* 351 F.3d 1364, 1368 (Fed. Cir. 2003) (citing *Tate Access Floors, Inc. v. Interface Architectural Res., Inc.,* 279 F.3d 1357, 1371-72 n.4 (Fed. Cir. 2002); *Vitronics Corp. v. Conceptronic, Inc.,* 90 F.3d 1576, 1582 (Fed. Cir. 1996))."
> "In a six-four *en banc* decision in *Lighting Ballast Control LLC v. Philips Electronics North Am. Corp.*, the Federal Circuit confirmed its practice of *de novo* claim construction review. Judge Newman stated:"Implementing the Supreme Court's decision in *Markman v. Westview Instruments, Inc.*, 517 U.S. 370 (1996) (*Markman II*), aff'g *Markman v. Westview Instruments, Inc.*, 52 F.3d 967 (Fed. Cir. 1995) (*en banc*) (*Markman I*), this court in *Cybor* held that patent claim construction receives *de novo* determination on appeal, that is, review for correctness as a matter of law. Such review is conducted on the administrative record and any additional information in the record of the district court, and is determined without deference to the ruling of the district court." "Given the Supreme Court guidance in *Markman II* that claim construction is "better suited to determination by a judge rather than a jury," Judge Newman saw three options for the appropriate standard of review:
> "The first, urged by Lighting Ballast, holds that "patent claim construction is most reasonably classified as a question of fact" and so should be reviewed only for clear error. The second, supported by the Solicitor General for the United States, holds that claim construction should be subject to a "hybrid of *de novo* review and deferential review," with "the factual aspects of claim construction to be reviewed on the clearly erroneous standard, while the final conclusion receives review as a matter of law." The

third is that *Cybor* is a "reasonable and correct" interpretation of *Markman II*, such that the practice of *de novo* claim construction review should be maintained."

"**Judge Lourie's Concurrence**..."It would hardly promote uniformity ... for us to bless a claim construction in one district court, based on that court's judging the credibility and demeanor of the expert witnesses in one case, when a different case might lead to a different result based on a different district judge's appraisal of different witnesses.

[C]laim construction is not a process that normally involves historical facts. It primarily involves reading the patent's written description as well as the prosecution history of the patent, and this court is quite as able to do that as any district court, sometimes better."

Judge O'Malley cites several law review articles for the proposition that

"[p]arties do not make claim drafting decisions based on the standard of review we apply to trial court claim constructions. Nor could they, given the panel-dependent nature of our own determinations."

"Claim construction disputes are very fact specific—patents do not follow a formulaic structure, or even contain oft repeated language. Claims are drafted, redrafted, and amended in ways intended to reflect and capture particular inventions in a particular field, to avoid very specific prior art, and to respond to the rejections of the unique patent examiner involved in the application process. It is rare that any two claims we review contain the same phrasing, and **even more rare that the context in which the phrasing is used would not alter the meaning of even almost identical words**.... Combining the uniqueness of each claim term to be reviewed with the variations in rationale employed by the divergent members of this court, provides little practical guidance regarding how any claim construction dispute might be resolved in this forum—and certainly not the uniform reliability of outcome with which the majority now credits our jurisprudence in this area...we know how to delve into the "very fact specific" record, to trace the prosecution history of a claim that was "drafted, redrafted, and amended," to understand the "particular inventions" and the distinguishing features from the "very specific prior art." It doesn't matter that the claim construction in one case is not likely to apply to a different case involving a different patent. What matters is that the body of case law under *Cybor* has given us a framework within which to apply the principles of claim construction in a predictable manner."

In Dr. Arunachalam's parent 6,212,556 ('556) patent prosecution history (**Exh. G**), the inventor, Dr. Arunachalam distinguished her invention over the cited art, U.S. Patent No. 5,828,666 ("Focsaneanu"). **Delaware Defendants omit** that prosecution history estoppel already has established that the term is not indefinite and relates to application layer network switches, not with a network layer switch; and that prior art is not only cited, but also discussed

in detail in the specification of the '506 patent. The claim language, disclosure in the written description, and the meaning to persons of ordinary skill are fact specific. CAFC states:

> "cited art as intrinsic evidence for purposes of claim construction....claims should be construed in view of the prosecution history's treatment of the prior art so as to determine what the applicant gave up in obtaining allowance of the claims...When prior art that sheds light on the meaning of a term is cited by the patentee, it can have particular value as a guide to proper construction of the term, because it may indicate not only the meaning of the term to persons skilled in the art, but also that the patentee intended to adopt that meaning." *Arthur A. Collins, Inc. v. Northern Telecom Ltd,* 216 F 3d. 1042 (Fed. Cir. 2000)."

Delaware Defendants omit that Dr. Arunachalam's priority provisional application S/N60/006,634 (**Exh H** pp. 4-5) distinguishes between a valued-added service network, from a facilities network, gives analogy with telephone service network, that physical poles and cables of a phone network is the facilities network, that the voice service network is the application network that delivers voice services, that voice is the value-added network service or VAN service.

> "...Web evolving as...medium for electronic commerce (EC), new value - added network (VAN) services are expected to emerge... simple telephone call is...well - known example of a value - added network service...telephone network has two different but interrelated aspects: In terms of its physical components, it is a "facilities network." In terms of the varieties of VAN services that it provides, it is a set of many "traffic networks", each representing a particular interconnection of facilities. Traffic is the flow of multi – media information through the network....consider, for example, a simple transaction of daily commerce, such as ordering and paying for pizza, or home banking, or payroll services for businesses from banks, offered as a VAN service. The Internet, like the telecommunications network, is a system of interconnected facilities that could carry traffic from a variety of EC services. From the perspective of its physical components, the "Facilities Network" for EC exists today...There is no direct access to the end user from the VAN service providers, such as a Bank. There are some missing elements needed to capture and control the end user environment. The "Traffic Network" is THE challenge." (Exh H pp 4-5)

Ethernet cord and OSI network layer router or switch (col. 5) are examples of a facilities network, which is a TCP/IP-based (cols. 5-6) network with physical hardware components. Example of a value-added service network over the Web is a Web banking application network.

Delaware Defendants omit that the specification (1) distinguished between the network layer vs application layer, (*see* cols. 4-5 and Fig. 3) which defines clearly the **metes and bounds** of what the structure is; (2) evidences that any ambiguity has been resolved by the specification disclosing a metric that distinguishes the value-added service network as an application network including the application displayed on a Web browser limitation and the distinction from a facilities network, which is a TCP/IP-based physical Internet or Web. *Halliburton Energy Servs.,* 514F.3d,1255-56,85USPQ2d,1663 "...quantitative metric (e.g...limitation as to a physical property) rather than a qualitative functional feature"); (3) provide[s] a formula for calculating a property "along with examples that meet the claim limitation and examples that do not;" (4) discloses a "value-added service network" which is an OSI layer 7 application network that includes an application displayed on a Web browser (**providing examples** of such a "value-added service network" **meeting the claim limitation**, eg, Web banking network, that includes a Web banking application displayed on a Web browser, Figs 6A,5D, 5C) and is distinct from a facilities network, an IP-based facilities network, which only goes up to layer 4 of the OSI model, such as the physical Internet and the Web. POSvc application is a term coined by the inventor and can only take on the meaning ascribed to it by the inventor and is not indefinite; (5) **provides examples that do not meet the claim limitation** as in cols. 5-6, of an IP-based facilities network as in col. 5, such as the Internet, Web... (*id.* 1256, 85 USPQ2d at 1663 (citing *Oakley, Inc. v. Sunglass Hut Int'l,* 316F.3d 1331,1341,65 USPQ2d1321,1326 (Fed. Cir. 2003)). "Dialing into the bank via a modem line" is an example of a facilities network;

> "...user **100**... **dialing into the bank via a modem line**. If user **100** is a Web user...no current mechanism for performing...real-time transaction with the bank, as illustrated in FIG. 4A ... bank...unable to be a true "Web merchant," namely a **merchant capable of providing complete transactional services on the Web.**" (col. 5)

(6) **provides** a general **guideline and examples sufficient** to **teach a person skilled in the art when the claim limitation was satisfied** (see *Marosi,* 710 F.2d at 803, 218 USPQ at 292);

(7) demonstrates that the **boundaries of the claim term in the claim as a whole are clear and precise**, upon primary inquiry as to whether the language leaves room for ambiguity or whether the boundaries are clear and precise.

The Delaware District Court construed "VAN service provider" as a provider of a POSvc application. The Court ***must*** construe "value-added service network" consistent with "VAN service provider" "value-added network" and "VAN service." PTAB ***erroneously*** (and frankly suspiciously given undisclosed litigant financial holdings by the judges) construed it as "a network on which services, other than underlying network communication services, are provided." Patent Owner ("PO") construed it as "an OSI application layer network running on top of a facilities network and that provides value-added network services (VAN services)." Prelim. Resp. 18. "**VAN Services**" are "applications displayed on a Web browser, that provides a value-add to the network," (eg, Web banking application is an example of a value-add to the network.) A "**facilities network**" is "an IP-based network with physical hardware components that provides underlying network communication services up to layer 4 of the OSI model." This construction for "service network," "Value-added Service Network" is consistent with PO's construction of VAN service provider and also the specification. PTAB construed it similarly, distinguishing between a facilities network (which provides the underlying network services from layers 1-4 of the OSI model) and a "service network," "Value-added Service Network" which provides the value-added services like Web banking, consistent with the specification (col. 6). PTAB acknowledges that a service network includes an Exchange which displays a Web page **505** that includes applications **510**. (col. 5): "Five components interact to provide this

service network functionality, namely, exchange, ...graphical user interface." The specification discloses that a necessary component of a service network or "Value-added Service Network" is an application displayed on a Web browser and that the service network or "Value-added Service Network" is an OSI application layer network running on top of a TCP/IP-based facilities network, such as the Web, the physical Internet, or email networks, as PTAB acknowledged. (cols. 5-6) The service network or "Value-added Service Network" delivers VAN services or applications displayed on a Web browser. (col. 9). PTAB acknowledged in IPR2013-00194, IPR2013-00195, CBM2013-00013 and CBM2014-00018 that a service other than an underlying service is an application like the Bank POSvc application. PTAB itself has defined what "value-add" means, that it is a "service other than an underlying service is an application like the Bank POSvc application." PTAB acknowledged what VAN services means. VAN service is a term coined by the inventor, just as POSvc application is a term coined by the inventor and can only take on the meaning ascribed to these terms in the specification or prosecution history by the inventor. Application service **704** and VAN service **704** are one and the same as disclosed in the specification. The specification at col. 2 discloses "application" or "service." So VAN services are applications displayed on a Web page or Web browser.

D. Delaware Defendants' willful omissions, obstruction of justice, allegations about Dr. Arunachalam and her patents and terrorizing Dr. Arunachalam (Exh. K) mask racketeering evident from SAP's founding role (2001) in the IBM Eclipse Foundation, hijacking Dr. Arunachalam's inventions that created the millennial generation (Exh. J: eclipse.org, members, Eclipse code which includes said inventions)

Delaware Defendants obstructed justice involving multiple parties thus denying Dr. Arunachalam a due process hearing, without giving a chance to be heard nor being given a fair chance and due process by the Courts, using counterfeit logic to manufacture false allegations about Dr. Arunachalam and her patents that masks violation of U.S. laws and misrepresentation

by individual lawyers, expert witnesses, judges, PTAB, enterprises and their employees, that has caused great personal and financial injury to Dr. Arunachalam.

SAP colluded with IBM to hijack and illegally distribute Dr. Arunachalam's invention to multiple IBM Eclipse Foundation members.

Dated: April 18, 2016

Respectfully submitted,

Lakshmi Arunachalam

Tel: 650 690 0995
laks22002@yahoo.com

Dr. Lakshmi Arunachalam
222 Stanford Ave, Menlo Park, CA 94025

Pro Se Plaintiff

Exhibit C1: A partial list of Documented Retaliations which Plaintiff had suffered *prior* to the date on which this federal case was first filed (April 18, 2016.)

- IBM and IBM's customer JPMorgan and SAP, Wells Fargo, CitiBank have been engaged in obstruction of justice; tampering with a witness, Marvin Sirbu by SAP, and Ms. Spielman by JPMorgan; interference with commerce, robbery and extortion; racketeering (the Hobbs Act);
- IBM had a scheme to defraud and defendant IBM's knowing participation in that scheme, as evidenced by The IBM Eclipse Foundation;
- IBM had a specific intent to defraud; See **Exhibit D2**.
- SAP, JPMorgan, Wells Fargo, CitiBank, FiServ, all of whom are members of the IBM Eclipse Foundation made false representation of material facts and made material omissions of facts; that they knew were false, that they made the material representation or omission with the intent to induce the plaintiff/judges to rely, action by the plaintiff/judges in reliance on the misrepresentation or omission, injury to the plaintiff as a result of such reliance;
- IBM and SAP and their customers, JPMorgan, CitiBank, Wells Fargo are engaged in monetary transactions in property derived from specified unlawful activity and interstate transportation of stolen property, by illegally distributing Eclipse code which includes Dr. Arunachalam's inventions, through the IBM Eclipse Foundation.
- IBM, SAP, JPMorgan have been engaged in a pattern of racketeering activity of at least two acts of racketeering activity and the last of which occurred within ten years after the commission of a prior act of racketeering activity and with the threat of continuing activity. The factor of continuity plus relationship combines to form a pattern. This is evident from the IBM Eclipse Foundation. This conduct forms a pattern as IBM and other

members of the IBM Eclipse Foundation embrace unlawful acts that have the same or similar purposes, results, participants, victims, or methods of commission, or are otherwise interrelated by distinguishing characteristics and are not isolated events. IBM, SAP and JPMorgan have been engaged in such unlawful activity during a closed period of repeated conduct and also engaged in past conduct that by its nature projects into the future with a threat of repetition.

- The enterprise is the IBM Eclipse Foundation. The persons who commit the predicate offenses are IBM, SAP, JPMorgan, the judges, individual lawyers, expert witnesses, and they are distinct from the "enterprise," the IBM Eclipse Foundation.
- '1961(4) "enterprise" includes any individual, partnership, corporation, association, or other legal entity, and any union or group of individuals associated in fact although not a legal entity. The enterprise is the **IBM Eclipse Foundation.**
- IBM does not disclose where the underlying code comes from, namely, Dr. Arunachalam and Mike McKibben and Leader Technologies, Inc. of Columbus, Ohio.

18 U.S.C. "1962(a) through (d) prohibit four types of relation-ships between a pattern of racketeering activity and an enterprise.

'1962(a)

> It shall be unlawful for any person who has received income, directly or indirectly, from a pattern of racketeering activity or to use or invest, directly or indirectly, any part of such income, or the proceeds of such income, in the acquisition of any interest in, or the establishment or operation of, any enterprise which is engaged in, or the activities of which affect, inter-state or foreign commerce.

- Section 1962(a) requires a nexus between the income or proceeds from the underlying criminal activity and the enterprise, for the essence of the violation is the use of the illegal income in the enterprise. **The IBM Eclipse Foundation is evidence of existence of such nexus**.

A sufficient nexus between the illicit income and the enterprise has been established with the evidence of the IBM Eclipse Foundation where:

- The deposit of income in one of the defendant's companies (in the form of bank loan proceeds which were obtained by fraud) coincided with a com-parable amount earned in the enterprise.[i]
- Substantial deposits of income in the enterprise were being made at the same time that defendant was engaged in illicit activity.[ii]

'1962 (b)

> It shall be unlawful for any person through a pattern of racketeering activity ... to acquire or maintain, directly or indirectly, any interest in or control of any enterprise which is engaged in, or the activities of which affect, inter-state or foreign commerce.

The majority of courts require a proprietary interest, such as ownership of stock, to establish an "interest" in an enterprise under Section 1962(b).[iii]

- Defendant who was serving as leasing agent and was a partner in a real estate venture defrauded his partners by mismanaging partner-ship property, allowing a co-defendant to acquire an interest in the partnership inexpensively. **The court rejected the '1962(a) claim because the "use of proceeds" element was missing, but upheld the claim under '1962(b) because the co-defendant**

promised that the defendant would remain as leasing agent once the co-defendant acquired the property—giving the defendant a sufficient "interest" in the enterprise. Note: This case takes an expansive view of "interest."[iv]

- '1962(b) liability was rejected in a churning case where the customer always retained the power to terminate the broker.[v]
- '1962(b) liability was upheld where an oil company injured its competitor by using undue influence to obtain oil at below market prices.[vi]
- **'1962(c)**
- It shall be unlawful for any person employed by or associated with any enterprise engaged in, or the activities of which affect, interstate or foreign commerce, to conduct or participate, directly or indirectly, in the conduct of such enterprise's affairs through a pattern of racketeering activity....
- '1962(c) focuses on the conduct of the defendant, IBM, not "enterprise," The IBM Eclipse Foundation

'1962(d)

It shall be unlawful for any person to conspire to violate any of the provisions of subsections (a), (b), or (c) of this section.

- A RICO conspiracy is composed of two agreements:

(1) An agreement to commit at least two predicate acts which form the pattern of racketeering activity; and

(2) An agreement to the conduct which violates subsection (a), (b) or (c) of '1962, e.g. an agreement to conduct or participate in the affairs of an enterprise (sub-section(c)).[vii]

- A RICO conspiracy generally involves two groups of people- the conspirators and the enterprise.
- Aiding and abetting liability has been imposed where, for each alleged predicate act, the defendant was associated with the wrongful conduct, participated with the intent to bring it about, and sought by his actions to make it succeed.[viii]

THE CIVIL CAUSE OF ACTION

- **'1964(c)**
- Any person injured in his business or property by reason of a violation of section 1962 of this chapter may sue therefore in any appropriate United States district court and shall recover threefold the damages he sustains and the cost of suit, including a reasonable attorneys' fee.
- Drawing on the Supreme Court's broad interpretation of the Commerce Clause in the U.S. Constitution, courts have held that virtually any business activity which involves the flow of goods or services in "commerce" affects interstate commerce.
- 1964(c) requires that the injury to business or property occur "by reason of' the RICO violation. **The injury to Dr. Arunachalam and her property occurred by reason of the RICO violation by IBM.**

- **Facts of IBM's Racketeering:**

- IBM signed NDA with Dr. Arunachalam and her companies as early as April 1995, in 2001, 2003 and also later.
- IBM negotiated with Dr. Arunachalam to joint venture with her on numerous occasions between 1994 and 2011.

- IBM provided office space to Dr. Arunachalam at IBM, Sunnyvale in 1994 and also at IBM, San Mateo, CA in 2003.
- IBM offered to joint venture with Dr. Arunachalam to promote her Web application products with which she was engaged in a pilot trial with France Telecom in 2001.
- IBM offered to buy Dr. Arunachalam's patent portfolio in 2006 for several million dollars.
- IBM copied Dr. Arunachalam's inventions, which are now part of the IBM Eclipse Foundation source code available for download at www. Eclipse.org (eg, **see Eclipse code version 2.0.1 that include Dr. Arunachalam's inventions**.)
- IBM has been engaged in a similar pattern of racketeering activity and copied the inventions of other inventors, for example, of Leader Technologies, Inc. of Columbus, Ohio and Michael McKibben, who is the inventor of the social networking Facebook web application, which is now part of the IBM Eclipse Foundation source code available for download at www. Eclipse.org (eg, **see Eclipse code version 2.0.1 that include Mike McKibben's inventions.**)
- **The Executive Branch of the U.S. Government played a very important founding role in the IBM Eclipse Foundation**.
- **SAP played a very important founding role in the IBM Eclipse Foundation**.
- All of the activity of the IBM Eclipse Foundation has gone on in stealth to such an extent that not many know of the Eclipse code.
- SAP, Citizen's Financial Group, CitiBank, Wells Fargo Bank, JPMorgan Chase and Company and Kronos' (collectively "Delaware Defendants") arguments are irrelevant to the facts of the '506 patent and contrary to April 5, 2016 Federal Circuit ("CAFC")

Ruling (Exh. B) in Case 14-1562, *Cardpool, Inc. v. Plastic Jungle, Inc.* that "axed patent claims do not doom amended ones."

- Delaware Defendants obstructed justice involving multiple parties thus denying Dr. Arunachalam a due process hearing, without giving a chance to be heard nor being given a fair chance and due process by the Courts, using "counterfeit logic" to manufacture false allegations about Dr. Arunachalam and her patents that masks violation of U.S. laws and misrepresentation by individual lawyers, expert witnesses, judges, PTAB, enterprises and their employees, that has caused great personal and financial injury to Dr. Arunachalam.

SAP colluded with IBM to hijack and illegally distribute Dr. Arunachalam's invention to multiple IBM Eclipse Foundation members.

IBM provided their internal patent counsel as the USPTO's Commissioner, Dave Kappos (who was one of the IBM Agreement Stewards since 2001 of Eclipse Common Public License Version 0.5, Sec. 7, paragraph 4, that was initially used by SAP and others; "The Agreement Steward reserves the right to publish new versions, including revisions of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement". IBM is the initial Agreement Steward.") commissioned to kill valuable patents by Dr. Arunachalam who invented Web applications on a Web browser and by Michael McKibben, who invented social networking Web application used by Facebook. This is evident from the fact that even though Michael McKibben won the Markman Hearing in Delaware District Court and won three times at the USPTO in re-examinations, the Commissioner, Dave Kappos, initiated a re-exam against Michael McKibben's patents, unheard of in the history of the USPTO.

- IBM and the U.S. Government ensured that Dr. Arunachalam's Web application patents get killed in the Delaware District Court by JPMorgan Chase and Company.
- The IBM Eclipse Foundation installed the Eclipse code at JPMorgan for Web banking applications as a showcase system and awarded JPMorgan as best of breed using Eclipse code that includes Dr. Arunachalam's patented inventions and technology. See **Exhibit J**.
- IBM and SAP held Board membership in the IBM Eclipse Foundation Board and also held strategic roles managing the IP in the IBM Eclipse Foundation. **Exhibit J**
- Six months earlier in 2001 about the same time that the IBM Eclipse Foundation was formed, Judge Sue Robinson of the Delaware District Court and CAFC's Jan Horbaly Clerk of Court and Court Executive, and close associate of the IBM Eclipse Agreement Stewards participated in decisions in the Judicial Conference and re-defined the term "financial interest" away from industry standard set by the IRS and SEC and public accounting standards, to benefit judges to hide stock behind a thin veil of mutual funds and not recuse.
- Facebook's underwriters were JPMorgan Chase and Company, Wells Fargo, Citi Bank, and other Dr. Arunachalam litigants. **Exhibit J**.
- IBM, SAP's key customer is JPMorgan Chase and Company and they ensured that the judges in the Delaware District Court and CAFC and the U.S. Supreme Court did not allow Dr. Arunachalam to be heard, even though JPMorgan Chase and Company did not provide clear and convincing evidence of invalidity of the '500, '158 and '492 patents, contrary to the 35 U.S.C. Section 282 of the Patent Act.
- SAP's external counsel, Jon Strang did a clerkship under CAFC Judge Kimberly Moore and was lead counsel for SAP from Sterne Kessler at the CAFC against the inventor.

Jonathan Strang | Sterne, Kessler, Goldstein & Fox

www.skgf.com/**jonstrang**

Sterne, Kessler, Goldstein & Fox

Mr. **Strang** is an associate in the **Sterne Kessler** Litigation Group specializing in patent ...

Mr. **Strang** re-joined the **firm** after clerking for the Honorable Kimberly Moore at the ...

Before **law** school, Mr. **Strang** served as an officer in the U.S. Navy .

- Dr. Arunachalam's need to attend to her health in medical distress is an "inalienable right," a fundamental and compelling interest, guaranteed by the Bill of Rights. CAFC abridged this right, causing medical injury to Dr. Arunachalam. CAFC dismissed the case without a hearing or an opening appeal brief, when *pro se* Dr. Arunachalam, a senior citizen with disabilities from illness, genuinely trying to meet court rules and deadlines, was in medical distress, to which the CAFC was notified. CAFC's dismissal did not advance a legitimate government interest. Where fundamental rights are infringed, strict scrutiny is the test and the challenged law is generally struck down. *Shapiro v. Thompson*, 394 U.S. 618 (1969); *Shaw v. Hunt*, 517 U.S. 899, 908 (1996); *Vacco v. Quill*, 521 U.S. 793, 799 (1997). CAFC's erratic and disparate treatment of Dr. Arunachalam are the hallmarks of invidious discrimination. *Romer v. Evans*, 517 U.S. 620, 631 (1996). CAFC infringed Dr. Arunachalam's liberty-based substantive due process. In such cases, the U.S. Supreme Court recognizes a non-textual "liberty" which then limits or voids laws limiting that liberty. *Roe v. Wade,* 410 U.S. 113 (1973) (right to choose to have or not have an abortion).
- Eight Justices of the U.S. Supreme Court, CAFC Panel Judges and Delaware District Court Judges have conflicts of interest (financial, relationship or

other) in a litigant, JPMorgan, per their own annual financial disclosure statements and SEC Edgar. They are precluded from ruling in Cases 15-691, 14-1495 and 1:12-cv-282, voiding *ab initio* all judgments. Delaware District Court Judges Robinson and Andrews had conflicts of interest in JPMorgan, when Judge Robinson issued the Markman ruling and judgment in favor of JPMorgan in May 2014. Dr. Arunachalam is guaranteed the protections of 28 U.S.C. §§ 455, 144 and Canons 2 and 3 and FRCP 60(d) and 60(b) which also give the Court the power to grant relief to a party from a judgment, yet she was denied these protections.

- CAFC's medical interference breached multiple laws, depriving Dr. Arunachalam of the protections of the Bill of Rights, fourteenth Amendment, 35 U.S.C. §282 of the Patent Act, Civil Rights Act, American Disabilities Act, FRCP Rule 60(b), 60(d).

- **Chief Justice Roberts set a precedent in recusing himself in *Microsoft Corp.* v. *i4i Limited Partnership*, 563 U.S. (2011), due to conflicts of interest, Microsoft holdings and his relationships to Microsoft counsel Theodore Olson and Thomas Hungar, Gibson Dunn & Crutcher LLP**

Microsoft is a Third Party Requester in Re-Examinations of Dr. Arunachalam's patents, in particular, the '506 patent. Justice Roberts also has JPMorgan holdings. He did not rely on safe harbor to sit on the *Microsoft* case, even though many of his mutual fund holdings contain Microsoft stock, just like Judge Andrews has admitted that many of his mutual funds hold JPMorgan stock. Judge Andrews admitted he bought JPMorgan stock during the pendency of the JPMorgan case 1:12-cv-282.

- **Judges have conflicts of interest in multiple litigants in Dr. Arunachalam's patent cases. Dr. Arunachalam is the inventor of Web applications displayed on a Web browser, like Web banking, social networking, in ubiquitous use.**

Dr. Arunachalam's patented inventions created the millennial generation and transformed the way we live, work and play.

A. Delaware District Court Judge Robinson set a precedent and recused in May 2015, immediately upon Dr. Arunachalam's motion to recuse (App. 83a) in Case 1:12-cv-282

Judge Robinson tainted the JPMorgan case with her conflicts of interest in re-defining "financial interests" contrary to industry accounting standards to suit judges. All rulings in Case No. 1:12-cv-282 are void and must be voided.

B. Judge Robinson and CAFC's Jan Horbaly participated in Judicial Conference policy decisions that re-defined "financial interests" to excuse judges from disclosing holdings in litigants behind a profoundly abused "safe harbor *concept*" writing and mutual fund veil, contrary to IRS, SEC and public accounting standards

The ordinary dictionary definition of "financial interest," and of the IRS, SEC and Business Judgment Rule trump any conflicting or ambiguous definition. Ambiguous definitions in law must be resolved by the superior, controlling definition.

Horbaly resigned soon after failing to docket Dr. Arunachalam s *amicus curiae* entries in *Leader Tech v. Facebook.*

C. Judge Robinson's definition of "financial interests" is being used to deny Dr. Arunachalam's motions to recuse across the board related to judge holdings in litigants JPMorgan, Wells Fargo, Citigroup, Bank of America, Microsoft, SAP

Judges beneficially enjoy profits and losses from these holdings and must pay taxes on those holdings to the IRS. Therefore, they have a very real JPMorgan financial interest, rendering them biased. These Judges have a financial interest, direct stock or mutual funds, in Dr. Arunachalam litigants, presided over Dr. Arunachalam's cases, relying upon Judge Robinson's definition of "financial interests," refusing to recuse.

Petitioner moved that Judge Robinson is the source of all refusals to recuse and must recuse. Judge Robinson recused in May 2015, thereby voiding her orders of May 2014.

D. Judge Robinson failed to disclose relationships among CAFC, Skadden Arps, JPMorgan attorneys Dan DeVito and Ed Tulin, JPMorgan, Judge Andrews, Mayer Brown LLP, Judge Stark, that bias her judgment

Judge Robinson is tainted by Judges Andrews/Stark's financial holdings in JPMorgan. Judge Andrews has relationship conflicts of interest from having worked at Mayer Brown. He presided over the case for over two years before handing it to Judge Robinson on April 9, 2014, one week before the Markman Hearing. She made an erroneous and biased Markman Ruling shortly thereafter, misled by JPMorgan's false evidence. Chief Judge Stark worked at Skadden Arps (JPMorgan's counsel) for many years before becoming a judge. Dan DeVito worked with Reines at Weil Gotschal, the latter's insider relationships at the CAFC triggered Chief Judge Rader's resignation. Weil Gotschal hired CAFC Judge Kimberly Moore as an expert witness in a patent case presided by Judge Robinson, making this conflict unseemly. Ed Tulin clerked before the Judges at the Delaware District Court. The District Court is completely tainted by these relationship conflicts of interest. The collusion caused great harm to Dr. Arunachalam.

E. The U.S. Constitution guarantees litigants unbiased judges, a fundamental right

This case must be heard by Judges who do not have financial holdings and relationships in the litigant(s).

F. District Court and CAFC Judges should have been disqualified under 28 U.S.C. §§455, 144, Canon 2, Canon 3(c)

Judge Andrews has financial and relationship conflicts of interest in JPMorgan, presided over the case between March 12, 2012 and April 9, 2014 and currently presides since May 15, 2015, instead of recusing. This creates the strong appearance of impropriety for which relief through disqualification is warranted, an endemic problem affecting multiple district and appellate courts. Judge Andrews improperly dismissed Dr. Arunachalam's patent cases: *Fulton Bank* (1:14-cv-490-RGA), *Dell* (1:08-cv-00132-RGA), *Fedex* (1:08-cv-00133-RGA)

cases, despite conflicts of interest, triggering Judge Laporte's improper dismissal of Dr. Arunachalam's *Fremont Bank* (1:15-cv-00023-EDL) case in the Northern District of California.

- The entire docket entries in Cases 1.12-cv-282 (D.Del) in the *JPMorgan* case, *Fulton Bank* (1:14-cv-490-RGA), *Dell* (1:08-cv-00132-RGA), *Fedex (*1:08-cv-00133-RGA), *Citizens* (1:12-cv-355) in D. Del, Dr. Arunachalam's *Fremont Bank* (1:15-cv-00023-EDL) case, *SAP*'s 4:13-cv-01248-PJH in the Northern District of California, the appeals and Petitions for Writ of Mandamus in the Third Circuit and Federal Circuit cases 14-1495, 16-110, all of the IPR, CBM Appeals in the CAFC Case Nos. 15-1424, -1429, -1869, 1433, and Fremont Bank case No in the CAFC 15-1831, and the IPR, CBM docket entries at the PTAB on Dr. Arunachalam's Patent Nos. 8,037,158; 5,987, 500; and 8,108,492 are **all incorporated by reference herein as if fully re-stated herein.**

Judge Andrews' holdings in JPMorgan include stock in: VWENX Vanguard Wellington Admiral with $1,347,496,000 in JPMorgan, the 3rd largest holding in the fund; BVCVX Fidelity Blue Chip Value Fund with $6,961,569,000 in JPMorgan, the 8th largest holding in the fund. A Vice President in BVCVX served as JPMorgan treasurer. Chief Judge Stark has multiple holdings in JPMorgan, detailed in *Leader Tech v. Facebook*, Case 2011-1366 (Fed. Cir. 2011), Renewed Motion for Leave to File Amicus Curiae Brief, July 27, 2012. JPMorgan was underwriter to Facebook. He holds stock in: FUSEX Fidelity Spartan 500 Index Inv with $896,713,000 in JPMorgan, their 10th largest holding; VINIX Vanguard Institutional Index with $ 2,190,882,000 in JPMorgan, their 10th largest holding. Judges' nondisclosure of these interests in JPMorgan does not avoid the appearance of impropriety. *See* 28 U.S.C. §455(c). *Porter v.*

Singletary, 49 F.3d 1483, 11th Cir '95. Judge Andrews admitted he acquired direct stock in JPMorgan during the pendency of the case.

Judge Andrews admitted he has JPMorgan holdings, that he worked at Mayer Brown, as per his Senate Confirmation Hearings, that Mayer Brown has longstanding relationships with JPMorgan, Wells Fargo, Citigroup, Bank of America and Fedex. (D.I. 120, p. 8, 1:12-cv-355-RGA). A prior judicial relationship with a major law firm has no statute of limitations with which to conclude that there is not a conflict. Conflicts are conflicts, no matter their age.

CAFC failed to declare mistrial or remand the case because the Delaware District Court and CAFC panel judges should have been disqualified under 28 U.S.C. § 455, Canons 2 and 3, but refused to recuse.

G. CAFC failed to provide impartial judges, dismissed the Appeal without an opening brief or a hearing, when *pro se* Dr. Arunachalam was in medical distress

CAFC's medical interference violated Dr. Arunachalam's liberty rights. *Memorial Hospital v. Maricopa County*, 415 U.S. 250, 254-55 (U. S. 1974).

H. JPMorgan did not provide "clear and convincing evidence" of patent invalidity required by 35 U.S.C. § 282 of the Patent Act, in the Delaware District Court Case

CAFC's dismissal prevented arguments on the merits and handed Dr. Arunachalam's valuable property to JPMorgan without justification.

JPMorgan willfully misled the court, with false arguments, out of context, defrauding the fact-finding process.

I. Mutual fund "safe harbor *concept*" developed by Judge Robinson is <u>not</u> a law, rule, advisory or even a guideline. Plain language of the Code of Conduct for Judges prevails over subsequent judicial interpretations

U.S. law prohibits inferior guidelines, rulings and opinions, especially ambiguous ones like the "safe harbor concept," from superseding well settled law and precedent.

"[J]udicial interpretations of a statute by reenactment cannot overcome the plain meaning of a statute. '...does not constitute an adoption of a previous administrative construction.'" *Demarest v. Manspeaker*, 498 U.S. 184, 603 (1991).

The U.S. Supreme court clearly stated that an advisory opinion, like the safe harbor concept, is "entitled only to some deference." *Christensen v. Harris County*, 529 US 576 (2000) at 587. The safe harbor concept was not "arrived at after... formal adjudication or notice-and-comment rulemaking... Interpretations such as those in opinion letters... agency manuals... lack the force of law— do not warrant Chevron-style deference." *Chevron U. S. A. Inc.* v. *Natural Resources Defense Council, Inc.*, 467 U. S. 837 (1984). The Guide to Judicial Policy, Vol. 2B, Ch. 2 does not contain the force of law, as does the Code of Conduct for U.S. Judges, Canon 2.

J. Composition of Mutual Fund is Critical

A mutual fund makes no money apart from the profits and losses of its underlying holdings. The statute requires disclosure of "every source of income." The *sources of income* in a mutual fund —the portfolio stocks and bonds—are the components of a mutual fund that should be disclosed, not merely the fund's name, to assess conflicts of interest. If mere disclosure of the *name* of the mutual fund were sufficient, then this judiciary policy would not be needed.

K. "Safe Harbor" is an ambiguous concept in the Advisory

Even the safe harbor caveat states "it is important for a judge to determine whether a particular proposed investment is a 'mutual or common fund' and, therefore, qualifies under the safe harbor provision of Canon 3C. This advisory statement is ambiguous since Canon 3C nowhere uses the term "safe harbor." Whether or not the judge complies with this ambiguous provision is itself ambiguous. The court cannot reject as frivolous Dr. Arunachalam's concern

for impartiality since even the judge cannot ascertain whether he or she is compliant with an ambiguous "safe harbor concept."

L. "Participates in the management of the fund" is ambiguous

Canon 3 (3)(c)(i) is ambiguous, since "financial interest" is not ambiguous anywhere else in law, except when applied to judges. In such situations in law, especially since the judge pays taxes on those holdings, it does not exempt judges from a normal and routine definition of "financial interest." To acknowledge that one must pay taxes on financial investments held in litigants, and still be permitted to preside over cases where decisions favorable to a litigant will benefit one's investments, stands the whole notion of judicial impartiality on its head.

To be aware of the portfolio holdings, and to leave one's money in that fund vs. another and reviewing the funds quarterly or semi-annual results, is to manage one's fund holding.

The Court must differentiate why holding mutual funds invested in JPMorgan securities would not be considered a "material fact."

II. PTAB Judges McNamara and Stephen Siu have conflicts of interest in Microsoft, JPMorgan, SAP and other Litigants in Dr. Arunachalam's Patent Re-examinations, voiding their rulings

Judge McNamara refused to recuse despite his direct stock holding in Microsoft and other conflicts of interest, denying electronic filing. Judge Siu's Microsoft conflicts preclude him from ruling on Microsoft's Re-exam against Dr. Arunachalam, voiding his ruling.

III. SEC/EDGAR Summary of Justices and Judges' Financial Holdings

SEC/Edgar summarize the materiality of JPMorgan holdings by Chief Justice Roberts, seven Justices, CAFC Panel Judges, Judges Andrews, Stark's mutual funds, for example:

Chief Justice Roberts holds Fidelity Contrafund:

How to look up a mutual fund portfolio at www.sec.gov:

Determine Ticker Symbol, e.g., Fidelity Contrafund: FCNTX.

Go to http://www.sec.gov/

Select "FILINGS | Company Filings Search" on drop down menu

Type "FCNTX" in Fast Search box ("Ticker or CIK")

Select "Documents" button for most recent "N-CSR"

Select "htm" file for the full report, "Type" column, "N-CSR"

URL for Fidelity Contrafund, FCNTX, N-CSR, Feb. 26, 2015, U.S. SEC:

http://www.sec.gov/Archives/edgar/data/24238/000070420715000083/conmain.htm
Click Ctrl F.
Type "JPMorgan" – there are 24 instances.

Fidelity Contra Fund- FCNTX, FCNKX is a sector fund in financial services with heavy emphasis on Dr. Arunachalam's litigants. The holdings are summarized at the SEC: at least $785.4M invested in JPMorgan; $20.4B in Banks and Financial Services; Microsoft $2.1B; M&T Bank, Visa, $2B; BofA $1.02B; Wells Fargo Bank $3.9B; Citigroup $696.2M; Fiserv $225.9M; Google $6.2B, Facebook $3.6B, Apple $3.7B, Berkshire Hathaway $5.5B, IT $28.6B; e-retailer litigants $25.3B; subtotaling to at least $104B in conflicts of interest of the Justices, CAFC panel Judges and Judges Andrews and Stark in this fund.

The Justices' own disclosure statements and SEC/Edgar evidence at least the appearance of impropriety, if not outright impropriety. They have JPMorgan and other litigant holdings in at least the following: Justice Breyer in Vanguard 500 Index Fund, direct stock in IBM, Lowes; Justices Alito and Kagan in Vanguard Total Stock Market Index Fund; Justice Alito in ishares S&P 500 Growth Fund, IVW; Justice Scalia in Vanguard, Wells Fargo Bank, PIMCO, Blackrock, Fidelity, Templeton, Schwab mutual funds; Justice Clarence Thomas in Capital Growth and Income Funds, CWGIX; (JPMorgan is the custodian of assets in his AEPGX and RERGX funds); Justice Sotomayor in Templeton Global Bond A Fund, TPINX; Nuveen NWQ

Large Cap Value A fund, NQCAX, (which has stock in JPMorgan $21.5B; Citigroup $28.5B; Wells Fargo Bank $19.7B); Columbia TRI LC Growth A Fund LEGAX; Blackrock GLB allocation FD class A fund, MDLOX.; Justice Ginsburg has multiple JPMorgan mutual funds, JPM Tax Aware Equity Fund, etc. Some Justices also have direct stock in litigants.

CAFC panel and/or Delaware District Court judges have:

MFS Value Fund – MEIAX has $1.56B in JPMorgan investments and-$1.12B in Wells Fargo. **Eaton Vance Large Cap Value EHSTX** has $135.2M in JP Morgan investments, $121.2M in BofA, $69.4M in Wells Fargo, $118.7M in CitiGroup. **American Growth Fund, AGTHX** has $612.1M invested in JPMorgan; $12B in Banks and Financial Services; Microsoft $1.9B; Wells Fargo Bank $641M; Citigroup $2.64B; Google, Facebook, Apple, Berkshire Hathaway, IT services $38.4B; e-retailer litigants $32B; with $90.6B in conflicts of interest of the Judges in this fund.

The judges know of these JPMorgan holdings in these funds from which they receive reports at least twice a year pursuant to SEC rules.[1]

[1] *See* SEC Final Rule: Shareholder Reports and Quarterly Portfolio Disclosure of Registered Management Investment Companies, Securities and Exchange Commission, 17 CFR Parts 210, 239, 249, 270, and 274, [Release Nos. 33-8393; 34-49333; IC-26372; File No. S7-51-02], RIN 3235-AG64 http://www.sec.gov/rules/final/33-8393.htm#IB

Exhibit D1: A subset of those Documented Retaliations which also qualify as one or more of the RICO Predicate Acts that are itemized at 18 U. S. C. §§ 1961(1)(B), (1)(D), and (5).

Same items as contained in Exhibits A2, C1 and D2.

Exhibit D2: These documents are true and accurate copies of files downloaded from www.eclipse.org on April 18, 2016

1. 2002-08-29 Common Public License (CPL) Version 0.5 http://www.eclipse.org/legal/cpl-v05.html
2. 2004-09-02 Tentative IP Log for eclipse.platform, eclipse.jdt and eclipse.pde http://www.eclipse.org/projects/ip_log.php?projectid=eclipse.platform,eclipse.jdt,eclipse.pde
3. 2004-09-02 Eclipse CPL to EPL Transition Plan http://www.eclipse.org/legal/cpl2epl/

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

DR. LAKSHMI ARUNACHALAM, **Plaintiff,** **v.** **INTERNATIONAL BUSINESS MACHINES CORPORATION AND DOES 1-100,** **Defendant(s).**	**C.A. No. _____** **COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS;** Date Filed: April 18, 2016 **JURY TRIAL DEMANDED** **18 U. S. C. 1961 et seq.;** **18 U. S. C. 1964** **(Civil RICO Remedies);**

DECLARATION OF DR. LAKSHMI ARUNACHALAM IN SUPPORT OF PLAINTIFF DR. LAKSHMI ARUNACHALAM'S COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS

I, LAKSHMI ARUNACHALAM, declare:

I am the inventor and assignee of the U.S. Patent No. 7,340,506/US 7,340,506 C1 that has re-emerged successfully from an *inter-partes* re-examination by the United States Patent and Trademark Office initiated by Microsoft, and also of the prior patents-in-suit in the JPMorgan case 1:12-cv-282 (D.Del.), all of which derive their priority date from my provisional patent application with S/N 60/006,634 filed November 13, 1995. I reside at 222 Stanford Avenue,

Menlo Park, CA 94025. I am *pro se* Plaintiff in the above-captioned action. I make this declaration based on personal knowledge and, if called upon to do so, could testify competently thereto.

1. Attached as **Exhibit A1** is a true and correct copy of Dr. Arunachalam's patent, U.S. Patent No. 7,340,506.

2. Attached as **Exhibit B1** is a true and correct copy of Inter Partes Re-examination Certificate for Dr. Arunachalam's re-examined and re-issued patent, U.S. Patent No. 7,340,506C1.

3. Attached as **Exhibit A2** is a true and correct copy of a partial list of RICO Predicate Acts by IBM, SAP, JPMorgan and additional background.

4. Attached as **Exhibit C1 is** a true and correct copy of a partial list of Documented Retaliations which Plaintiff had suffered *prior* to the date on which this federal case was first filed (April 18, 2016.)

5. Attached as **Exhibit D1 is** a true and correct copy of a subset of those Documented Retaliations which also qualify as one or more of the RICO Predicate Acts that are itemized at 18 U. S. C. §§ 1961(1)(B), (1)(D), and (5).

6. Attached as **Exhibit D2 is** a true and correct copy of CPL Agreement of Eclipse code, which shows IBM-SAP collusion from the Eclipse website. These documents are true and accurate copies of files downloaded from www.eclipse.org on April 18, 2016: 2002-08-29 Common Public License (CPL) Version 0.5 http://www.eclipse.org/legal/cpl-v05.html ; 2004-09-02 Tentative IP Log for eclipse.platform, eclipse.jdt and eclipse.pde http://www.eclipse.org/projects/ip_log.php?projectid=eclipse.platform,eclipse.jdt,eclipse.

pde ; and 2004-09-02 Eclipse CPL to EPL Transition Plan http://www.eclipse.org/legal/cpl2epl/

7. Attached as **Exhibit A** is a true and correct copy of Judge William Alsup's Order in Case No. C 08-05149 WHA (N. Dt. CA) on February 17, 2009.

8. Attached as **Exhibit B** is a true and correct copy of April 5, 2016 Federal Circuit ("CAFC") Ruling in Case 14-1562, *Cardpool, Inc. v. Plastic Jungle, Inc.*

9. Attached as **Exhibit C** is a true and correct copy of the Mandate issued on July 24, 2015 in CAFC Case No. 14-1495, *JPMorgan v. Dr. Arunachalam and Pi-Net International, Inc.*

10. Attached as **Exhibit D** is a true and correct copy of CAFC's Order denying *en banc* rehearing issued in June 2015 in CAFC Case No. 14-1495, *JPMorgan v. Dr. Arunachalam and Pi-Net International, Inc.*

11. Attached as **Exhibit E** is a true and correct copy of U.S. Supreme Court's Letter to CAFC on Order denying rehearing of Dr. Arunachalam's Petition for Writ of Certiorari in Case No. 15-691.

12. Attached as **Exhibit F** is a true and correct copy of Claims 14, 20 and 21 in U.S. Patent No. 7,340,506/US 7,340,506 C1.

13. Attached as **Exhibit G** is a true and correct copy of excerpts pp. 175-181, 189-191 of the prosecution history of the related U.S. Patent No. 6,212,556, the ('556) patent in the same priority chain as the '506 patent.

14. Attached as **Exhibit H** is a true and correct copy of pp 1-5 of the parent provisional patent application with S/N 60/006,634 filed November 13, 1995.

15. Attached as Exhibit **I** is a true and correct copy of excerpts pp 82-93 from the prosecution history of the parent U.S. Patent No. 5,778,178, the ('178) patent in the same priority chain as the '506 patent.

16. Attached as **Exhibit J** is a true and correct copy of the web page for eclipse.org where Eclipse code is available for download including Plaintiff's inventions; list of members showing SAP, JPMorgan, IBM as members; board of directors showing SAP as a Board member; board meeting minutes of Dec 8, 2004 showing SAP's lead role; Eclipse awarded JPMorgan "Best Deployment of Eclipse Technology in an enterprise" at EclipseCon March 6, 2007; article entitled "JPMorgan raises the Bar for Banking Applications;" Amendment No. 8 to Form S-1 Registration statement for Facebook, Inc. showing JPMorgan, BofA, Barclays, Citigroup, Wells Fargo; and list of tutorials, sample code on Eclipse SOAP, REST, OData services from SAP.

17. I also certify that that the eclipse code, all versions, including version 2.0.1 is available for download at www.eclipse.org.

18. Attached as **Exhibit K** is a true and correct copy of letter from SAP's counsel Greg Lanier to Dr. Arunachalam, terrorizing her on April 8, 2016.

I declare under the penalty of perjury under the laws of the United States and the State of California and Delaware that the foregoing is true and correct. Executed this 18th day of April, 2016 in Menlo Park, California.

Lakshmi Arunachalam

Dr. Lakshmi Arunachalam

222 Stanford Avenue
Menlo Park, CA 94025
650 690 0995, laks22002@yahoo.com

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

DR. LAKSHMI ARUNACHALAM, **Plaintiff,** **v.** **INTERNATIONAL BUSINESS MACHINES CORPORATION AND DOES 1-100,** **Defendant(s).**	**C.A. No. _____** **COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS;** Date Filed: April 18, 2016 **JURY TRIAL DEMANDED** **18 U. S. C. 1961 et seq.;** **18 U. S. C. 1964** **(Civil RICO Remedies);**

CERTIFICATE OF MAILING

I, Dr. Lakshmi Arunachalam, hereby certify that on April 18, 2016, I filed an original and one copy of the attached "COMPLAINT FOR PATENT INFRINGEMENT AND VERIFIED COMPLAINT FOR DECLARATORY AND INJUNCTIVE RELIEF AND DAMAGES FROM RACKETEERING, CONSPIRACY TO ENGAGE IN A PATTERN OF RACKETEERING ACTIVITY AND RELATED CLAIMS," Dr. Arunachalam's Declaration and Verification in support thereof, and Exhibits along with the cover sheet, requisite filing fees in the form of a money order and Form AO-440, Summons in a Civil Action with the Clerk of the Court, U.S. District Court for the District of Delaware by Express Mail via the U.S. Post Office by overnight delivery for filing and docketing in this case to:
Clerk of Court
U.S. District Court for the District of Delaware,
844 N. King Street, Unit 18,
Wilmington, DE 19801.

Lakshmi Arunachalam

DATED: April 18, 2016

Dr. Lakshmi Arunachalam
222 Stanford Avenue
Menlo Park, CA 94025
650 690 0995
laks22002@yahoo.com

---

[i] *1. Bryn Mar. Ltd v. Carlton Browne and Co., Inc.,* No. 82-0696-E (S.D. Cal. 1983).

[ii] *United States v. Cauble,* 706 F.2d 1322 (5th Cir.), *cert den.,* 465 U.S. 1005 (1983)

[iii] *See In re American Honda Motor Co., Inc. Dealerships Relations Litigation,* 941 F.Supp. 528, *555* (D. Md. 1996) (adopting *Moffat* 's reasoning that 1962(a) and (b) properly apply to activities in the nature of acquiring a proprietary stake in an enterprise, not simply obtaining some influence over discretionary activities); *Moffat Enterprises, Inc. v. Borden, Inc.* 763 F. Supp. 143 (W.D. Pa. 1990).

[iv] *State v. Nine Say. Accounts, 553* So. 2d 823 (La. 1989); *Guerro v. Katzen,* 571 F. Supp. 714 (D.C. Cir. 1983).

[v] *O'Brien v. Dean Witter Reynolds, Inc.* 1984 WL 608 (D. Ariz. 1984).

[vi] *Sutliff Inc. v. Donavan Companies, Inc.,* 727 F.2d 648 (7th Cir. 1984) (criticized on other grounds by *Rose v. Mony Life Ins.,* 82 F. Supp. 2d 920, 924 (N.D. Ill. 2000)).

[vii] *US. v. Campione,* 942 F.2d 429, 438 (7th Cir. 1991)

[viii] *In Re Sahien & Assoc., Inc. Securities Litigation,* 773 F. Supp. 342 (S.D. Fla. 1991). *See Jaguar Cars, Inc. v. Royal Oaks Motor Car Co.,* 46 F.3d 258 (3d. Cir. 1995) (holding that in order to prove aiding and abetting in predicate act party must show that the defendant alleged to have aided and abetted the act knew of the commission of the act and acted with intent to facilitate it).

Exhibit A1



(12) **United States Patent**
Arunachalam

(10) **Patent No.: US 7,340,506 B2**
(45) **Date of Patent: Mar. 4, 2008**

(54) **VALUE-ADDED NETWORK SWITCHING AND OBJECT ROUTING**

(75) Inventor: **Lakshmi Arunachalam**, Menlo Park, CA (US)

(73) Assignee: **WebXchange, Inc.**, Menlo Park, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 807 days.

(21) Appl. No.: **09/792,323**

(22) Filed: **Feb. 23, 2001**

(65) **Prior Publication Data**

US 2002/0007406 A1 Jan. 17, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/296,207, filed on Apr. 21, 1999, now Pat. No. 6,212,556.

(51) **Int. Cl.**
G06F 13/00 (2006.01)

(52) **U.S. Cl.** **709/219**; 709/203; 709/227; 719/313

(58) **Field of Classification Search** 709/203, 709/217, 219, 238, 245; 719/313, 315, 316
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,148,474	A	9/1992	Haralambopoulos et al.	379/111
5,239,662	A	8/1993	Danielson et al.	
5,329,619	A	7/1994	Page et al.	
5,347,632	A	9/1994	Filepp et al.	
5,442,771	A	8/1995	Filepp et al.	
5,455,903	A	10/1995	Jolissaint et al.	
5,491,800	A *	2/1996	Goldsmith et al.	709/221
5,537,464	A	7/1996	Lewis et al.	379/114
5,557,780	A	9/1996	Edwards et al.	395/500.48
5,577,251	A *	11/1996	Hamilton et al.	718/101
5,677,708	A	10/1997	Matthews, III et al.	
5,710,887	A	1/1998	Chelliah et al.	395/226

(Continued)

OTHER PUBLICATIONS

Davidson, Andrew, Coding with HTML forms: HTML goes interactive. (hpertext markup language)(Tutorial); Dr. Dobb's Journal; vol. 20, No. 6, Jun. 1995; 19 pages.

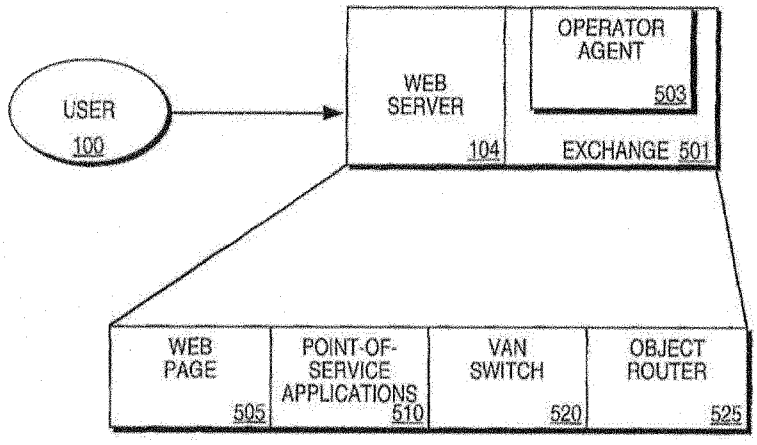
(Continued)

Primary Examiner—Viet D. Vu
(74) *Attorney, Agent, or Firm*—Trellis Intellectual Property Law Group, PC; Charles J. Kulas

(57) **ABSTRACT**

The present invention provides a method and apparatus for providing real-time, two-way transactional capabilities on the network. Specifically, one embodiment of the present invention discloses a configurable value-added network switch for enabling real-time transactions on the network. The configurable value added network switch comprises means for switching to a transactional application in response to a user specification from a network application, means for transmitting a transaction request from the transactional application, and means for processing the transaction request. Additionally, a method for enabling object routing is disclosed, comprising the steps of creating a virtual information store containing information entries and attributes associating each of the information entries and the attributes with an object identity, and assigning a unique network address to each of the object identities. Finally, a method is disclosed for enabling service management of the value-added network service, to perform OAM&P functions on the services network.

19 Claims, 29 Drawing Sheets



U.S. PATENT DOCUMENTS

5,715,314	A	2/1998	Payne et al.	380/24
5,724,424	A	3/1998	Gifford	380/24
5,771,354	A	6/1998	Crawford	
5,794,234	A	8/1998	Church et al.	707/4
5,812,779	A	9/1998	Ciscon et al.	
5,828,666	A	10/1998	Focsaneanu et al.	370/389
5,864,866	A	1/1999	Henckel et al.	707/103
5,870,473	A	2/1999	Boesch et al.	580/21
5,870,724	A	2/1999	Lawlor et al.	
5,878,403	A	3/1999	DeFrancesco et al.	705/38
5,892,821	A	4/1999	Turner	379/220
5,893,076	A	4/1999	Hafner et al.	705/28
5,901,228	A	5/1999	Crawford	
5,909,492	A	6/1999	Payne et al.	380/24
5,913,061	A	6/1999	Gupta et al.	395/680
5,956,509	A	9/1999	Kevner	
6,014,651	A	1/2000	Crawford	
6,049,785	A	4/2000	Gifford	705/39
6,049,819	A	4/2000	Buckle et al.	709/202
6,055,514	A	4/2000	Wren	
6,092,053	A	7/2000	Boesch et al.	705/26
6,101,527	A *	8/2000	Lejeune et al.	709/201
6,128,315	A	10/2000	Takeuchi	370/466
6,185,609	B1	2/2001	Rangarajan et al.	709/219
6,192,250	B1	2/2001	Buskens et al.	455/463
6,212,556	B1	4/2001	Arunachalam	
6,212,634	B1	4/2001	Geer, Jr. et al.	713/156
6,249,291	B1 *	6/2001	Popp et al.	345/473
6,327,579	B1	12/2001	Crawford	
6,411,943	B1	6/2002	Crawford	
6,553,427	B1 *	4/2003	Chang et al.	719/314
6,850,996	B2 *	2/2005	Wagner	710/33
7,080,051	B1	7/2006	Crawford	

OTHER PUBLICATIONS

iPIN Home Web Page [online].iPIN Interactive Transaction Services, Inc. 2000.[retrieved on May 23, 2001] Retrieved from the Internet:<URL:http//:www.ipin.com/ pp. 1.

iPIN Company Info Web Page [online]. iPIN Interactive Transaction Services, Inc. 2000.[retrieved on May 23, 2001] Retrieved from the Internet:<URL: http://www.ipin.com/01comp.html pp. 1-2.

The iPIN Approach, iPIN Products Web Page [online].iPIN Interactive Transaction Services, Inc. 2000. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.ipin.com/021prod.html pp1.

iPIN Technology, iPIN Products Web Page [online]. iPIN Interavtive Transaction Services Inc. 2000. [retrieved on May 23, 2001]. Retrieved from the Internet:<URL:http://www.ipin.com/01prod_tech.html pp. 1-2.

iPIN Solutions, iPIN Products Web Page [online]. iPin Interactvie Transction Services, Inc. 2000. [retrieved on May 23, 2001]. Retrieved from the Internet:<URL:http://www.ipin.com/02prod_solution.html pp. 1-2.

iPIN Service Options, iPIN Products Web Page [online]. iPIN Interactive Transaction Services, Inc. 2000.[retrieved on May 23, 2001]. Retrieved from the Internet:<URL:http://www.ipin.com/02prod_service.html pp. 1-2.

iPIN Partners Web Page[online]. iPIN Interactive Transaction Services, Inc. 2000. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.ipin.com/03part.html pp..

Server Operating System a Technology Overview, Microsoft Component Services. White Paper [online]. MicrosoftCOM. [retrieved on May 22, 2001]. Retrieved form the Internet:<URL:http://www.microsoft.com/com/wpaper/compsvscs.asp pp. 1-8.

Dr. Gui on Components, Com, and ATL. White Paper [online].2001 Microsoft Corporation. [retrieved on May 22, 2001] Retrieved from the Internet:<URL:http://www.msdn.microsoft.com/library/welcome/dsmsdn/msdn_drguion020298.htm pp. 1-61.

Dr. Gui's Gentle to COM Web Page [online]. Microsoft Corporation 2001.[retrieved on May 22, 2001] Retrieved from the Internet:<URL:http://www.microsoft.com/Com/news/drgui.asp pp. 1-3.

Christian Gross. Taking The Splash Diving into ISAPI Programming. White Paper [online].ISAPI Programming. Microsoft InteractiveDeveloper, Jan. 1997.[retrieved on May 22, 2001]. Retrieved from the Internet :<URL:http://www.Microsoft.com/mind/0197/isapi.htm pp. 1-10.

NSAPI Basics. Web page [online]. Netscape Communications Corporation, 1997. [retrieved on May 22, 2001.] Retrieved from the Internet:<URL:http://www.developer.netscape.com/docs/manuals/enterprise/nsapi/svrop.htm pp. 1-8.

The Common Gateway Interface. Web Page [online]. [retrieved on May 22, 2001]. Retrieved from the Internet:<URL:http://hoohoo.ncsa.uiuc.edu/cgi/primer.html pp. 1-4.

Open Market Content-Driven eBusiness Solutions. Web Page[online]. Open Market [retrieved on May 15, 2001] Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-3.

Open Market Content Server. Web Page[online]. Open Market. [retrieved on May 15, 2001] Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-4.

Open Market Content Centre. Web Page [online]. Open Market, Inc.-Enterprise Content Management and Delivery. [retrieved on May 15, 2001] Retrieved on the Internet:<URL:http:www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . .

Open Market Integration Centre. Web Page[online]. Open Market, Inc.- Enterprise Content Management & Delivery.[retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . .

Open Market Personilization Centre. Web Page [online]. Open Market, Inc.—Enterprise Content Management & Delivery.[retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . .

Open Market Catalog Centre. Wb Page [online. Open Market, Inc.—Enterprise Content Managnement & Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . .

Open Market Marketing Studio. Web Page [online]. Open Market Inc.—Enterprise Content Management Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-4.

Open Market Satellite Server. Web Page[online]. Open Market Inc.—Enterprise Content Management Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-4.

Open Market Commerce Products. Web Page[online]. Open Market Inc.—Enterprise Content Management Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-2.

Open Market Transact. Web Page[online]. Open Market Inc.—Enterprise Content Management Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/

AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-2.
Open Market ShopSite. Web Page[online]. Open Market Inc.—Enterprise Content Management Delivery. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/Xcelerate/Render&c=Artic . . . pp. 1-2.
Open Market ShopSite 5.0. Web Page[online]. [retrieved on May 15, 2001]. Retrieved on the Internet:<URL:http://www.openmarket.com/cgi-bin/gx.cgi/AppLogic+FTContentServer?pagename=FutureTense/Apps/.
Wireless Soultions, An Open Market eBusiness Solution Brief. WhitePaper. Open Market, Feb. 13, 2001.Forrester Research TechRankings™, Feb. 2001.
Portal Solutions, an Open Market eBusiness Solution Brief. White Paper. Open Market, Feb. 21, 2001. Forrester Research TechRankings™, Feb. 2001.
B2B Payment Services. Web Page[online]. CyberCash, 1996. [retrieved on May 23, 2001]. Retrieved from the Internet:<URL:http://www.cybercash.com pp. 1-2.
CyberCash Products. Web Page[online]. CyberCash, 1996. [retrieved on May 23, 2001]. Retrieved from the Internet:<URL:http://www.CyberCash.com/products/ pp. 1-2.
Cverify for Windows.2.5 Upgrade. Web Page[online]. CyberCash ICVERIFY 2.5 Upgrade,1996. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.Cybercash.com/icverify/upgrade.html pp. 1-2.
Cash Register Internet Payment Service. Web Page [online]. CyberCash Cash Register—Online Secure Payment Service. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/cashregister pp. 1-2.
CyberCash—Cash Register—How it Works. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/cashregister/howitworks pp. 1-3.
CyberCash—Industry Leading Features. Web Page [online]. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/cashregister/features.html pp. 1-4.
CyberCash- Why Choose CashRegister? Web Page[online]. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/cashregister/why.html pp. 1-.
CyberCash Cash Register—Online Secure Payment Service. CashRegister Demos. Web Page [online]. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.webdata.cybercash.com/demos/ pp. 1-2.
CyberCash WebAuthorize—Enterprise and Hosting Payment Processing. Web Page [online].[retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/webauthorize/ pp. 1-2.
CyberCash Corporate Information, CyberCash B2B Payment Services. Web Page [online].[retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/b2b/ pp. 1-2.
CyberCash FraudPatrol™ Service. Web Page [online]. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/fraudpatrol/ pp. 1-2.
CyberCash FraudPatrol—How It Works. Web Page [online]. [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/fraudpatrol/howitworks.html pp. 1-2.
CyberCash Affiliate Marketing Service. Web Page [online] [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/products/affiliatemarketing.html pp. 1-2.
CyberCash ICVERIFY—Features. Web Page [online] [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/icverify.html pp. 1-3.
CyberCash, Inc. -The E-Commerce Leader in Payment Processing PC Authorize. Web Page [online] [retrieved on May 23, 2001] Retrieved from the Internet:<URL:http://www.cybercash.com/pcauthorize/ pp. 1-2.
Hickey, "Shopping at Home: One Modem Line, No Waiting", Home PC, Dec. 1, 1994, p. 307, Dialog, File 647, Acc#01038162.
Lang, "Cashing In: The Rush is on to Buy and Sell on the Internet But . . . on Sidelines for Now", Advertising Age, Dec. 19, 1994, p. 11, Dialog, File 16, Acc# 05419137.
Lichty, Tom, "America Online Tour Guide", MacIntosh Edition, Version 2, Preface, Chapter 1, Ventana Press, 1992.
Lichty, Tom, "America Online Tour Guide", MacIntosh Edition, Version 2, Preface, Chapter 3, Ventana Press, 1992.
Lichty, Tom, "America Online Tour Guide", MacIntosh Edition, Version 2, Preface, Chapter 8, Ventana Press, 1992.
Lichty, Tom, "America Online Tour Guide", MacIntosh Edition, Version 2, Preface, Chapter 10, Ventana Press, 1992.
"Tymnet", Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/tymnet, May 2007.
Cox, Benjamin et al., "NetBill Security and Transaction Protocol", Carnegie Millon University, Pittsburgh, PA 15212-3890, undated.
Lamond, Keith, "Credit Card Transactions Real World and Online", http://www.virtualschool.edu/mon/ElectronProperty/klamond/credit_card.htm, pp. 1-16, 1996.
"HotJava", Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/HotJava. May 2007.
Banks, Michael A., "America Online: A Graphics-based Success" Link-Up, Jan./Feb. 1992.

* cited by examiner

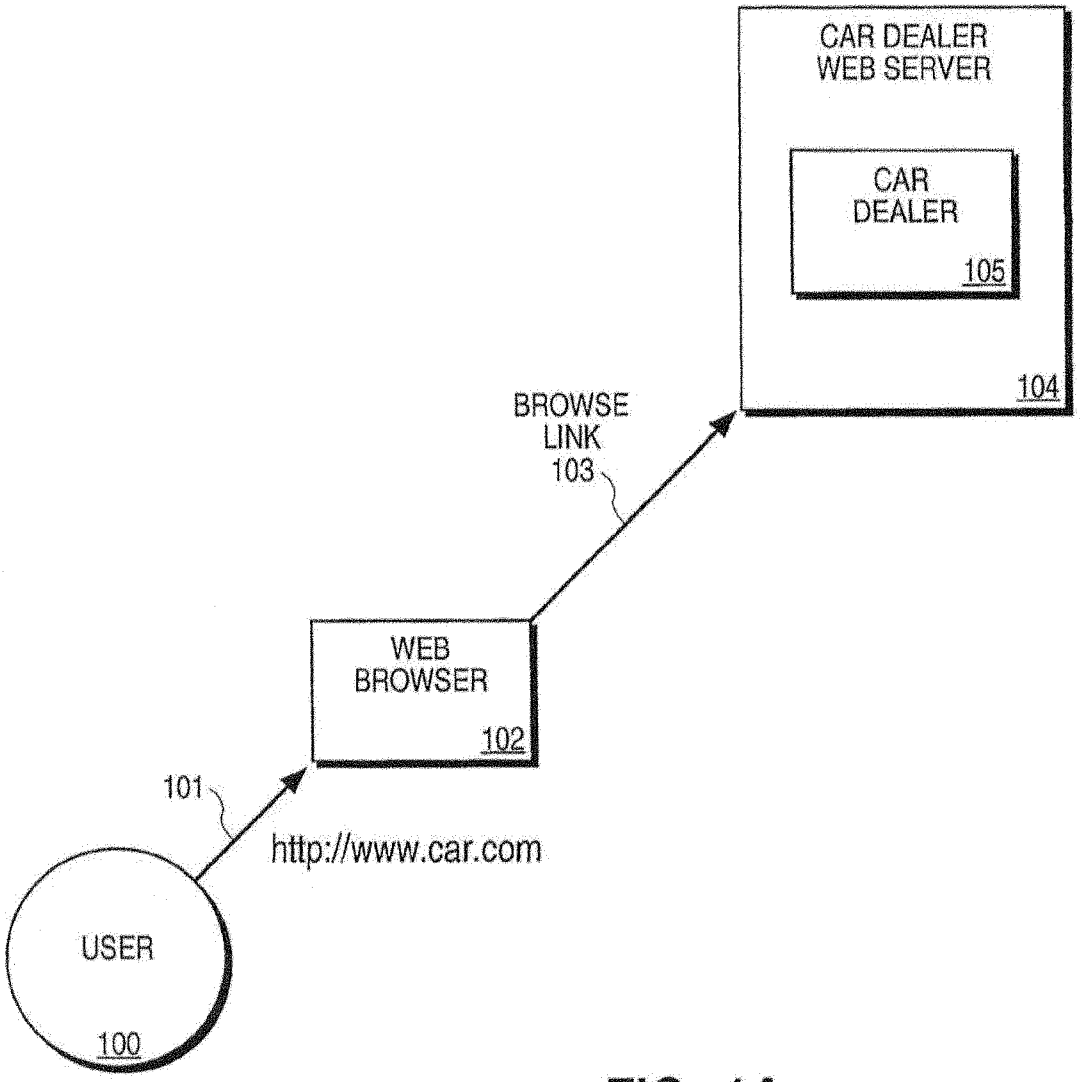


FIG. 1A
(PRIOR ART)

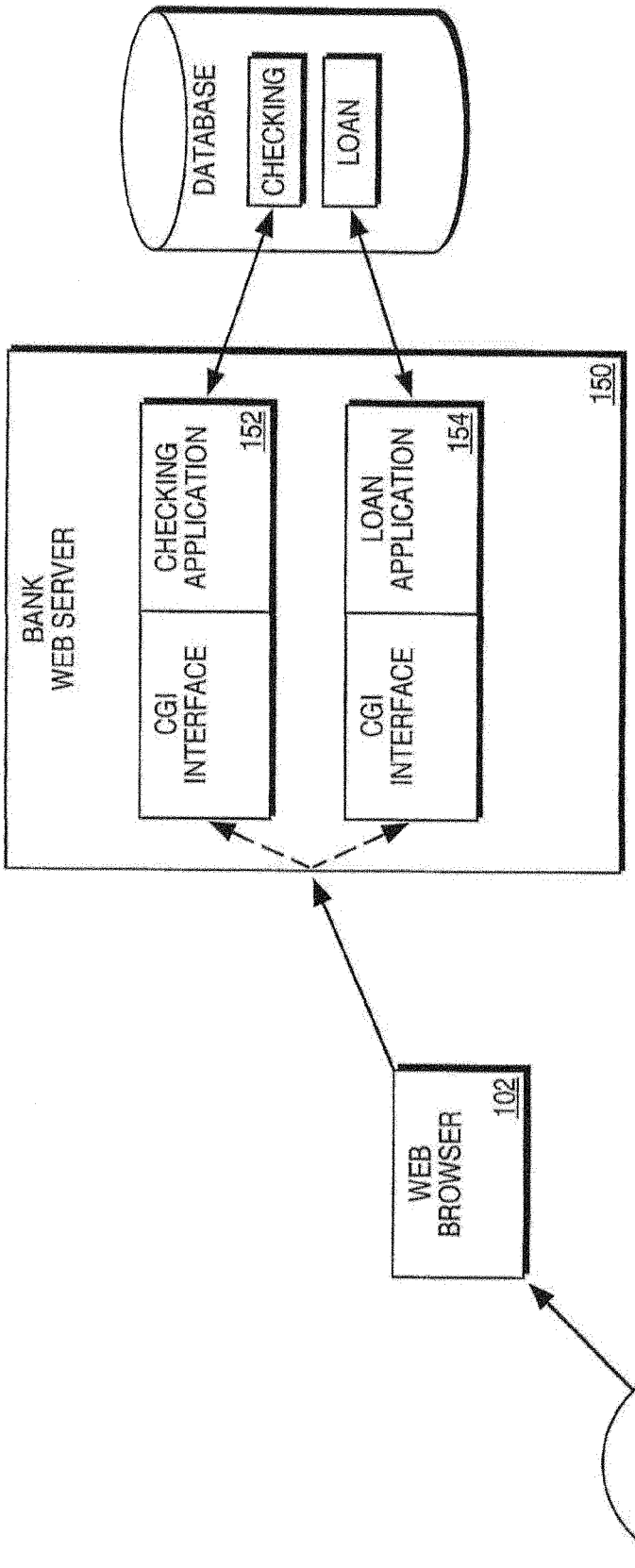


FIG. 1B
(PRIOR ART)

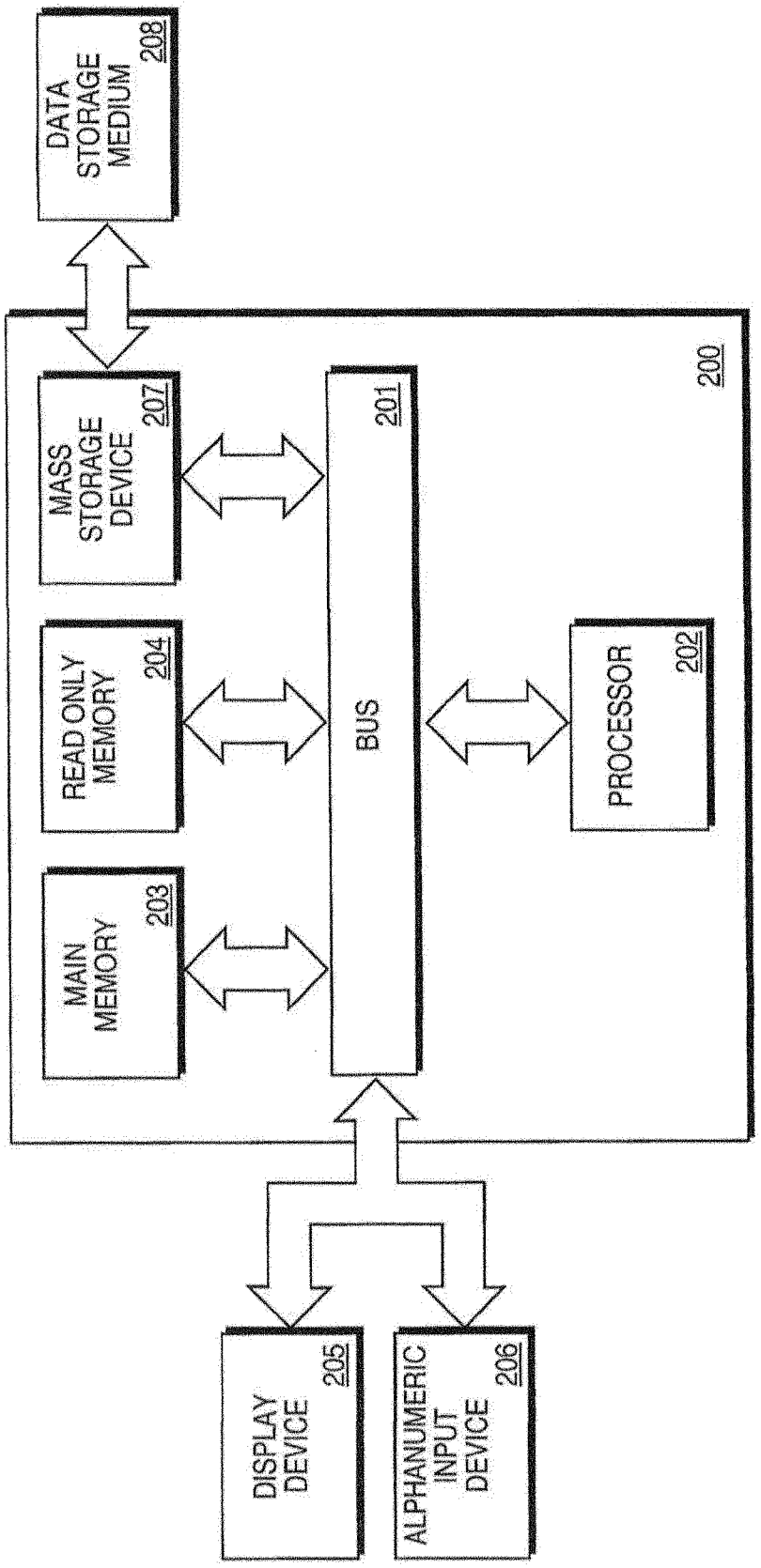

DATA STORAGE MEDIUM
208
200
MASS STORAGE DEVICE
207
201
READ ONLY MEMORY
204
BUS
PROCESSOR
202
MAIN MEMORY
203
DISPLAY DEVICE
205
ALPHANUMERIC INPUT DEVICE
206

FIG. 2

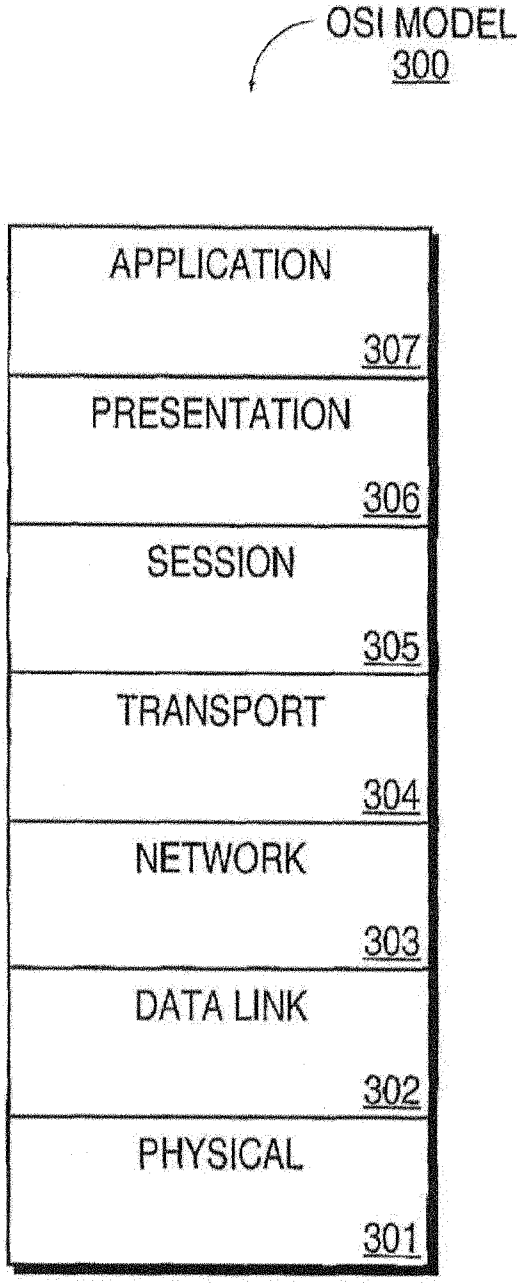

OSI MODEL
300
APPLICATION
307
PRESENTATION
306
SESSION
305
TRANSPORT
304
NETWORK
303
DATA LINK
302
PHYSICAL
301

FIG. 3

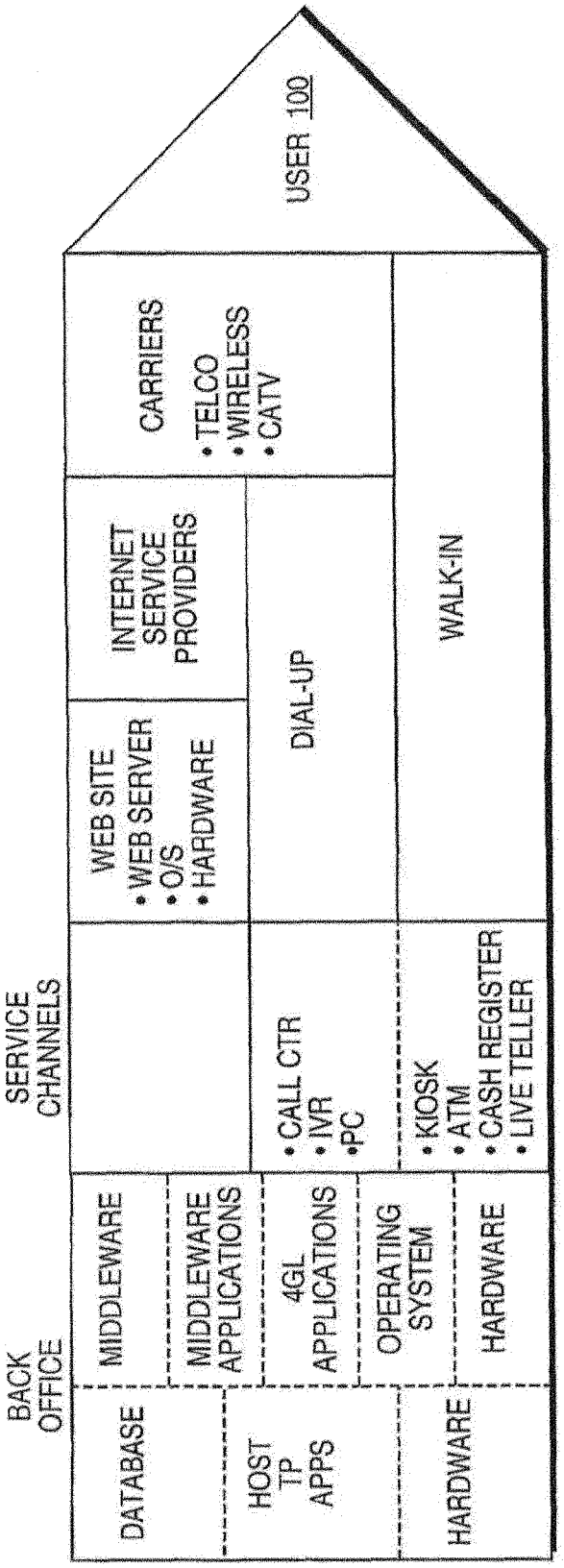

USER 100
CARRIERS
• TELCO
• WIRELESS
• CATV
INTERNET SERVICE PROVIDERS
WEB SITE
• WEB SERVER
• O/S
• HARDWARE
DIAL-UP
WALK-IN
SERVICE CHANNELS
• CALL CTR
• IVR
• PC
• KIOSK
• ATM
• CASH REGISTER
• LIVE TELLER
BACK OFFICE
MIDDLEWARE
MIDDLEWARE APPLICATIONS
4GL APPLICATIONS
OPERATING SYSTEM
HARDWARE
DATABASE
HOST TP APPS
HARDWARE

FIG. 4A

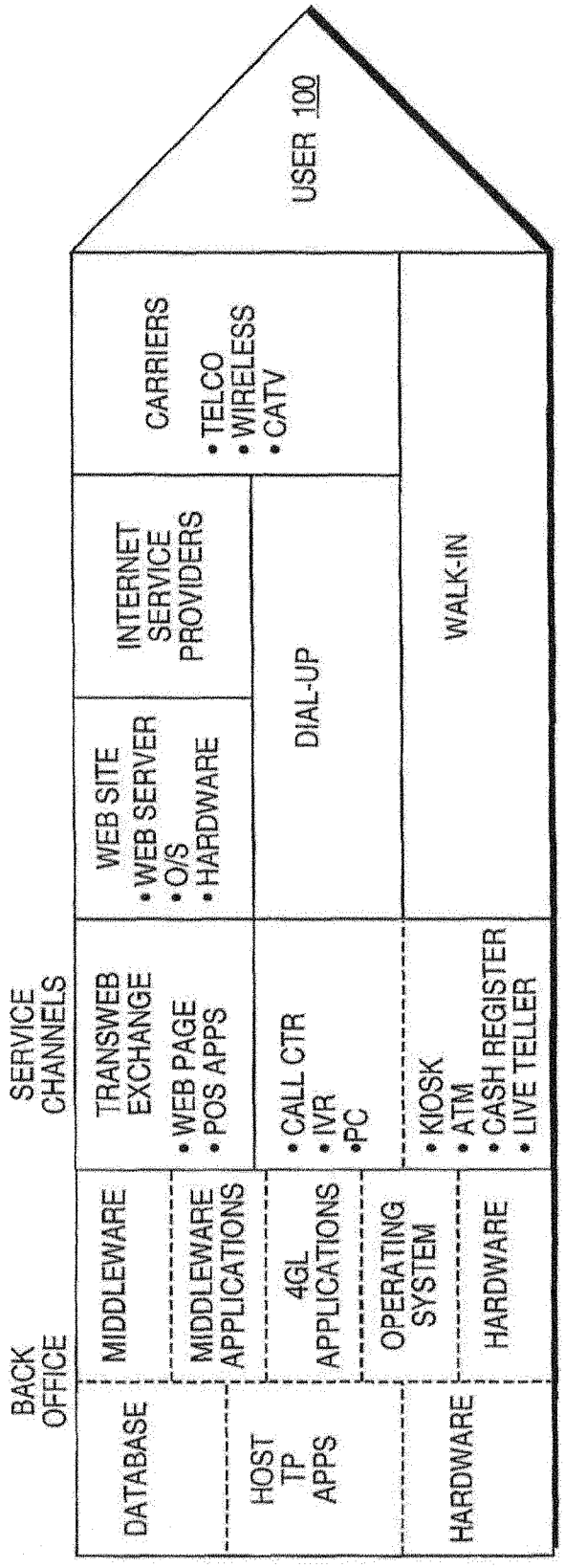

USER 100
CARRIERS
• TELCO
• WIRELESS
• CATV
INTERNET SERVICE PROVIDERS
WEB SITE
• WEB SERVER
• O/S
• HARDWARE
DIAL-UP
WALK-IN
SERVICE CHANNELS
TRANSWEB EXCHANGE
• WEB PAGE
• POS APPS
• CALL CTR
• IVR
• PC
• KIOSK
• ATM
• CASH REGISTER
• LIVE TELLER
BACK OFFICE
MIDDLEWARE
MIDDLEWARE APPLICATIONS
4GL APPLICATIONS
OPERATING SYSTEM
HARDWARE
DATABASE
HOST TP APPS
HARDWARE

FIG. 4B

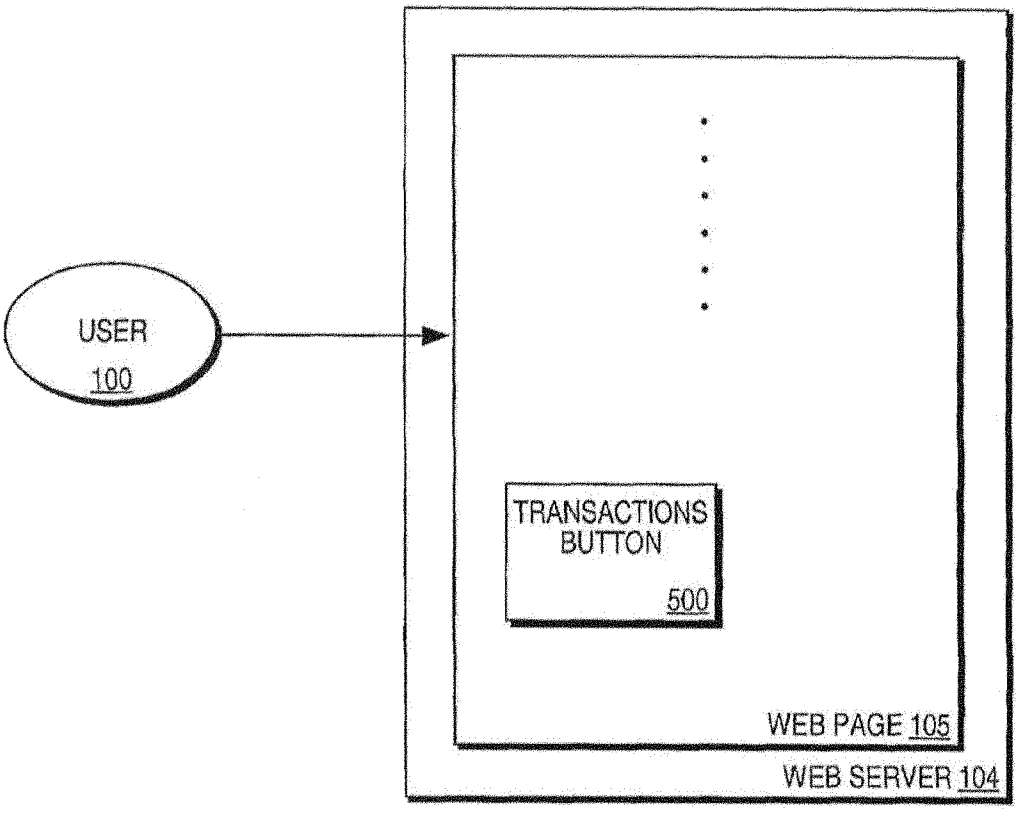

USER
100
TRANSACTIONS BUTTON
500
WEB PAGE 105
WEB SERVER 104

FIG. 5A

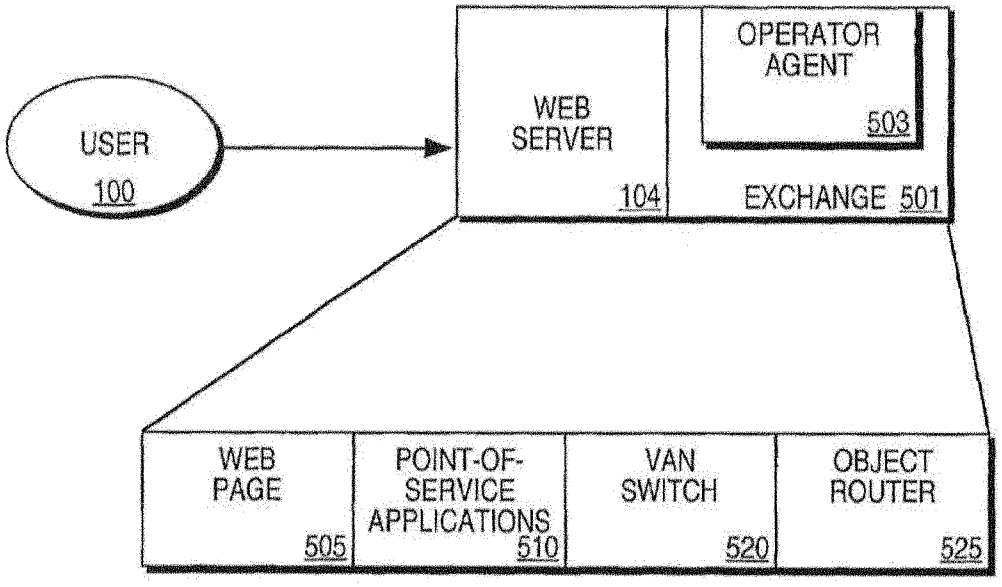

USER
100
WEB SERVER
104
OPERATOR AGENT
503
EXCHANGE 501
WEB PAGE
505
POINT-OF-SERVICE APPLICATIONS
510
VAN SWITCH
520
OBJECT ROUTER
525

FIG. 5B

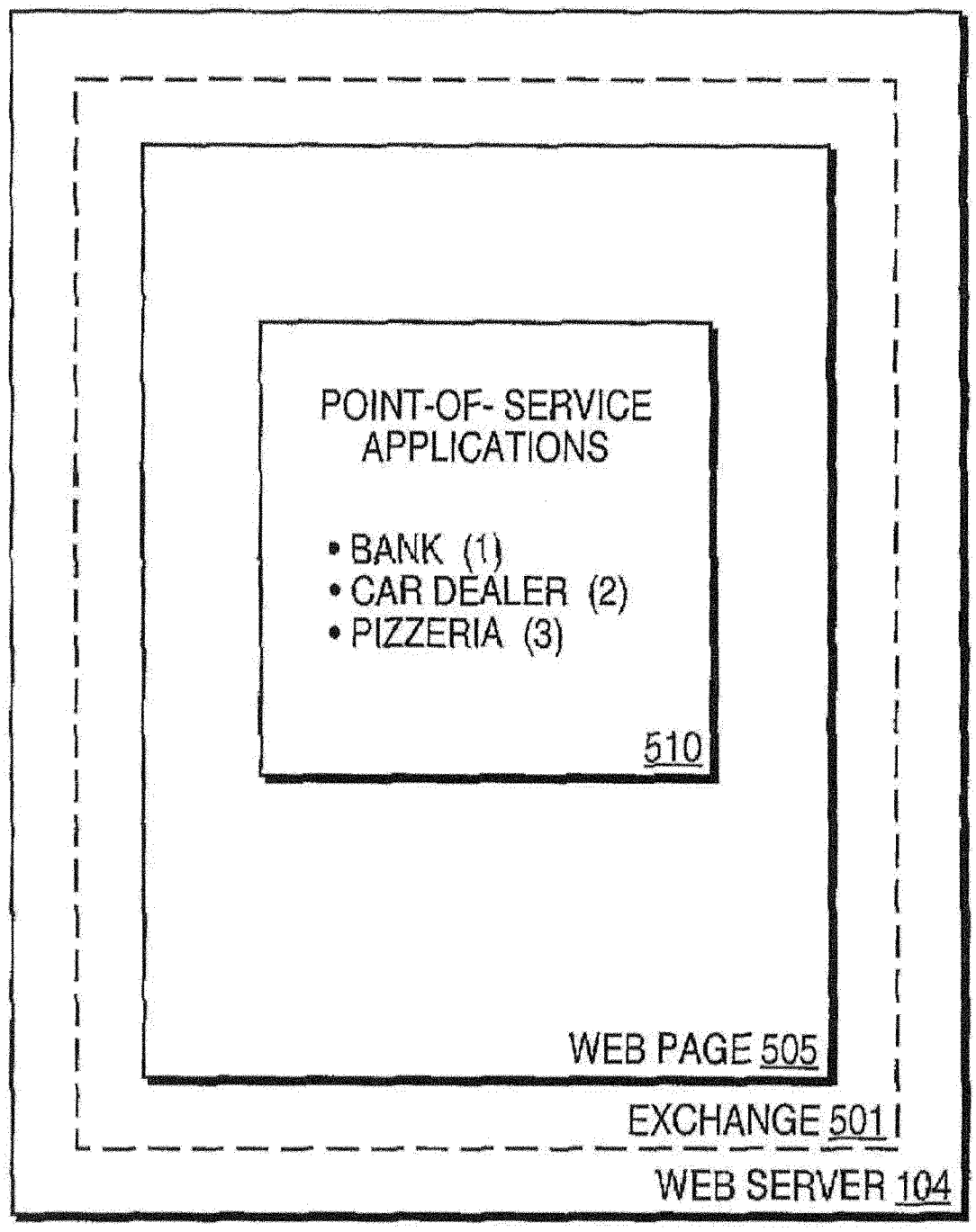

POINT-OF- SERVICE APPLICATIONS
• BANK (1)
• CAR DEALER (2)
• PIZZERIA (3)
510
WEB PAGE 505
EXCHANGE 501
WEB SERVER 104

FIG. 5C

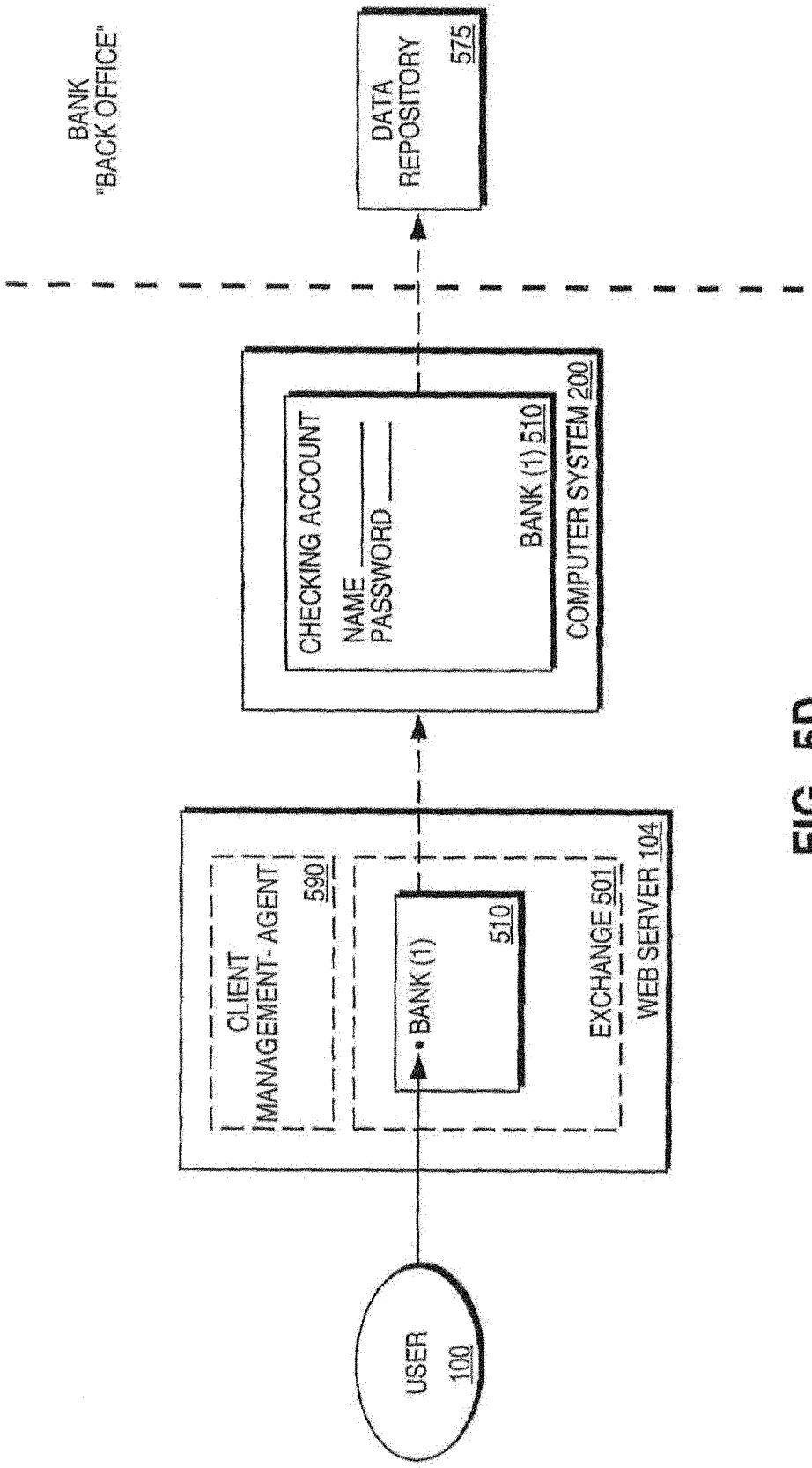

BANK
"BACK OFFICE"
DATA
REPOSITORY
575
CHECKING ACCOUNT
NAME
PASSWORD
BANK (1) 510
COMPUTER SYSTEM 200
CLIENT
MANAGEMENT- AGENT
590
BANK (1)
510
EXCHANGE 501
WEB SERVER 104
USER
100

FIG. 5D

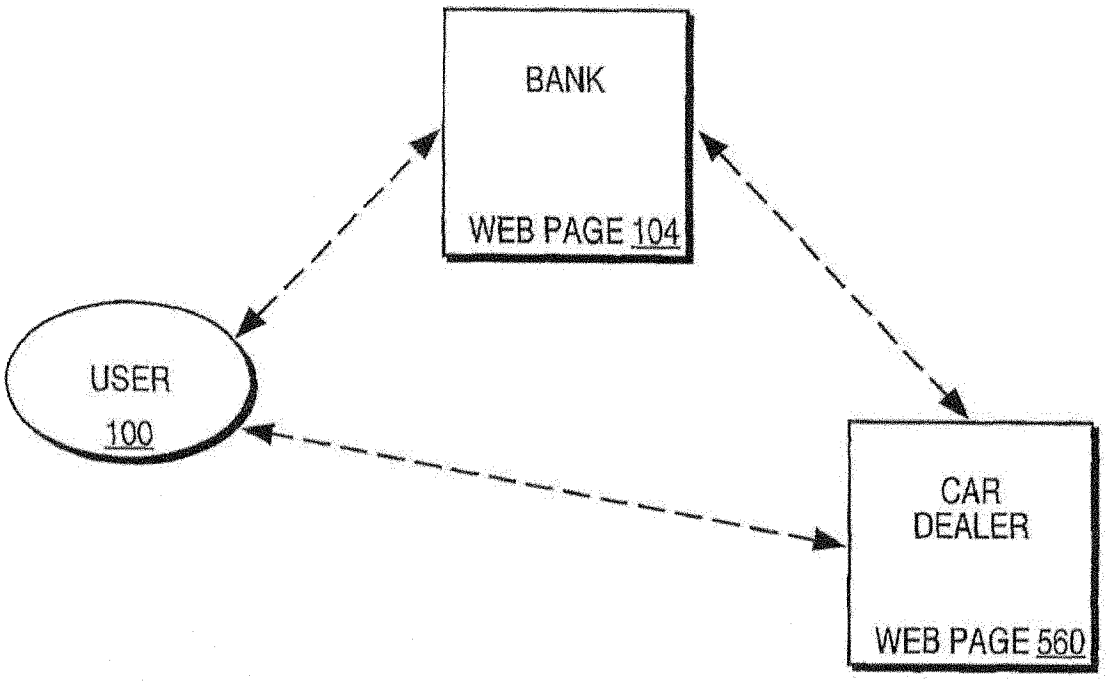


FIG. 5E

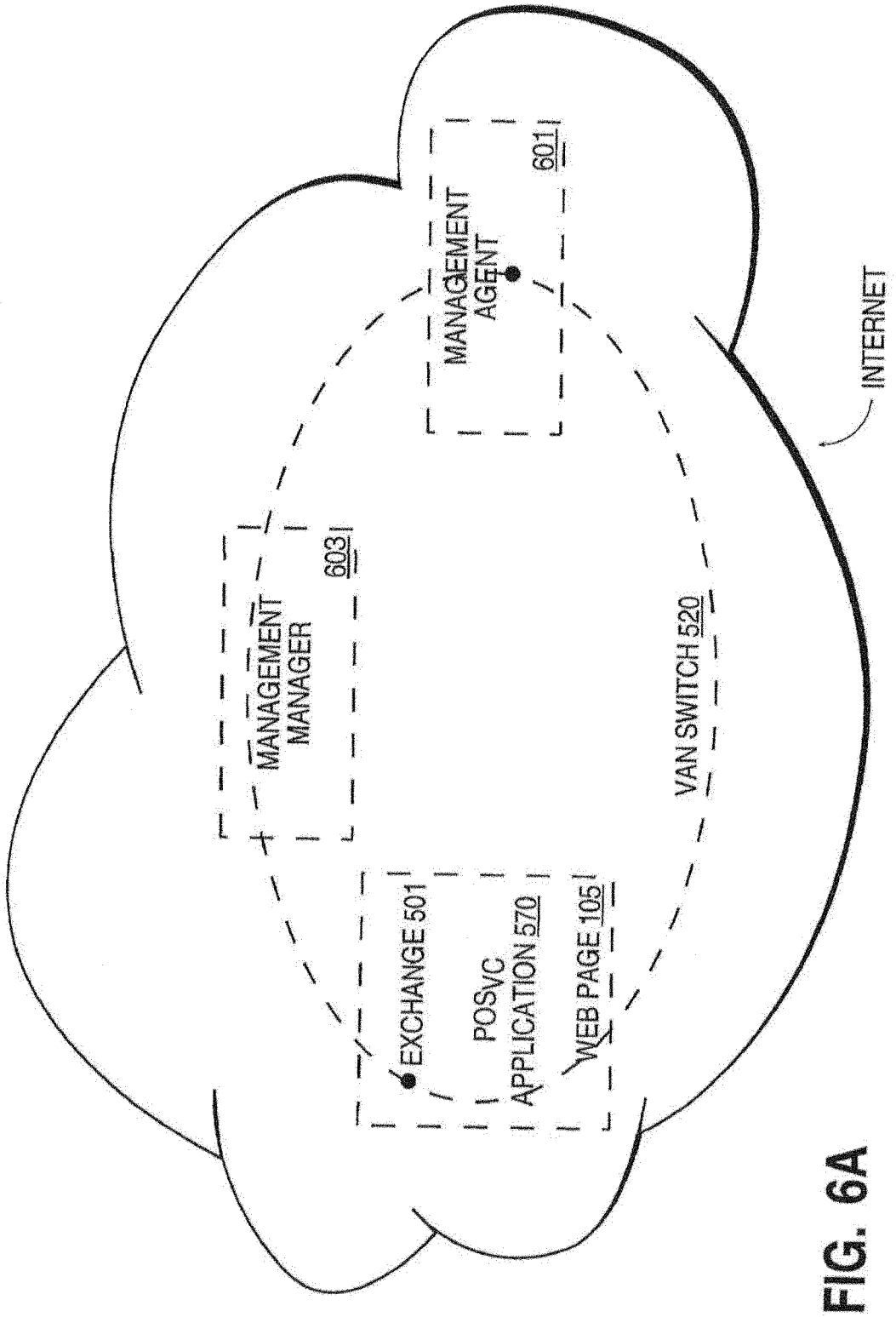

INTERNET
MANAGEMENT AGENT
601
MANAGEMENT MANAGER
603
VAN SWITCH 520
EXCHANGE 501
POSVC APPLICATION 570
WEB PAGE 105

FIG. 6A

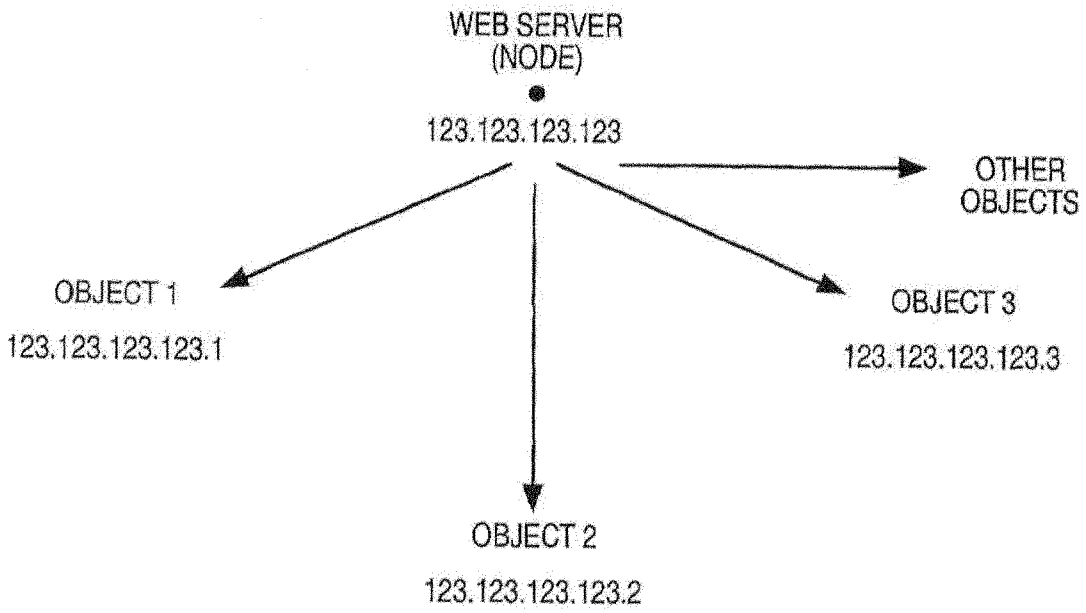

WEB SERVER
(NODE)
123.123.123.123
OTHER
OBJECTS
OBJECT 1
123.123.123.123.1
OBJECT 3
123.123.123.123.3
OBJECT 2
123.123.123.123.2

FIG. 6B

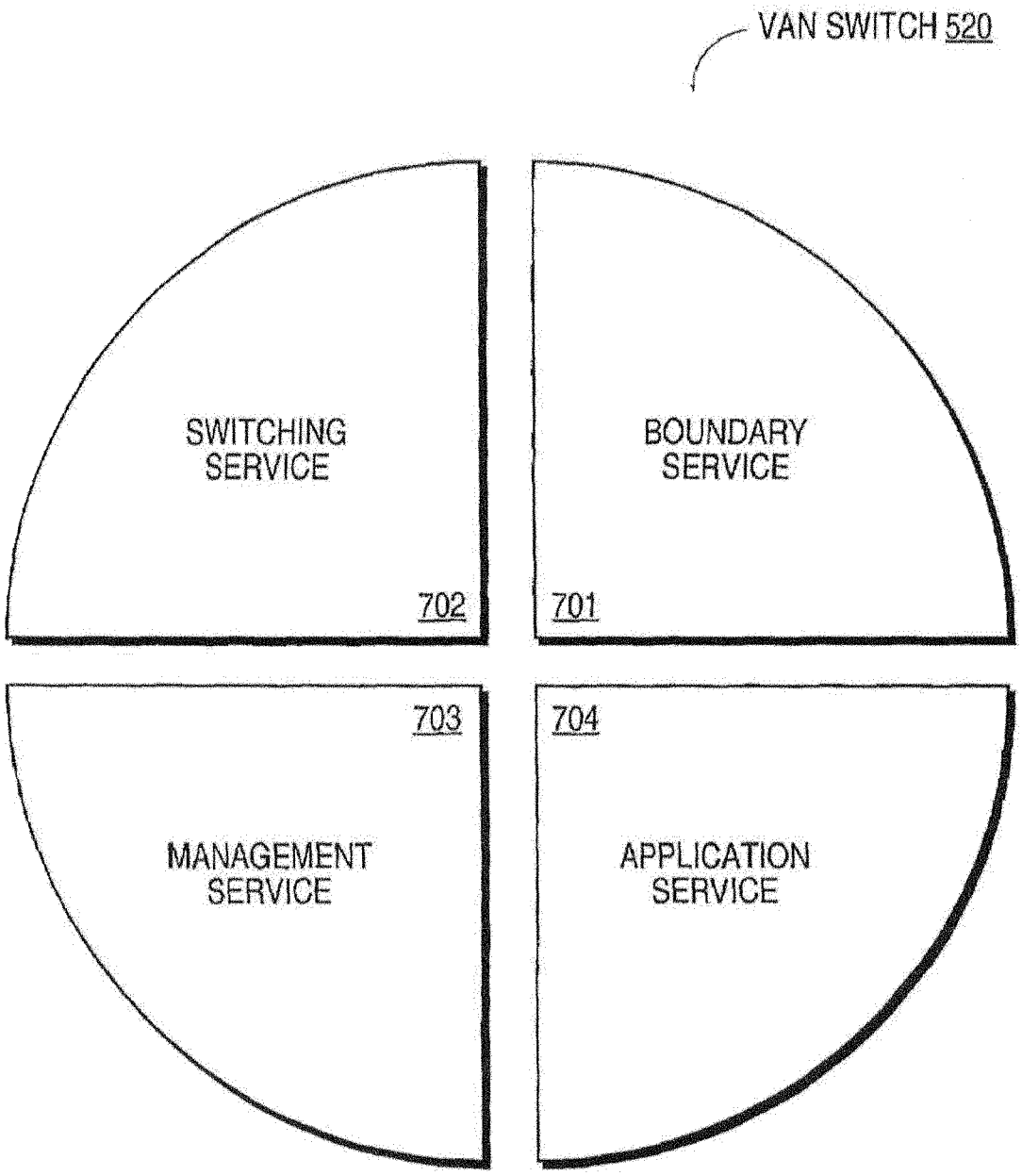

VAN SWITCH 520
SWITCHING SERVICE
702
BOUNDARY SERVICE
701
703
MANAGEMENT SERVICE
704
APPLICATION SERVICE

FIG. 7

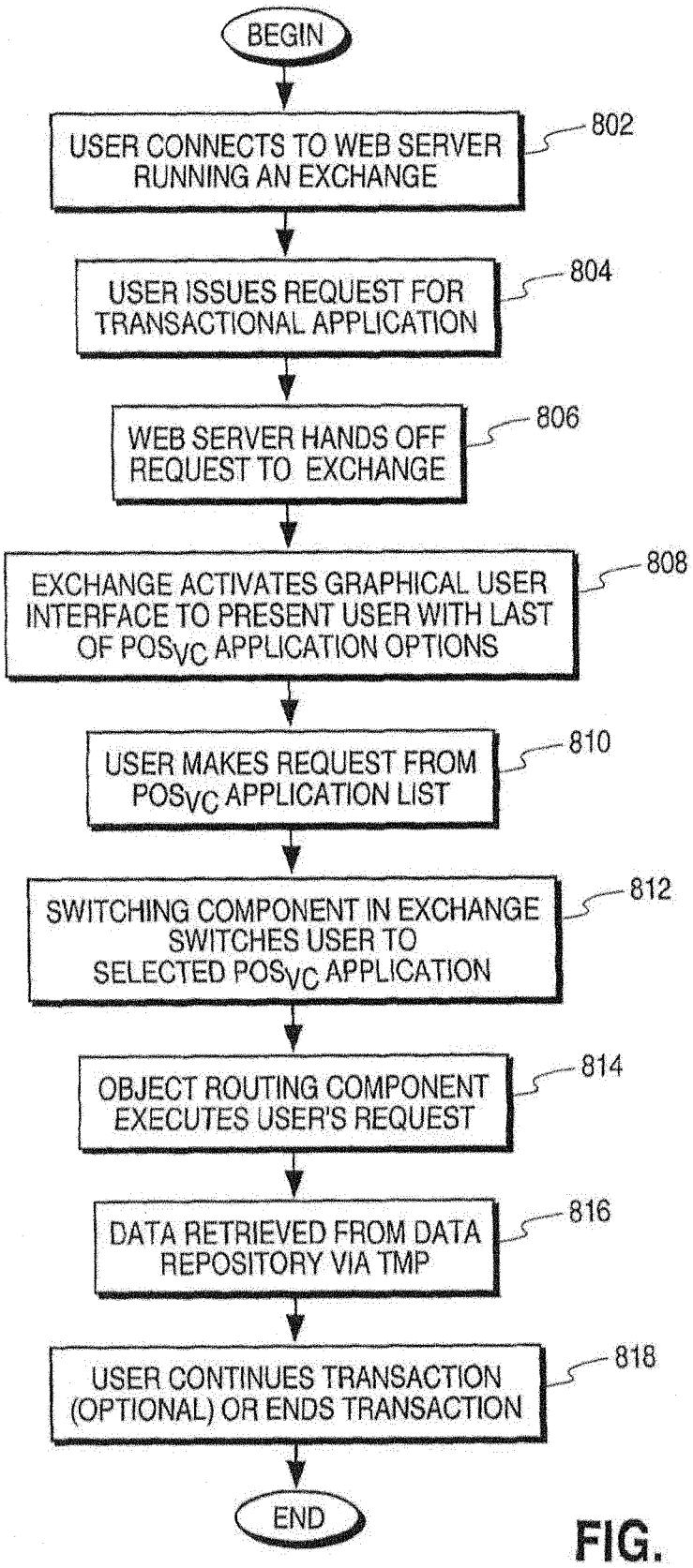

BEGIN
USER CONNECTS TO WEB SERVER RUNNING AN EXCHANGE
802
USER ISSUES REQUEST FOR TRANSACTIONAL APPLICATION
804
WEB SERVER HANDS OFF REQUEST TO EXCHANGE
806
EXCHANGE ACTIVATES GRAPHICAL USER INTERFACE TO PRESENT USER WITH LAST OF POSVC APPLICATION OPTIONS
808
USER MAKES REQUEST FROM POSVC APPLICATION LIST
810
SWITCHING COMPONENT IN EXCHANGE SWITCHES USER TO SELECTED POSVC APPLICATION
812
OBJECT ROUTING COMPONENT EXECUTES USER'S REQUEST
814
DATA RETRIEVED FROM DATA REPOSITORY VIA TMP
816
USER CONTINUES TRANSACTION (OPTIONAL) OR ENDS TRANSACTION
818
END

FIG. 8

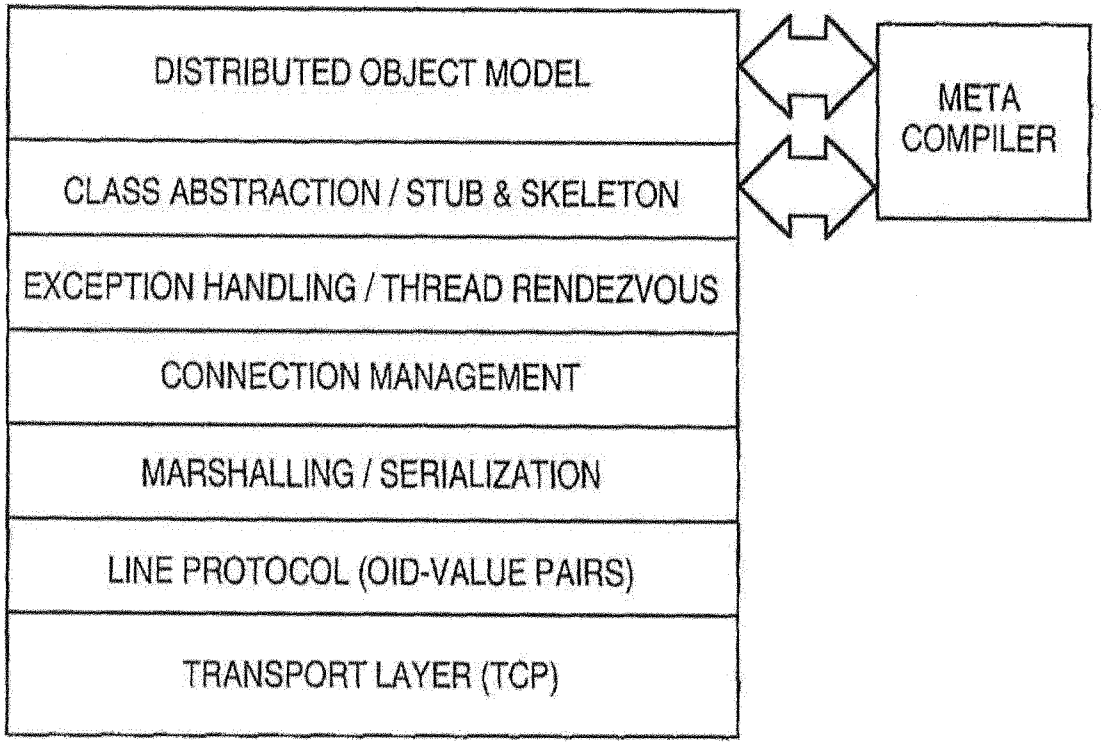

DISTRIBUTED OBJECT MODEL
CLASS ABSTRACTION / STUB & SKELETON
EXCEPTION HANDLING / THREAD RENDEZVOUS
CONNECTION MANAGEMENT
MARSHALLING / SERIALIZATION
LINE PROTOCOL (OID-VALUE PAIRS)
TRANSPORT LAYER (TCP)
META COMPILER

FIG. 9

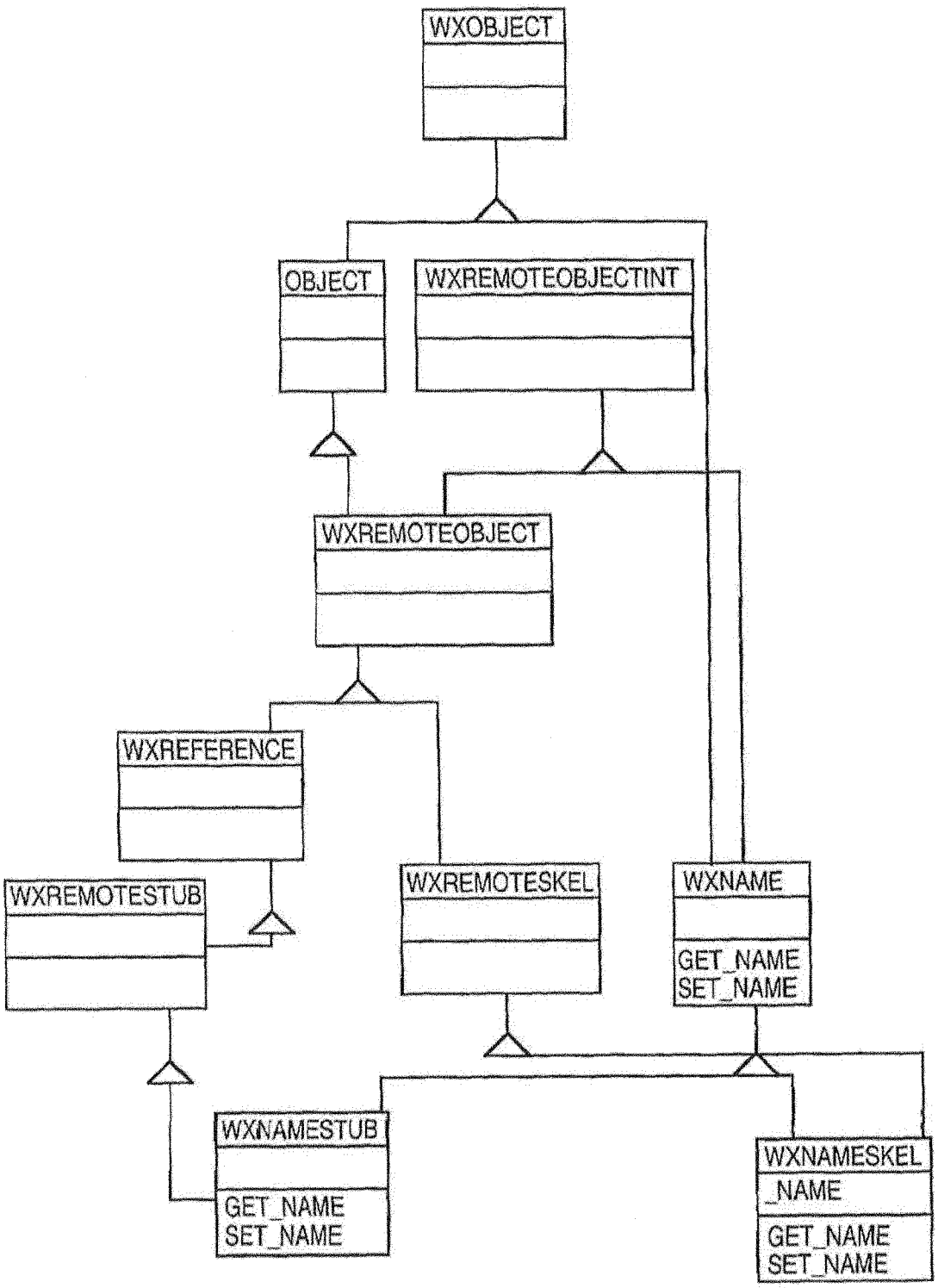

WXOBJECT
OBJECT
WXREMOTEOBJECTINT
WXREMOTEOBJECT
WXREFERENCE
WXREMOTESTUB
WXREMOTESKEL
WXNAME
GET_NAME
SET_NAME
WXNAMESTUB
GET_NAME
SET_NAME
WXNAMESKEL
_NAME
GET_NAME
SET_NAME

FIG. 10

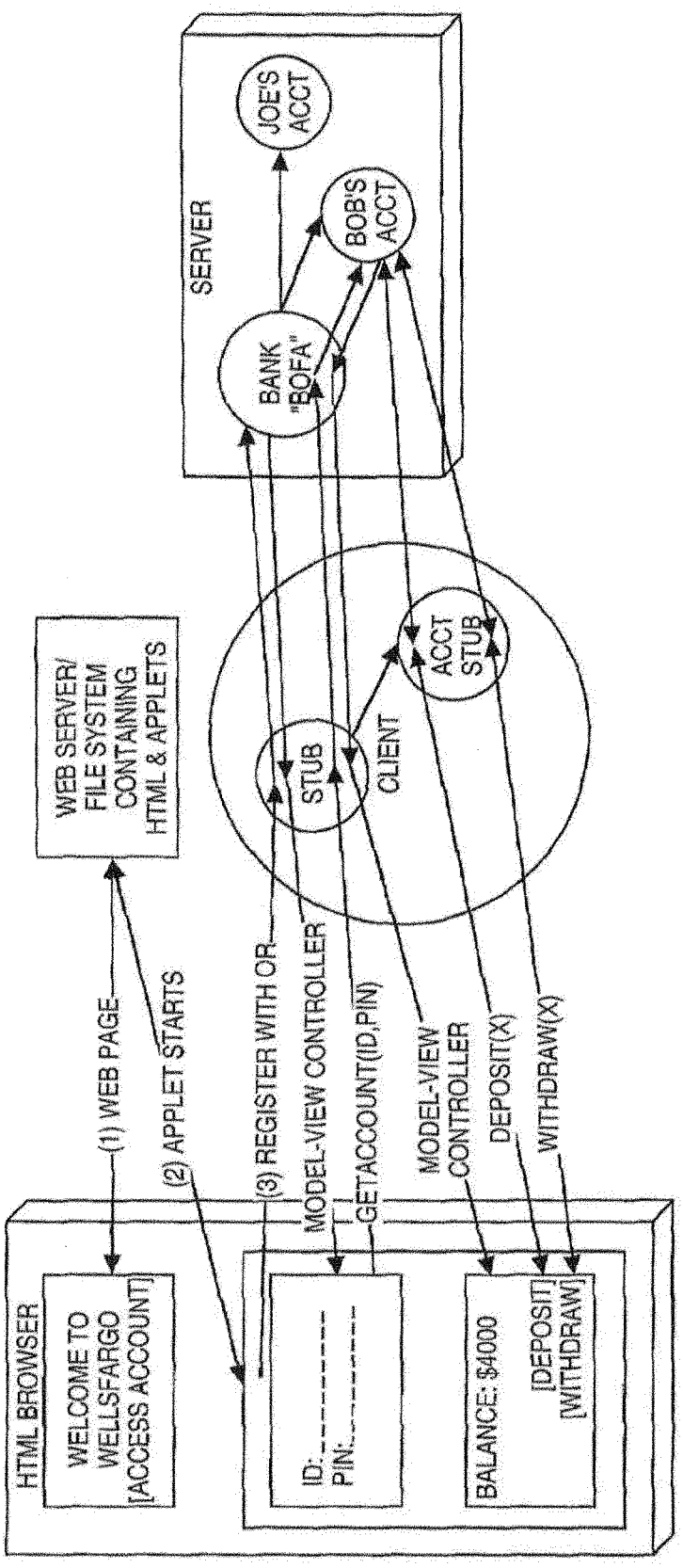

SERVER
JOE'S ACCT
BOB'S ACCT
BANK "BOFA"
WEB SERVER/ FILE SYSTEM CONTAINING HTML & APPLETS
CLIENT
STUB
ACCT STUB
(1) WEB PAGE
(2) APPLET STARTS
(3) REGISTER WITH OR
MODEL-VIEW CONTROLLER
GETACCOUNT(ID,PIN)
MODEL-VIEW CONTROLLER
DEPOSIT(X)
WITHDRAW(X)
HTML BROWSER
WELCOME TO WELLSFARGO [ACCESS ACCOUNT]
ID:_____
PIN:_____
BALANCE: $4000
[DEPOSIT]
[WITHDRAW]

FIG. 11

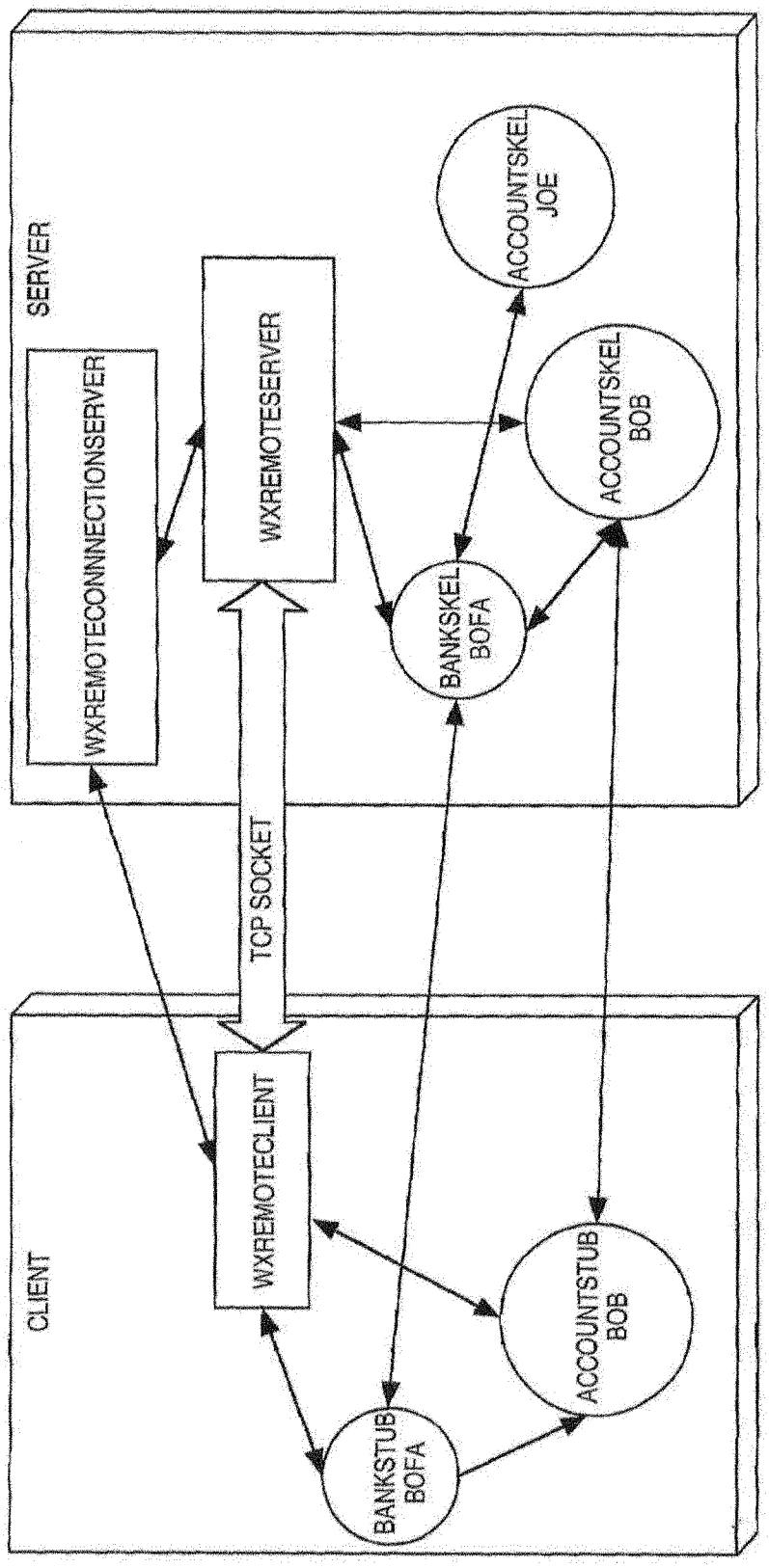

SERVER
WXREMOTECONNNECTIONSERVER
WXREMOTESERVER
ACCOUNTSKEL JOE
ACCOUNTSKEL BOB
BANKSKEL BOFA
TCP SOCKET
CLIENT
WXREMOTECLIENT
ACCOUNTSTUB BOB
BANKSTUB BOFA

FIG. 12

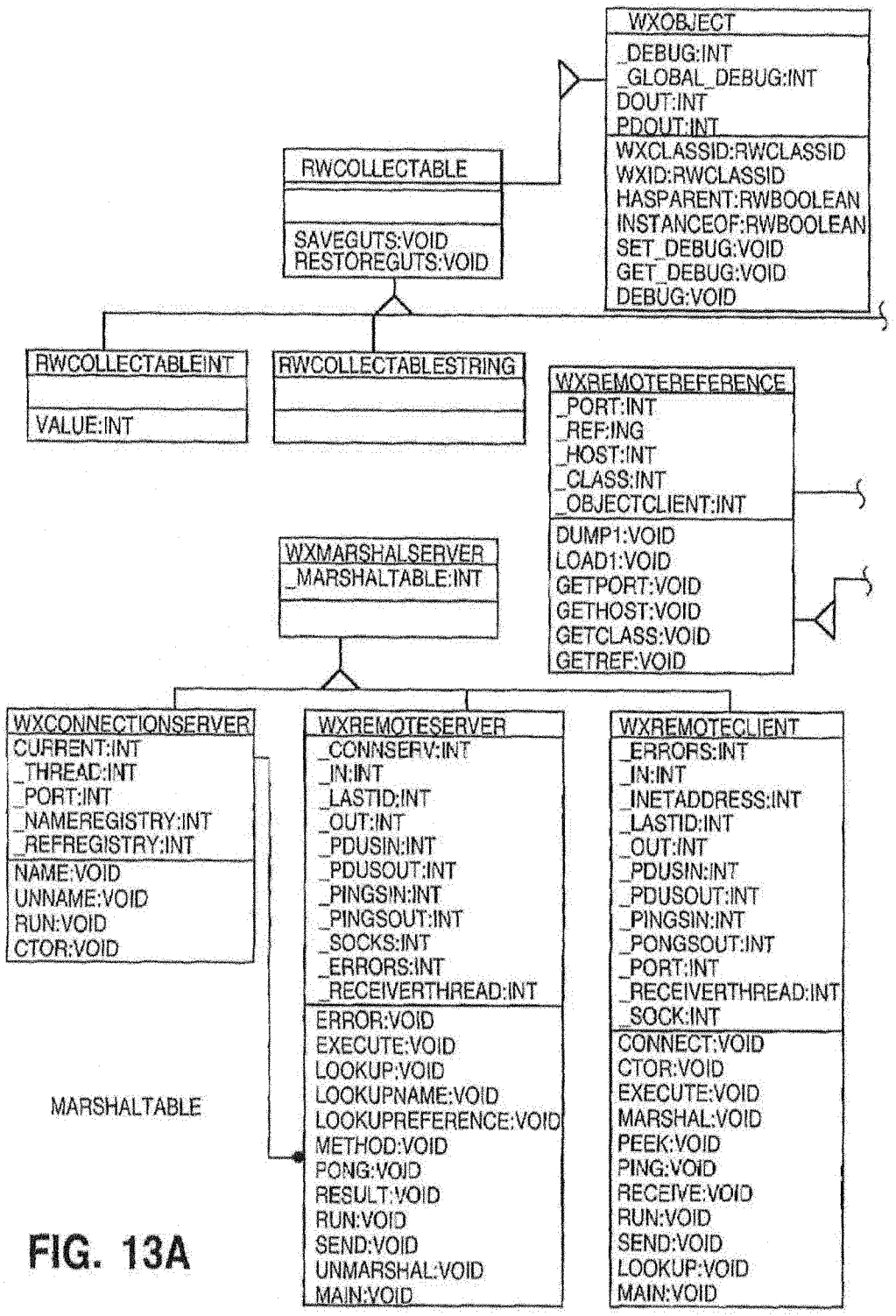

WXOBJECT
_DEBUG:INT
_GLOBAL_DEBUG:INT
DOUT:INT
PDOUT:INT
WXCLASSID:RWCLASSID
WXID:RWCLASSID
HASPARENT:RWBOOLEAN
INSTANCEOF:RWBOOLEAN
SET_DEBUG:VOID
GET_DEBUG:VOID
DEBUG:VOID
RWCOLLECTABLE
SAVEGUTS:VOID
RESTOREGUTS:VOID
RWCOLLECTABLEINT
VALUE:INT
RWCOLLECTABLESTRING
WXREMOTEREFERENCE
_PORT:INT
_REF:ING
_HOST:INT
_CLASS:INT
_OBJECTCLIENT:INT
DUMP1:VOID
LOAD1:VOID
GETPORT:VOID
GETHOST:VOID
GETCLASS:VOID
GETREF:VOID
WXMARSHALSERVER
_MARSHALTABLE:INT
WXCONNECTIONSERVER
CURRENT:INT
_THREAD:INT
_PORT:INT
_NAMEREGISTRY:INT
_REFREGISTRY:INT
NAME:VOID
UNNAME:VOID
RUN:VOID
CTOR:VOID
WXREMOTESERVER
_CONNSERV:INT
_IN:INT
_LASTID:INT
_OUT:INT
_PDUSIN:INT
_PDUSOUT:INT
_PINGSIN:INT
_PINGSOUT:INT
_SOCKS:INT
_ERRORS:INT
_RECEIVERTHREAD:INT
ERROR:VOID
EXECUTE:VOID
LOOKUP:VOID
LOOKUPNAME:VOID
LOOKUPREFERENCE:VOID
METHOD:VOID
PONG:VOID
RESULT:VOID
RUN:VOID
SEND:VOID
UNMARSHAL:VOID
MAIN:VOID
WXREMOTECLIENT
_ERRORS:INT
_IN:INT
_INETADDRESS:INT
_LASTID:INT
_OUT:INT
_PDUSIN:INT
_PDUSOUT:INT
_PINGSIN:INT
_PONGSOUT:INT
_PORT:INT
_RECEIVERTHREAD:INT
_SOCK:INT
CONNECT:VOID
CTOR:VOID
EXECUTE:VOID
MARSHAL:VOID
PEEK:VOID
PING:VOID
RECEIVE:VOID
RUN:VOID
SEND:VOID
LOOKUP:VOID
MAIN:VOID
MARSHALTABLE

FIG. 13A

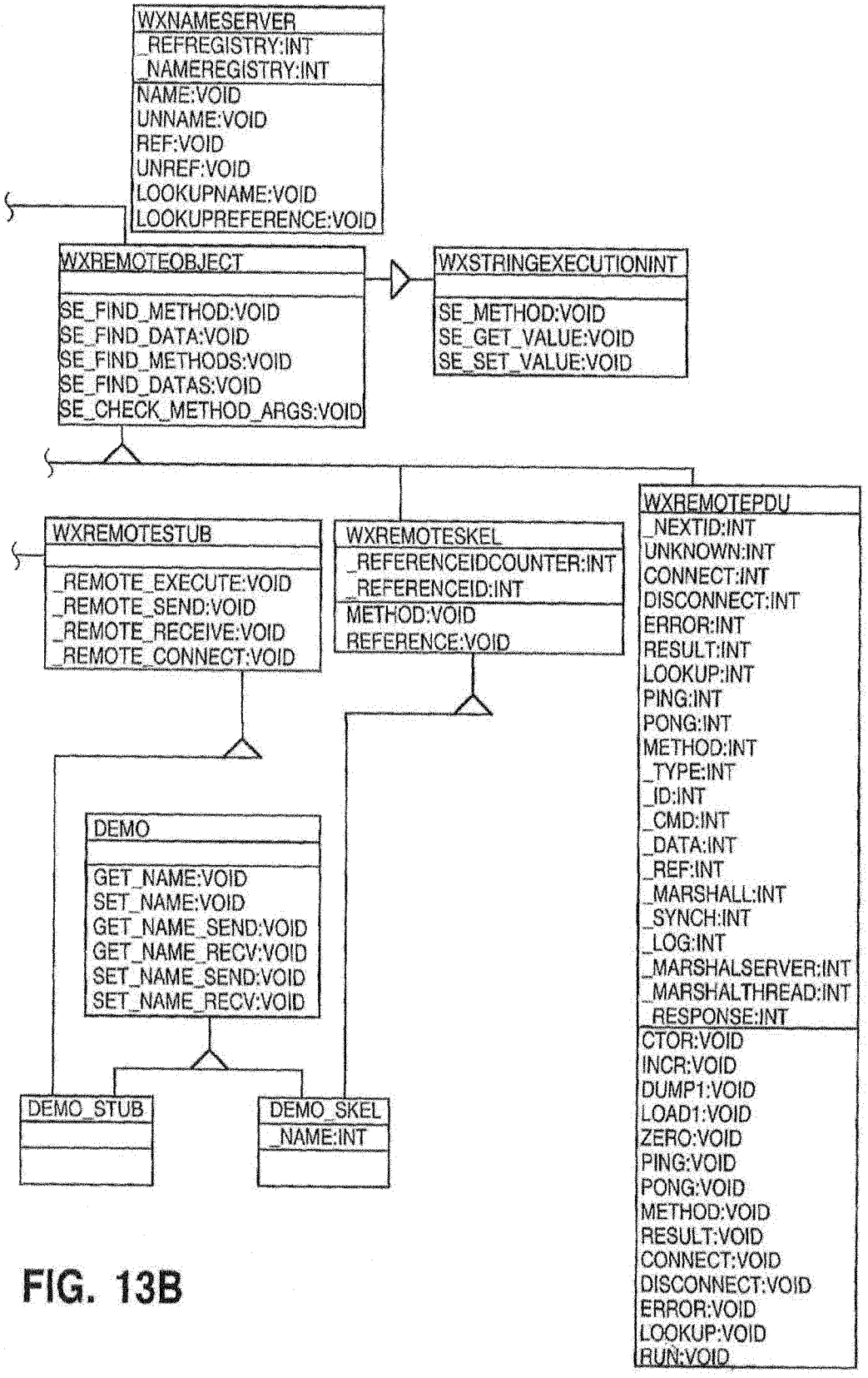

WXNAMESERVER
_REFREGISTRY:INT
_NAMEREGISTRY:INT
NAME:VOID
UNNAME:VOID
REF:VOID
UNREF:VOID
LOOKUPNAME:VOID
LOOKUPREFERENCE:VOID
WXREMOTEOBJECT
SE_FIND_METHOD:VOID
SE_FIND_DATA:VOID
SE_FIND_METHODS:VOID
SE_FIND_DATAS:VOID
SE_CHECK_METHOD_ARGS:VOID
WXSTRINGEXECUTIONINT
SE_METHOD:VOID
SE_GET_VALUE:VOID
SE_SET_VALUE:VOID
WXREMOTESTUB
_REMOTE_EXECUTE:VOID
_REMOTE_SEND:VOID
_REMOTE_RECEIVE:VOID
_REMOTE_CONNECT:VOID
WXREMOTESKEL
_REFERENCEIDCOUNTER:INT
_REFERENCEID:INT
METHOD:VOID
REFERENCE:VOID
WXREMOTEPDU
_NEXTID:INT
UNKNOWN:INT
CONNECT:INT
DISCONNECT:INT
ERROR:INT
RESULT:INT
LOOKUP:INT
PING:INT
PONG:INT
METHOD:INT
_TYPE:INT
_ID:INT
_CMD:INT
_DATA:INT
_REF:INT
_MARSHALL:INT
_SYNCH:INT
_LOG:INT
_MARSHALSERVER:INT
_MARSHALTHREAD:INT
_RESPONSE:INT
CTOR:VOID
INCR:VOID
DUMP1:VOID
LOAD1:VOID
ZERO:VOID
PING:VOID
PONG:VOID
METHOD:VOID
RESULT:VOID
CONNECT:VOID
DISCONNECT:VOID
ERROR:VOID
LOOKUP:VOID
RUN:VOID
DEMO
GET_NAME:VOID
SET_NAME:VOID
GET_NAME_SEND:VOID
GET_NAME_RECV:VOID
SET_NAME_SEND:VOID
SET_NAME_RECV:VOID
DEMO_STUB
DEMO_SKEL
_NAME:INT

FIG. 13B

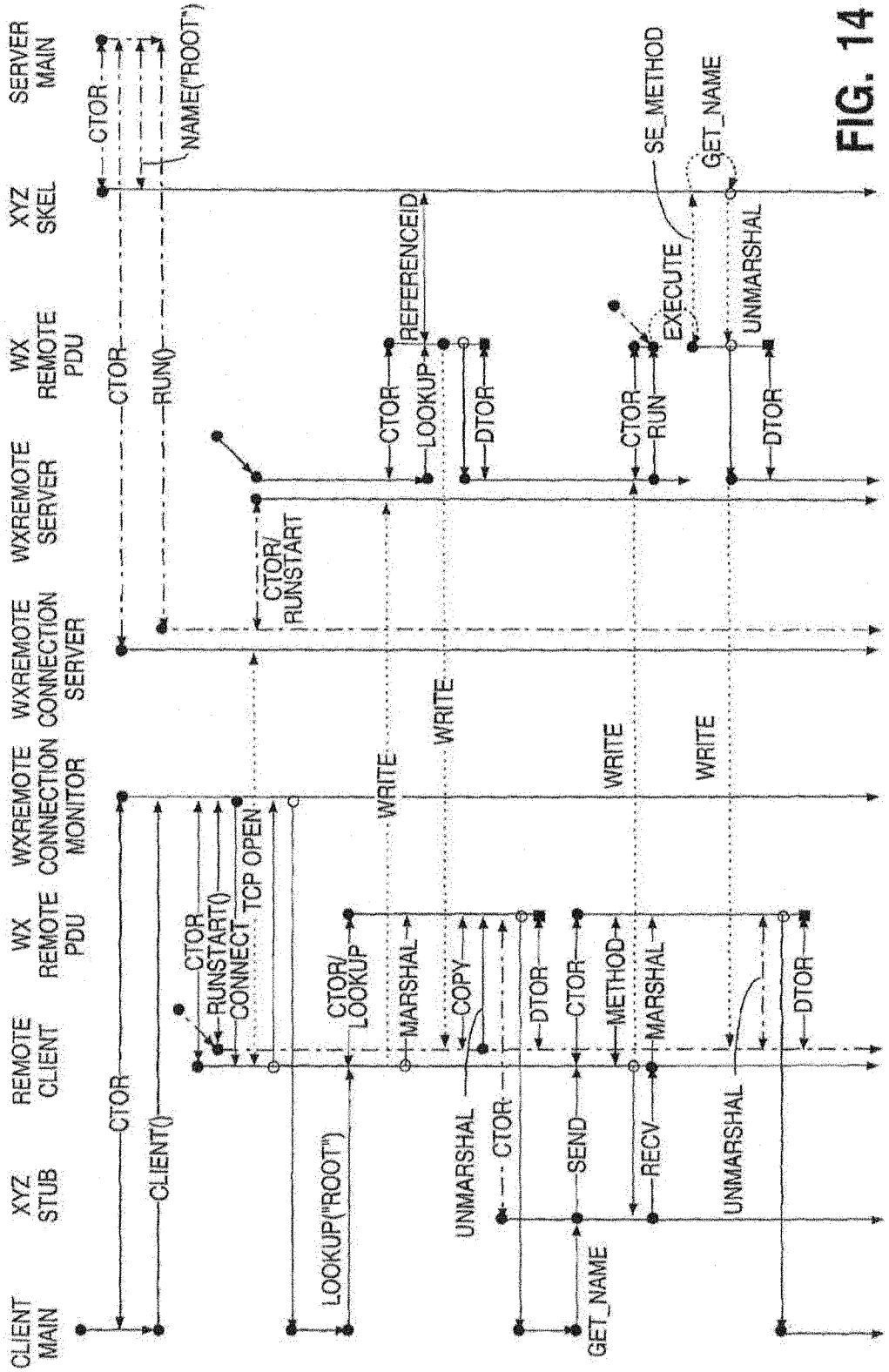

CLIENT MAIN
XYZ STUB
REMOTE CLIENT
WX REMOTE PDU
WXREMOTE CONNECTION MONITOR
WXREMOTE CONNECTION SERVER
WXREMOTE SERVER
WX REMOTE PDU
XYZ SKEL
SERVER MAIN
CTOR
CLIENT()
LOOKUP("ROOT")
CTOR/ LOOKUP
MARSHAL
UNMARSHAL
CTOR
SEND
GET_NAME
RECV
METHOD
MARSHAL
UNMARSHAL
DTOR
CTOR
RUNSTART()
CONNECT
TCP OPEN
WRITE
WRITE
WRITE
WRITE
CTOR/ RUNSTART
CTOR
RUN()
NAME("ROOT")
CTOR
LOOKUP
REFERENCEID
DTOR
CTOR
RUN
EXECUTE
SE_METHOD
GET_NAME
UNMARSHAL
DTOR

FIG. 14

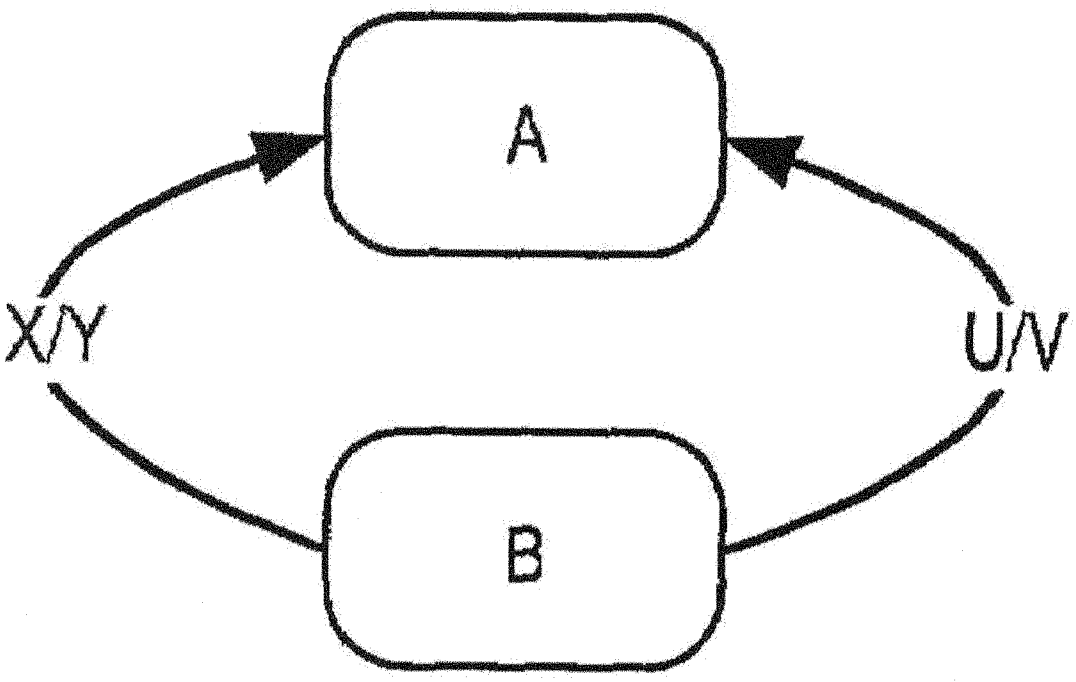


FIG. 15

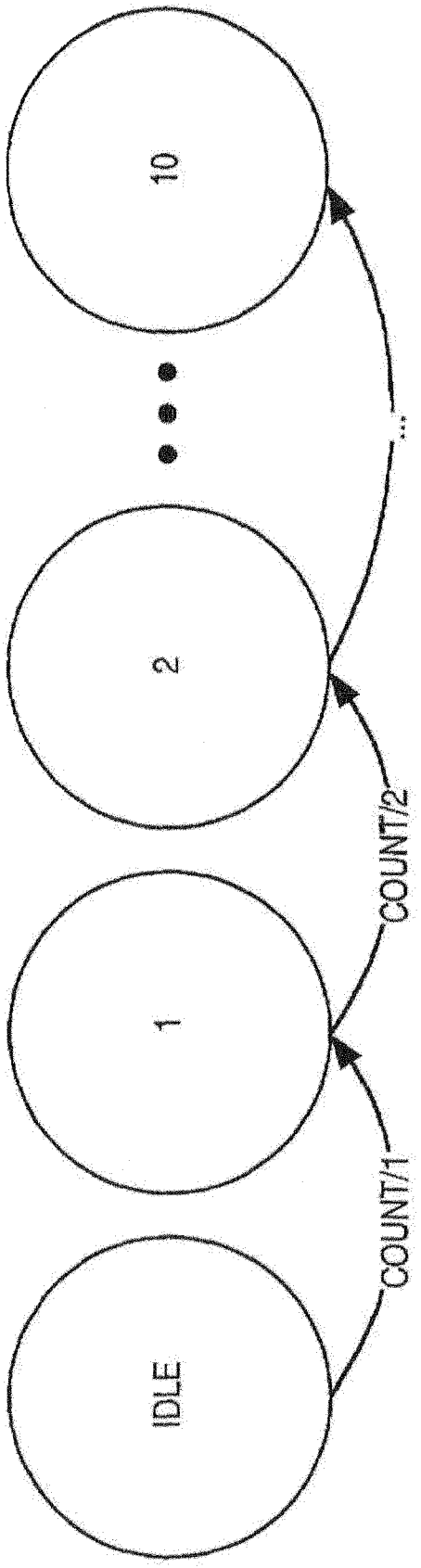


FIG. 16

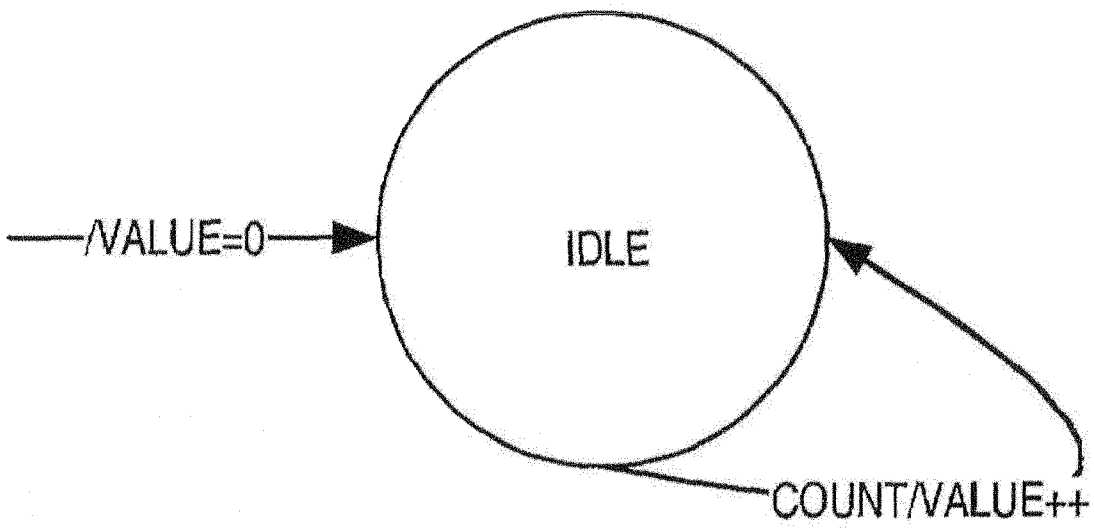


FIG. 17

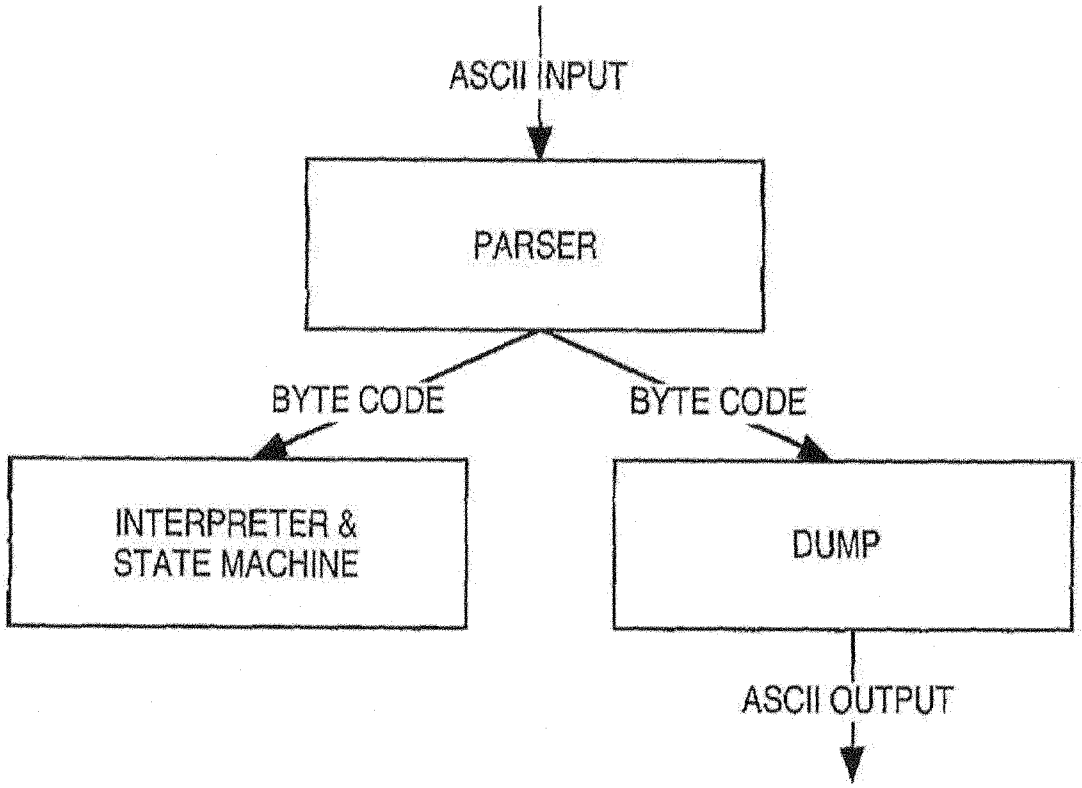

ASCII INPUT
PARSER
BYTE CODE
BYTE CODE
INTERPRETER & STATE MACHINE
DUMP
ASCII OUTPUT

FIG. 18

```
BEGINCLASS COREBUSINESSOBJECT
BEGINDATA
ENDDATA

BEGINMETHOD
INCLUDE HSKEL
PRIVATE:
# USED BY THE FINITE STATE MACHINE
VIRTUAL VOID FSM_INIT() { }
VOID FSM_ACTION_TIMEOUT(CONST CHAR* TIMEVAL);
VOID FSM_ACTION_THROW(CONST CHAR* MESSAGE);
VOID FSM_ACTION_RETURN(CONST CHAR* RESULT);
VOID FSM_ACTION_SEND(CONST CHAR* VALUE);
PUBLIC:
ENDINCLUDE

# TO CONIFIGURE THE FSM
METHOD      VOID   FSM_LOAD_DOLSIB {STRING FILENAME}
# TO TRIGGER AN EVENT IN THE FSM
METHOD      VOID   FSM_EVENT    {STRING NAME}  {STRING VALUE}
METHOD      STRING FSM_RESULT
# TO SET/GET VARIABLES FROM FSM
METHOD      VOID FSM_SET_STRING {STRING NAME} {STRING VALUE}
METHOD CONST STRING FSM_GET_STRING {STRING  NAME}
METHOD      VOID   FSM_SET_INTEGER {STRING NAME} {INT VALUE}
METHOD CONST INT   FSM_GET_INTEGER {STRING NAME}
ENDMETHOD

ENDCLASS
```

FIG. 19

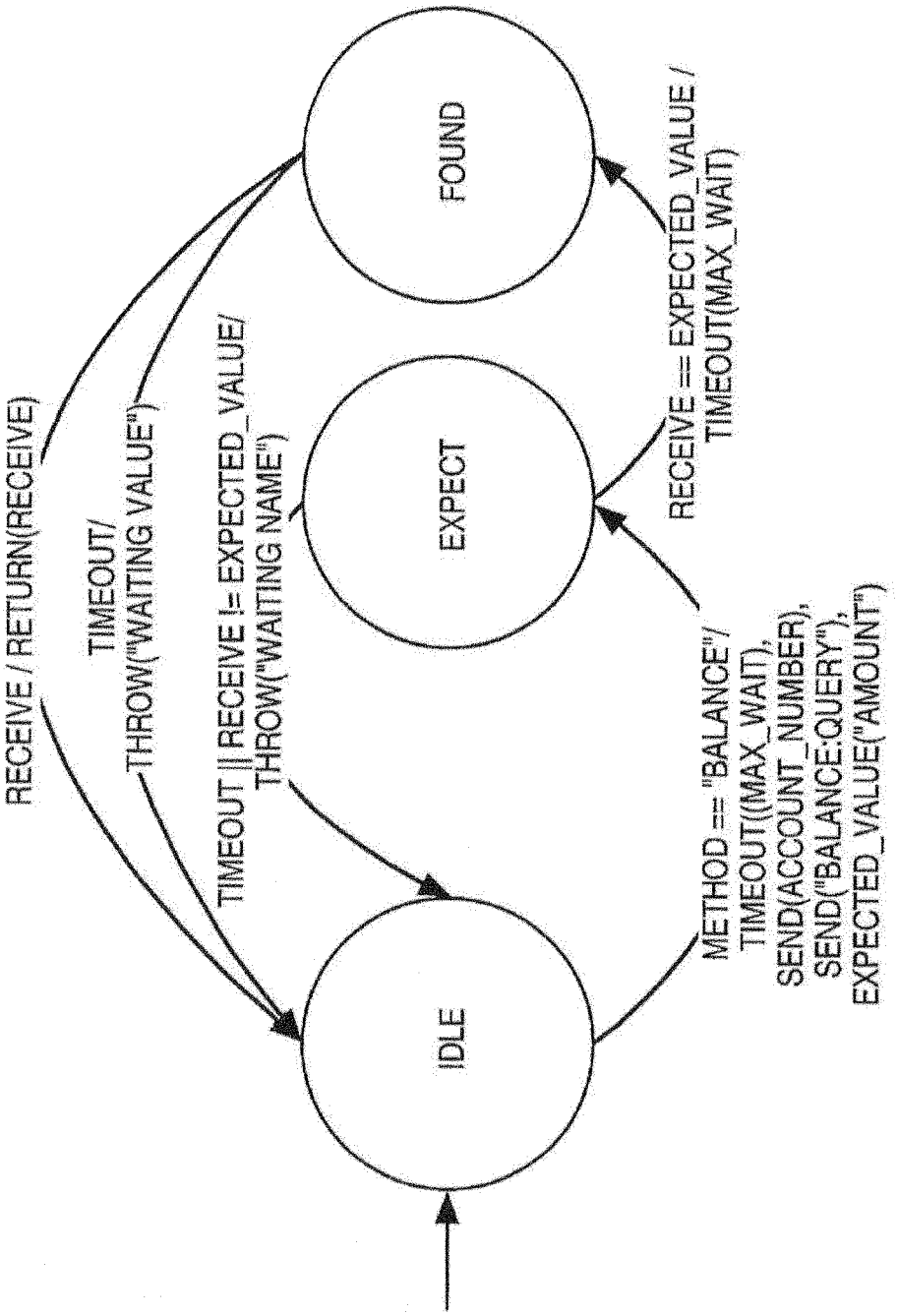

FOUND
EXPECT
IDLE
RECEIVE / RETURN(RECEIVE)
TIMEOUT/
THROW("WAITING VALUE")
TIMEOUT || RECEIVE != EXPECTED_VALUE/
THROW("WAITING NAME")
RECEIVE == EXPECTED_VALUE /
TIMEOUT(MAX_WAIT)
METHOD == "BALANCE"/
TIMEOUT((MAX_WAIT),
SEND(ACCOUNT_NUMBER),
SEND("BALANCE:QUERY"),
EXPECTED_VALUE("AMOUNT")

FIG. 20

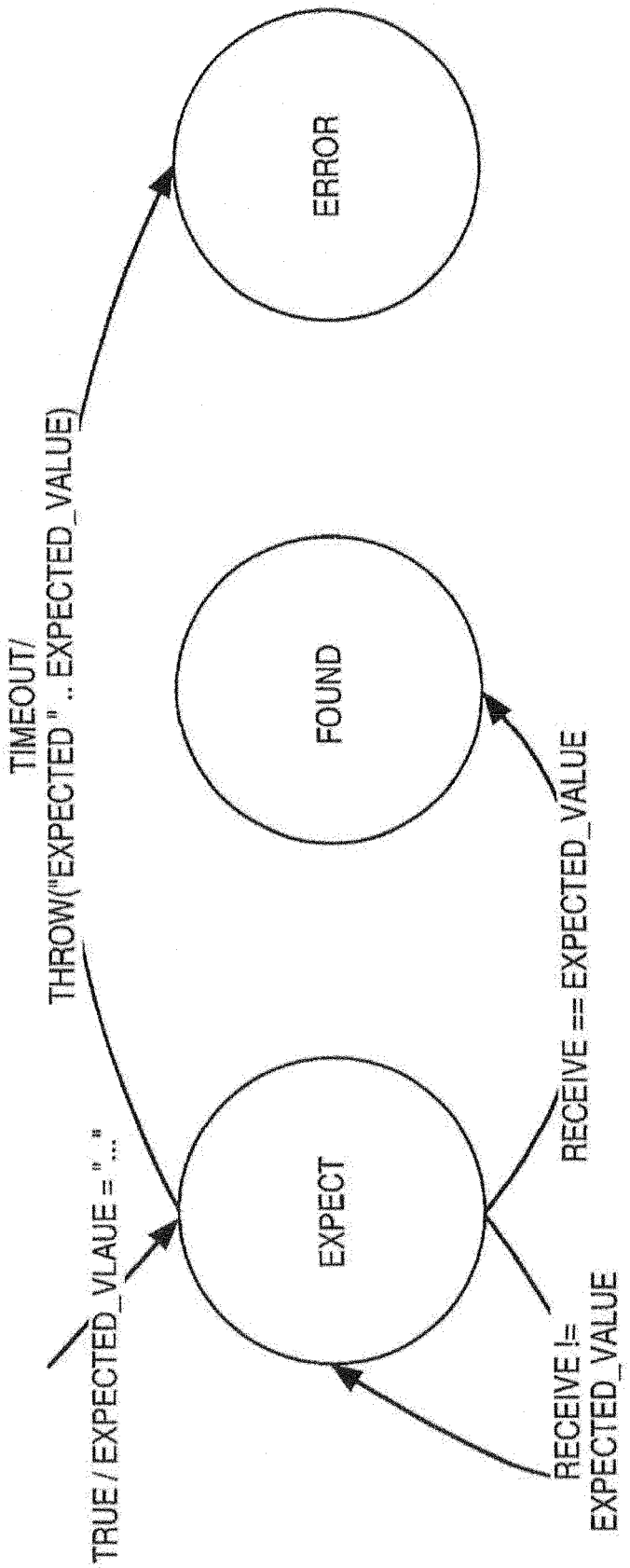

ERROR
FOUND
EXPECT
TIMEOUT/
THROW("EXPECTED " .. EXPECTED_VALUE)
TRUE / EXPECTED_VLAUE = " ... "
RECEIVE == EXPECTED_VALUE
RECEIVE != EXPECTED_VALUE

FIG. 21

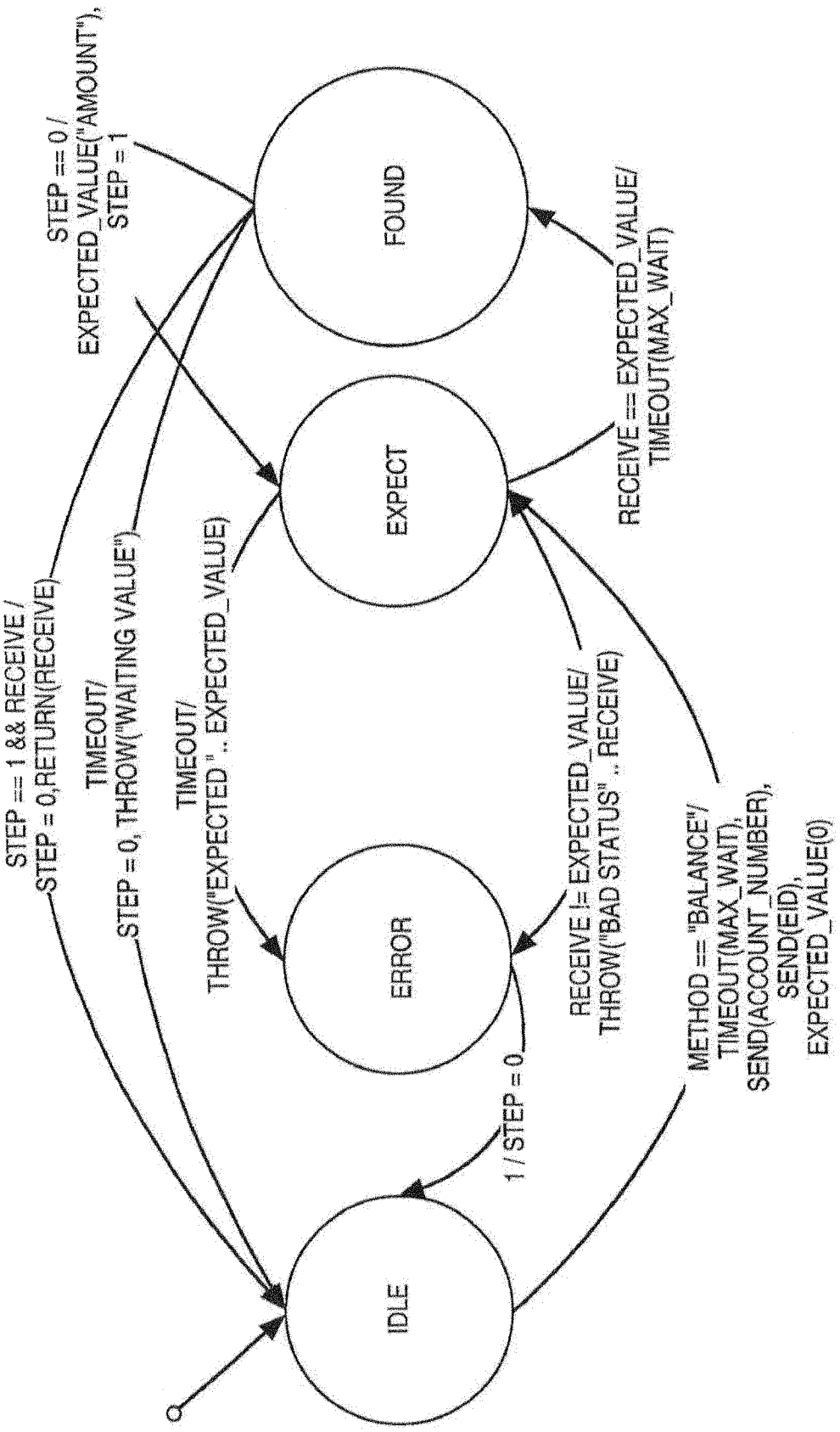

IDLE
ERROR
EXPECT
FOUND
STEP == 1 && RECEIVE /
STEP = 0,RETURN(RECEIVE)
TIMEOUT/
STEP = 0, THROW("WAITING VALUE")
TIMEOUT/
THROW("EXPECTED " .. EXPECTED_VALUE)
STEP == 0 /
EXPECTED_VALUE("AMOUNT"),
STEP = 1
RECEIVE != EXPECTED_VALUE/
THROW("BAD STATUS" .. RECEIVE)
RECEIVE == EXPECTED_VALUE/
TIMEOUT(MAX_WAIT)
1 / STEP = 0
METHOD == "BALANCE"/
TIMEOUT(MAX_WAIT),
SEND(ACCOUNT_NUMBER),
SEND(EID),
EXPECTED_VALUE(0)

FIG. 22

VALUE-ADDED NETWORK SWITCHING AND OBJECT ROUTING

RELATED APPLICATIONS

This application is a divisional patent application of application Ser. No. 09/296,207, filed Apr. 21, 1999, now U.S. Pat. No. 6,212,556 which is a continuation-in-part patent application claiming priority to U.S. Pat. No. 5,987,500, entitled Value-Added Network System For Enabling Real-Time, Bi-Directional Transactions On A Network, formerly application Ser. No. 08/879,958, and filed on Jun. 20, 1997. U.S. Pat. No. 5,987,500 is itself a divisional application claiming priority to U.S. Pat. No. 5,778,178, entitled Method And Apparatus For Enabling Real-Time Bi-Directional Transactions On A Network, formerly application Ser. No. 08/700,726, filed on Aug. 5, 1996, which also claims priority and benefit under 35 U.S.C. 119(e) to U.S. Provisional Application No. 60/006,634 filed on Nov. 13, 1995.

FIELD OF THE INVENTION

The present invention relates to the area of network communications, including heterogeneous networks such as the Internet, World Wide Web (WWW), telephone network, wireless networks, cable television networks, and private enterprise networks. Specifically, the present invention relates to a method and apparatus for configurable value-added network switching and object routing.

BACKGROUND OF THE INVENTION

With the Internet and the World Wide Web ("the Web") evolving rapidly as a viable consumer medium for electronic commerce, new on-line services are emerging to fill the needs of on-line users. An Internet user today can browse on the Web via the use of a Web browser. Web browsers are software interfaces that run on Web clients to allow access to Web servers via a simple user interface. A Web user's capabilities today from a Web browser are, however, extremely limited. The user can perform one-way, browse-only interactions. Additionally, the user has limited "deferred" transactional capabilities, namely electronic mail (e-mail) capabilities. E-mail capabilities are referred to as "deferred transactions" because the consumer's request is not processed until the e-mail is received, read, and the person or system reading the e-mail executes the transaction. This transaction is thus not performed in real-time.

FIG. **1A** illustrates typical user interactions on the Web today. User **100** sends out a request from Web browser **102** in the form of a universal resource locator (URL) **101** in the following manner: http://www.car.com. URL **101** is processed by Web browser **102** that determines the URL corresponds to car dealer Web page **105**, on car dealer Web server **104**. Web browser **102** then establishes browse link **103** to car dealer Web page **105**. User **100** can browse Web page **105** and select "hot links" to jump to other locations in Web page **105**, or to move to other Web pages on the Web. This interaction is typically a browse-only interaction. Under limited circumstances, the user may be able to fill out a form on car dealer Web page **105**, and e-mail the form to car dealer Web server **104**. This interaction is still strictly a one-way browse mode communications link, with the e-mail providing limited, deferred transactional capabilities.

Under limited circumstances, a user may have access to two-way services on the Web via Common Gateway Interface (CGI) applications. CGI is a standard interface for

running external programs on a Web server. It allows Web servers to create documents dynamically when the server receives a request from the Web browser. When the Web server receives a request for a document, the Web server dynamically executes the appropriate CGI script and transmits the output of the execution back to the requesting Web browser. This interaction can thus be termed a "two-way" transaction. It is a severely limited transaction, however, because each CGI application is customized for a particular type of application or service.

For example, as illustrated in FIG. **1B**, user **100** may access bank **150**'s Web server and attempt to perform transactions on checking account **152** and to make a payment on loan account **154**. In order for user **100** to access checking account **152** and loan account **154** on the Web, CGI application scripts must be created for each account, as illustrated in FIG. **1B**. The bank thus has to create individual scripts for each of its services to offer users access to these services. User **100** can then interact in a limited fashion with these individual applications. Creating and managing individual CGI scripts for each service is not a viable solution for merchants with a large number of services.

As the Web expands and electronic commerce becomes more desirable, the need increases for robust, real-time, bi-directional transactional capabilities on the Web. A true real-time, bi-directional transaction would allow a user to connect to a variety of services on the Web, and perform real-time transactions on those services. For example, although user **100** can browse car dealer Web page **105** today, the user cannot purchase the car, negotiate a car loan or perform other types of real-time, two-way transactions that he can perform with a live salesperson at the car dealership. Ideally, user **100** in FIG. **1A** would be able to access car dealer Web page **105**, select specific transactions that he desires to perform, such as purchase a car, and perform the purchase in real-time, with two-way interaction capabilities. CGI applications provide user **100** with a limited ability for two-way interaction with car dealer Web page **105**, but due to the lack of interaction and management between the car dealer and the bank, he will not be able to obtain a loan and complete the purchase of the car via a CGI application. The ability to complete robust real-time, two-way transactions is thus not truly available on the Web today.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method and apparatus for providing real-time, two-way transactional capabilities on the network, including heterogeneous networks such as the Internet, World Wide Web (WWW), telephone network, wireless networks, cable television networks, and private enterprise networks. Specifically, one embodiment of the present invention discloses a configurable value-added network switch for enabling real-time transactions on a network. The configurable value added network switch comprises means for switching to a transactional application in response to a user specification from a network application, means for transmitting a transaction request from the transactional application, and means for processing the transaction request.

According to another aspect of the present invention, a method and apparatus for enabling object routing on a network is disclosed. The method for enabling object routing

comprises the steps of creating a virtual information store containing information entries and attributes, associating each of the information entries and the attributes with an object identity, and assigning a unique network address to each of the object identities.

Other objects, features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description of the present invention as set forth below.

FIG. **1**A is an illustration of a current user's browse capabilities on a network via a network browser.

FIG. **1**B is an illustration of a current user's capabilities to perform limited transactions on the Network via CGI applications.

FIG. **2** illustrates a typical computer system on which the present invention may be utilized.

FIG. **3** illustrates the Open Systems Interconnection (OSI) Model.

FIG. **4**A illustrates conceptually the user value chain as it exists today.

FIG. **4**B illustrates one embodiment of the present invention.

FIG. **5**A illustrates a user accessing a Network server including one embodiment of the present invention.

FIG. **5**B illustrates the exchange component according to one embodiment of the present invention.

FIG. **5**C illustrates an example of a point-of-service (POSvc) application list.

FIG. **5**D illustrates a user selecting a bank POSvc application from the POSvc application list.

FIG. **5**E illustrates a three-way transaction according to one embodiment of the present invention.

FIG. **6**A illustrates a value-added network (VAN) switch.

FIG. **6**B illustrates the hierarchical addressing tree structure of the networked objects in DOLSIBs.

FIG. **7** illustrates conceptually the layered architecture of a VAN switch.

FIG. **8** is a flow diagram illustrating one embodiment of the present invention.

FIG. **9** illustrates the software layers of the Object Router.

FIG. **10** illustrates the data model integration.

FIG. **11** illustrates a bank scenario example.

FIG. **12** illustrates a bank client-server example.

FIGS. **13**A & **13**B illustrate.

FIG. **14** illustrates an Object Router Timing diagram of the preferred embodiment.

FIG. **15** illustrates a finite state machine.

FIG. **16** illustrates a counter implemented with a finite state machine.

FIG. **17** illustrates a counter implemented with the present invention.

FIG. **18** illustrates a sample state machine with files and programs.

FIG. **19** illustrates a CoreBusinessObject Object Router description.

FIG. **20** illustrates an example of a Bank B1 DOLSIB FSM Diagram for Balance.

FIG. **21** illustrates a diagram for expect, found, and error states.

FIG. **22** illustrates an example of a Bank B2 DOLSIB FSM Diagram for Balance.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to a method and apparatus for configurable value-added network switching and object routing and management. The network browser or "Web browser" as used in the context of the present specification includes conventional Web browsers such as NCSA Mosaic™ from NCSA and Netscape Mosaic™ from Netscape™. The present invention is independent of the Network browser being utilized and the user can use any Network browser, without modifications to the Network browser. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one of ordinary skill in the art, however, that these specific details need not be used to practice the present invention. In other instances, well-known structures, interfaces and processes have not been shown in detail in order not to unnecessarily obscure the present invention.

FIG. **2** illustrates a typical computer system **200** in which the present invention operates. The preferred embodiment of the present invention is implemented on an IBM™ Personal Computer manufactured by IBM Corporation of Armonk, N.Y. Alternate embodiments may be implemented on a Macintosh™ computer manufactured by Apple™ Computer, Incorporated of Cupertino, Calif. It will be apparent to those of ordinary skill in the art that other alternative computer system architectures may also be employed.

In general, such computer systems as illustrated by FIG. **2** comprise a bus **201** for communicating information, a processor **202** coupled with the bus **201** for processing information, main memory **203** coupled with the bus **201** for storing information and instructions for the processor **202**, a read-only memory **204** coupled with the bus **201** for storing static information and instructions for the processor **202**, a display device **205** coupled with the bus **201** for displaying information for a computer user, an input device **206** coupled with the bus **201** for communicating information and command selections to the processor **202**, and a mass storage device **207**, such as a magnetic disk and associated disk drive, coupled with the bus **201** for storing information and instructions. A data storage medium **208** containing digital information is configured to operate with mass storage device **207** to allow processor **202** access to the digital information on data storage medium **208** via bus **201**.

Processor **202** may be any of a wide variety of general purpose processors or microprocessors such as the Pentium™ microprocessor manufactured by Intel™ Corporation or the Motorola™ 68040 or Power PC™ brand microprocessor manufactured by manufactured by Motorola™ Corporation. It will be apparent to those of ordinary skill in the art, however, that other varieties of processors may also be used in a particular computer system. Display device **205** may be a liquid crystal device, cathode ray tube (CRT), or other suitable display device. Mass storage device **207** may be a conventional hard disk drive, floppy disk drive, CD-ROM drive, or other magnetic or optical data storage device for reading and writing information stored on a hard disk, a floppy disk, a CD-ROM a magnetic tape, or other magnetic or optical data storage medium. Data storage medium **208** may be a hard disk, a floppy disk, a CD-ROM, a magnetic tape, or other magnetic or optical data storage medium.

In general, processor **202** retrieves processing instructions and data from a data storage medium **208** using mass storage device **207** and downloads this information into random access memory **203** for execution. Processor **202**, then

executes an instruction stream from random access memory **203** or read-only memory **204**. Command selections and information input at input device **206** are used to direct the flow of instructions executed by processor **202**. Equivalent input device **206** may also be a pointing device such as a conventional mouse or trackball device. The results of this processing execution are then displayed on display device **205**.

The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system **200** in a conventional manner. Using well known techniques, the application software of the preferred embodiment is stored on data storage medium **208** and subsequently loaded into and executed within computer system **200**. Once initiated, the software of the preferred embodiment operates in the manner described below.

FIG. **3** illustrates the Open Systems Interconnection (OSI) reference model. OSI Model **300** is an international standard that provides a common basis for the coordination of standards development, for the purpose of systems interconnection. The present invention is implemented to function as a routing switch within the "application layer" of the OSI model. The model defines seven layers, with each layer communicating with its peer layer in another node through the use of a protocol. Physical layer **301** is the lowest layer, with responsibility to transmit unstructured bits across a link. Data link layer **302** is the next layer above physical layer **301**. Data link layer **302** transmits chunks across the link and deals with problems like checksumming to detect data corruption, orderly coordination of the use of shared media and addressing when multiple systems are reachable. Network bridges operate within data link layer **302**.

Network layer **303** enables any pair of systems in the network to communicate with each other. Network layer **303** contains hardware units such as routers, that handle routing, packet fragmentation and reassembly of packets. Transport layer **304** establishes a reliable communication stream between a pair of systems, dealing with errors such as lost packets, duplicate packets, packet reordering and fragmentation. Session layer **305** offers services above the simple communication stream provided by transport layer **304**. These services include dialog control and chaining. Presentation layer **306** provides a means by which OSI compliant applications can agree on representations for data. Finally, application layer **307** includes services such as file transfer, access and management services (FTAM), electronic mail and virtual terminal (VT) services. Application layer **307** provides a means for application programs to access the OSI environment. As described above, the present invention is implemented to function as a routing switch in application layer **307**. Application layer routing creates an open channel for the management, and the selective flow of data from remote databases on a network.

A. Overview

FIG. **4**A illustrates conceptually the user value chain as it exists today. The user value chain in FIG. **4**A depicts the types of transactions that are performed today, and the channels through which the transactions are performed. A "transaction" for the purposes of the present invention includes any type of commercial or other type of interaction that a user may want to perform. Examples of transactions include a deposit into a bank account, a request for a loan from a bank, a purchase of a car from a car dealership or a purchase of a car with financing from a bank. A large variety of other transactions are also possible.

A typical user transaction today may involve user **100** walking into a bank or driving up to a teller machine, and interacting with a live bank teller, or automated teller machine (ATM) software applications. Alternatively, user **100** can perform the same transaction by using a personal computer (PC), activating application software on his PC to access his bank account, and dialing into the bank via a modem line. If user **100** is a Network user, however, there is no current mechanism for performing a robust, real-time transaction with the bank, as illustrated in FIG. **4**A. CGI scripts provide only limited two-way capabilities, as described above. Thus, due to this lack of a robust mechanism by which real-time Network transactions can be performed, the bank is unable to be a true "Network merchant," namely a merchant capable of providing complete transactional services on the Network.

According to one embodiment of the present invention, as illustrated in FIG. **4**B, each merchant that desires to be a network merchant can provide real-time transactional capabilities to users who desire to access the merchants' services via the network. This embodiment includes a service network running on top of a facilities network, namely the Internet, the Web or e-mail networks. For the purposes of this application, users are described as utilizing PC's to access the network via network server "switching" sites. (Switching is described in more detail below). Users may also utilize other personal devices such as network computers or cellular devices to access the merchants' services via appropriate switching sites. These switching sites include non-Web network computer sites and cellular provider sites. Five components interact to provide this service network functionality, namely an exchange, an operator agent, a management agent, a management manager and a graphical user interface. All five components are described in more detail below.

As illustrated in FIG. **5**A, user **100** accesses network server **104**. Having accessed network server **104**, user **100** can decide that he desires to perform real-time transactions. When network server **104** receives user **100**'s indication that he desires to perform real-time transactions, the request is handed over to an exchange component. Thus, from network page **105**, for example, user **100** can select button **500**, entitled "Transactions" and network server **104** hands user **100**'s request over to the exchange component. The button and the title can be replaced by any mechanism that can instruct a network server to hand over the consumer's request to the exchange component.

FIG. **5**B illustrates exchange **501**. Exchange **501** comprises network page **505** and point-of-service (POSvc) applications **510**. Exchange **501** also conceptually includes a switching component and an object routing component (described in more detail below). POSvc applications **510** are transactional applications, namely applications that are designed to incorporate and take advantage of the capabilities provided by the present invention. Although exchange **501** is depicted as residing on network server **104**, the exchange can also reside on a separate computer system that resides on the Internet and at the intersection point between the Internet and other networks and on any network where any network intersects with the Internet and has an Internet address. Exchange **501** may also include operator agent **503** that interacts with a management manager (described in more detail below). Exchange **501** creates and allows for the management (or distributed control) of a service network, operating within the boundaries of an IP-based facilities network. Thus, exchange **501** and a management agent component, described in more detail below, under the head-

ings "VAN Switch and Object Routing," together perform the switching, object routing, application and service management functions according to one embodiment of the present invention.

Exchange **501** processes the consumer's request and displays an exchange network page **505** that includes a list of POSvc applications **510** accessible by exchange **501**. A POSvc application is an application that can execute the type of transaction that the user may be interested in performing. The POSvc list is displayed via the graphical user interface component. One embodiment of the present invention supports HyperText Markup Language as the graphical user interface component. Virtual Reality Markup Language and Java™ are also supported by this embodiment. A variety of other graphical user interface standards can also be utilized to implement the graphical user interface.

An example of a POSvc application list is illustrated in FIG. **5**C. User **100** can thus select from POSvc applications Bank **510**(1), Car Dealer **510**(2) or Pizzeria **510**(3). Numerous other POSvc applications can also be included in this selection. If user **100** desires to perform a number of banking transactions, and selects the Bank application, a Bank POSvc application will be activated and presented to user **100**, as illustrated in FIG. **5**D. For the purposes of illustration, exchange **501** in FIG. **5**D is shown as running on a different computer system (network server **104**) from the computer systems of the network merchants or content owners running POSvc applications (computer system **200**). Exchange **501** may, however, also be on the same computer system as one or more of the computer systems of the network merchants.

Once Bank POSvc application **510** has been activated, user **100** will be able to connect to Bank services and utilize the application to perform banking transactions, thus accessing data from a host or data repository **575** in the Bank "Back Office." The Bank Back Office comprises legacy databases and other data repositories that are utilized by the Bank to store its data. This connection between user **100** and Bank services is managed by exchange **501**. As illustrated in FIG. **5**D, once the connection is made between Bank POSvc application **510**(1), for example, and Bank services, an operator agent on network server **104** may be activated to ensure the availability of distributed functions and capabilities.

Each network merchant may choose the types of services that it would like to offer its clients. In this example, if Bank decided to include in their POSvc application access to checking and savings accounts, user **100** will be able to perform real-time transactions against his checking and savings accounts. Thus, if user **100** moves $500 from his checking account into his savings account, the transaction will be performed in real-time, in the same manner the transaction would have been performed by a live teller at the bank or an ATM machine. Therefore, unlike his prior access to his account, user **100** now has the capability to do more than browse his bank account. The ability to perform these types of robust, real-time transactions from a network client is a significant aspect of the present invention.

Bank can also decide to provide other types of services in POSvc application **510**(1). For example, Bank may agree with Car dealership to allow Bank customers to purchase a car from that dealer, request a car loan from Bank, and have the entire transaction performed on the network, as illustrated in FIG. **5**E. In this instance, the transactions are not merely two-way, between the user and Bank, but three-way, amongst the consumer, Bank and Car dealership. According to one aspect of the present invention, this three-way transaction can be expanded to n-way transactions, where n represents a predetermined number of merchants or other service providers who have agreed to cooperate to provide services to users. The present invention therefore allows for "any-to-any" communication and transactions on the network, thus facilitating a large, flexible variety of robust, real-time transactions on the network.

Finally, Bank may also decide to provide intra-merchant or intra-bank services, together with the inter-merchant services described above. For example, if Bank creates a POSvc application for use by the Bank Payroll department, Bank may provide its own employees with a means for submitting timecards for payroll processing by the Bank's Human Resources (HR) Department. An employee selects the Bank HR POSvc application, and submits his timecard. The employee's timecard is processed by accessing the employee's payroll information, stored in the Bank's Back Office. The transaction is thus processed in real-time, and the employee receives his paycheck immediately.

B. Van Switching and Object Routing

As described above, exchange **501** and management agent **601**, illustrated in FIG. **6**A, together constitute a value-added network (VAN) switch. These two elements may take on different roles as necessary, including peer-to-peer, client-server or master-slave roles. Management manager **603** is illustrated as residing on a separate computer system on the Internet or anywhere where the Internet meets or intersects or interfaces with any network. Management manager **603** can, however, also reside on the same machine as exchange **501**. Management manager **603** interacts with the operator agent **503** residing on exchange **501**.

VAN switch **520** provides multi-protocol object routing, depending upon the specific VAN services chosen. This multi-protocol object routing is provided via a proprietary protocol, TransWeb™ Management Protocol (TMP). TMP may incorporate the same or additional security features as the traditional Simple Network Management Protocol, SNMP. It also allows for the integration of other traditional security mechanisms, including RSA security mechanisms, SET1, SET2, etc.

One embodiment of the present invention utilizes TMP and distributed on-line service information bases (DOLSIBs) to perform object routing. Alternatively, TMP can incorporate SSL, s-HTTP, Java™, CORBA, DCOM, the WinSock API, ORB or any other object network layer with DOLSIBs to perform and manage object routing. DOLSIBs are virtual information stores optimized for networking. All information entries and attributes in a DOLSIB virtual information store are associated with a networked object identity. The networked object identity identifies the information entries and attributes in the DOLSIB as individual networked objects, and each networked object is assigned an Internet address. The Internet address is assigned based on the IP address of the node at which the networked object resides.

For example, in FIG. **5**A, network server **104** is a node on the Internet, with an IP address. All networked object associated with network server **104** will therefore be assigned an Internet address based on the network server **104**'s IP address. These networked objects thus "branch" from the node, creating a hierarchical tree structure. The Internet address for each networked object in the tree essentially establishes the individual object as an "IP-reachable" or accessible node on the Internet. TMP utilizes this Internet address to uniquely identify and access the object

from the DOLSIB. FIG. **6**B illustrates an example of this hierarchical addressing tree structure.

Each object in the DOLSIB has a name, a syntax and an encoding. The name is an administratively assigned object ID specifying an object type. The object type together with the object instance serves to uniquely identify a specific instantiation of the object. For example, if object **610** is information about models of cars, then one instance of that object would provide user **100** with information about a specific model of the car while another instance would provide information about a different model of the car. The syntax of an object type defines the abstract data structure corresponding to that object type. Encoding of objects defines how the object is represented by the object type syntax while being transmitted over the network.

C. Management and Administration

As described above, exchange **501** and management agent **601** together constitute a VAN switch. FIG. **7** illustrates conceptually the layered architecture of VAN switch **520**. Specifically, boundary service **701** provides the interfaces between VAN switch **520**, the Internet and the network, telephone companies, wireless systems, cable television networks, and multi-media end user devices such as PCs, televisions or telephones. Boundary service **701** also provides the interface to the on-line service provider. A user can connect to a local application, namely one accessible via a local VAN switch, or be routed or "switched" to an application accessible via a remote VAN switch.

Switching service **702** is an OSI application layer switch. Switching service **702** thus represents the core of the VAN switch. It performs a number of tasks including the routing of user connections to remote VAN switches, described in the paragraph above, multiplexing and prioritization of requests, and flow control. Switching service **702** also facilitates open systems' connectivity with both the Internet (a public switched network) and private networks including back office networks, such as banking networks. Interconnected application layer switches form the application network backbone. These switches are one significant aspect of the present invention.

Management service **703** contains tools such as Information Management Services (IMS) and application Network Management Services (NMS). These tools are used by the end users to manage network resources, including VAN switches. Management service **703** also provides applications that perform Operations, Administration, Maintenance & Provisioning (OAM&P) functions. These OAM&P functions include security management, fault management, configuration management, performance management and billing management for the service network. Providing OAM&P functions for applications in this manner is another significant aspect of the present invention.

Finally, application service **704** contains application programs that deliver customer services. Application service **704** includes POSvc applications such as Bank POSvc described above, and illustrated in FIG. **6**A. Other examples of VAN services include multi-media messaging, archival/retrieval management, directory services, data staging, conferencing, financial services, home banking, risk management and a variety of other vertical services. Each VAN service is designed to meet a particular set of requirements related to performance, reliability, maintenance and ability to handle expected traffic volume. Depending on the type of service, the characteristics of the network elements will differ. VAN service **704** provides a number of functions including communications services for both management and end users of the network and control for the user over the user's environment.

FIG. **8** is a flow diagram illustrating one embodiment of the present invention. A user connects to a Web server running an exchange component in step **802**. In step **804**, the user issues a request for a transactional application, and the web server or cell site or call center hands off the request to an exchange in step **806**. The exchange activates a graphical user interface to present user with a list of POSvc application options in step **808**. In step **810**, the user makes a selection from the POSvc application list. In step **812**, the switching component in the exchange switches the user to the selected POSvc application, and in step **814**, the object routing component executes the user's request. Data is retrieved from the appropriate data repository via TMP in step **816**, and finally, the user may optionally continue the transaction in step **818** or end the transaction.

The Present Invention Applied to Internet Commerce

The goal of Internet commerce is to effect a valid financial transaction from any access point on the net with any other point on the net, spanning both public and private enterprise networks, in real-time, interactively.

Achieving this goal of true, real-time, N-way transaction interactivity requires a combination of:

(1) a many-to-many transaction interface, (2) multi-media archival and retrieval management, and (3) a viable switching model, all with the following attributes:

robust
secure
reliable
high-performance
predictable
configurable
portable
compliant with standards (de facto or de jure)

Currently, a user with a PC and a network browser can access the Web page of various merchants. The present invention extends this one-way browsing capability to N-way, real-time, interactive communication, beyond the Web into IP networks and other types of heterogeneous networks as described above. The present invention brings a remote cash register, telephone and mailbox into the end user's terminal. It reaches everybody and everywhere, much as the telephone does.

The present invention extends the same paradigms as are traditionally used to manage routers and network devices, to interactively reach and manage information. The present invention invigorates the interactive transaction paradigm by offering scalability, robustness, and flexibility designed to be deployed across an entire value supply chain from original manufacturer to point of purchase.

The present invention uses an N-tier Manager-Agent model to implement the N-way communication capability. One part of the present invention is a value—added network application that resides at the transaction Network Entry Point. The other part of the invention resides at remote IP nodes. The present invention includes an N-way interactive object router that provides the link between the consumers, the cash source, and the service provider.

An electronic commerce service provider, such as a Bank, will have to operate like a mini-Telco in order to provide services, such as business-to-business electronic commerce, or homebanking, via the Internet.

Thus, the following system requirements are needed:

security, so as to avoid fraudulent transactions and invasions into the service provider's database.

unified numbering plan to identify each station by a unique address, that is convenient and readily understandable by other stations connected to the network.

uniquely identify, retrieve and route dynamically changing information elements that have to be accessed remotely, using multi-media, object routing.

The Object Router

The Object Router includes a library providing support for the application programming interfaces (APIs) to remotely access an object, its data and its functions in an object network. This interface provides two types of a class: a skeleton, which is the functionality of the object and its stub which allows remote access of the same object. It will be apparent to those of ordinary skill in the art that the items defined below represent an implementation of the preferred embodiment. Other embodiments will be apparent such as non-C++ and non-Java implementations and implementations such as within a DSP chipset.

Glossary

Item Its definition.

Abstract Class A C++ class which does not have all virtual functions defined.

API Application Programming Interface

Class A C++/Java data structure definition which defines both the data and functions.

Interface A Java term similar to the C++ Abstract Class.

Meta A "meta"-compiler translates a higher level "meta"-language (WebX) from the "meta"-file into a lower-level language (C++) output file for and before giving to a traditional compiler.

Object A C++/Java data structure instance, which is defined by a class.

TCL Tool Command Language, developed by Ousterhout at Berkeley.

Architectural Overview and Key Features

The Object Router of the present invention allows the user to specify functions which can be executed remotely. The basic operation uses objects and class information, instead of simple, unrelated functions and data. The arguments to the function is specified in the meta file and the type of argument passing (by value or by reference) can also be specified.

The Object Router allows for new data types to be constructed, using the basic data types of the programming language used in the preferred embodiment: int and String. Single simple inheritance classes can be constructed and then used as data members or pointer within another meta class.

Also, the Object Router allows for a transparent remote or local operation. The programmer does not need to know whether the object is "local" or "remote". The interface automatically determines this and will allow for the correct operation based on the type of object.

Requirements

There are several requirements for the Object Router to provide a clean and transparent operation of local and remote object. FIG. **9** illustrates these basic requirements and operations layers of the object layer.

A. Object Identity

This is needed for the Object Router to determine if the object is local (skeleton) or remote (stub). This is also needed to determine argument parameters and object serialization. This feature is provided in the preferred embodiment by the WxObject in the wx.lib library. Each new class "XYZ" must add a XYZ_ClassID, XYZStub_ClassID and XYZSkel_ClassID to the Wx/ClassID.h file.

B. Network Line Protocol/TCP

This is needed to communicate with the remote computer using some known alphabet. For the Object Router, WxRemotePDU provides this protocol data unit (PDU) and transmits the data using the RWCollectable data serialization.

C. Data Serialization

To transfer data between computers, the parameters and the objects themselves need to be sent over the network. This is the serialization of the data onto a network stream. The framework for this is provided by the persistence nature of RWCollectable. The code is generated by the meta compiler in the skeleton.

D. Data Marshaling

To ensure that all data is passed as arguments to a method requires the user to check all parameters and throw an exception, if there is something missing or wrong. This is provided by the meta compiler in the stub and skeleton.

E. Thread Rendezvous

To block a thread during network transmission and reception, the object must use a different thread to perform the actual network activity and control the calling thread status. This allows calling a remote object to behave similar to a local object. This is performed by the Rogue Wave RWCondition class in the WxRemotePDU class.

F. String Execution

This allows for a method to be called from an ASCII string representation. This is useful since it provides a simple, but unique means of calling a class method. This can also be used by the programmer directly. This data marshaling is created by the meta compiler in the base class.

G. Reference Counting

By maintaining reference counts in both the local and remote objects, time consuming malloc's and free's are avoided as well as confusing details of which function is responsible for deleting which object. This is provided by the Object Router in dealing with WxRemoteObject types. Likewise, the programmer must use reference counting. In other words, the user cannot destroy a WxRemoteObject child using delete (an exception is thrown, if this is tried), but rather the member function ol_unreference(). Also, if a user stores a copy of a WxRemoteObject child, it must indicate this by calling the method ol_reference(). Otherwise, the object may be destroyed by someone else. This interface is provided by WxRemoteObjectInt.

H. Abstract Base Class

To operate on an object without knowing its location (local or remote) requires an abstract class which forms as the base class. This is the base class, which is the parent of both the stub and skeleton. This is an abstract object (cannot be instantiated alone) and is created by the meta compiler.

I. Stub/Skeleton Multiple Inheritance

The client and server classes are inherited from the abstract base class and from their respective object layer classes. This allows them to inherit the functionality of the base class as well as the object layer routines. This is created by the meta compiler.

J. Meta Compiler

All of the above features need to be coded by hand unless there is some automated mechanism to do this. The Meta compiler takes a definition file and creates (using TCL or other encoding) the above requirements: Object Identity, Data Serialization, Data Marshaling, String Execution, Abstract Base Class, and the Stub/Skeleton Multiple Inheritance. If one were to code all of this by hand for even the simplest class, there would be too many errors to even manage. The meta compiler (rme2c) eliminates these errors.

K. User-defined Data Model

This is the user created set of classes built around and on top of the Object Router APIs. This is the basis of the object layer to the next level up the software layers. This foundation appears local to the programmers using the object layer even though it may be remote.

Components

There are only certain parts of the Object Router, which the programmer needs to be aware of. The most important is the base class WxRemoteObject. As shown in FIG. **10**, there is an interface specification WxRemoteObjectInt which is inherited by both the object layer objects and the data model objects. For purposes of example, WxName is a new data model object, which contains one data member "name" which is a String. The meta compiler automatically creates two access components for this data member: get_name and set_name. In addition, the meta compiler also creates the local and remote versions of this data object: WxNameSkel and WxNameStub, respectively.

WxRemoteObjectInt

The abstract base class not only contains all of the member methods and data access components, but it contains support functions which provide a uniform interface to behave as a WxRemoteObject. This includes:

- WxStringExecution to execute any method using a simple ASCII string, and Object type data.
- WxLock, which provides a thread synchronization mechanism.
- WxFlags, which provides a simple and consistent boolean flag variables.
- reference counts, which allow WxRemoteObjects to be shared without ownership concerns.
- conversions between Object, WxRemoteReference, WxRemoteStub and WxRemoteSkel types.

WxRemoteSkel (Server)

The skeleton represents the server side of the Object Router. This is a class, which is derived off the abstract base class WxName (in the diagram) and WxRemoteSkel. All of the methods (except data access components) must be defined by the programmer for the skeleton since this is the actual object. Whenever a real instance of the class is created, it will have the suffix "Skel".

WxRemoteStub (Client)

This is the client (or remote) side of an object. This is similar to the skeleton because it too is derived off the abstract base class WxName and an object layer class WxRemoteStub. However, there is nothing for the programmer to define for this class. All methods are generated by the meta compiler.

Meta Compiler

This is a TCL program in the preferred embodiment, which creates the three classes: abstract base, stub and skeleton. For each class an interface file (header file .h) and a C++ definition file (source code .cpp) is created. It will be apparent to those of ordinary skill in the art that the implementation defined below represents an implementation of the preferred embodiment. Other embodiments will be apparent such as non-TCL, non-C++ and non-Java implementations and implementations such as within a DSP chipset.

Style Issues

Prefixes

The StringExecutionInt class prepends all of its member functions with "se_". Likewise, the object layer classes all prepend their member functions with "ol_". This is needed to avoid any name-bashing with derived classes which are built using the base classes created by the meta compiler.

Member Data

In addition, in the skeleton, all member data is prefixed with "_" to remind the user that the data is local to the object and usually protected.

Synchronization

Any modification of local member data within a skeleton should be guarded by ReadLockGuard or WriteLockGuard when accessed since other threads may be sharing this data.

Stubs and Skeletons

All skeletons are suffixed with "Skel" while all stubs are appended with "Stub".

Value and References

When passing WxRemoteObject derived data, it is also passed with a pointer. However, since it is unclear from this alone who has ownership of the data, the suffixes "_ref" and "_val" are added by the meta compiler to indicate if the data is passed by value or reference. If it is passed by reference, the function then will return a pointer, which has had the reference count incremented. If passed by value, the data is copied from the original source and the receiver should unreference this with DeleteObject.

To indicate if a data member is passed by reference, append an asterisk "*" to the data type in the "data" declaration section of the Object Router meta file. This is also true for return types in the "method" section and as arguments to methods.

const Whenever data is passed by value into a function, it will be proceeded with "const" to signify that the object is not to be changed.

Wx/ClassID.h

This header file contains all ClassIDs for the base classes and their stubs and skeletons. The programmer should place the object ID in this before running the Object Router meta compiler.

CC and HH files

These files are included from the Object Router meta file to add additional functionality to the base, stub or skeleton. If, for example, a function void xyz() needed to be added to the XYZ skeleton class, then add to the XYZ.rme file:

```
include hskel
  void xyz();
endinclude
include cskel
#include "XYZSkel.cc"
endinclude
```

This will then include the simple declaration "void xyz()" into the header for the skeleton and also include the definition for xyz() from the XYZSkel.cc file. The advantage to using the suffix "cc" or "hh" over "cpp" and "h" is because the Object Router meta compiler uses those suffixes for the final XYZ files.

CPP and H files

These files are automatically generated by the Object Router meta compiler for the base, skel and stub. If the programmer wishes to add information to these files, see the examples above. The programmer should not, however, edit the cpp or h files directly. The programmer should be able to fix anything by modifying the rme file and recompiling.

Strings

Strings used in the object layer may be passed either by value using RWCString and "const char*" or by reference using the RWCollectableString (aka String). In some cases, the programmer knows which version is most desirable: pointer or static object. Based on programmer need, the programmer can choose either the function foo() which returns the string by value or foo_ptr(), which calls the same function but returns a copy of the string on the heap as a pointer.

Data access components

For each data member, two data access components are automatically created for the data: "get" and "set". There, however, are complications to these, based on the type of data.

A. Integer

This is the simplest and creates member functions int get_xyz() const and void set_xyz(int).

B. String

This is mentioned above and creates three methods: RWCString get_xyz() const, String *get_xyz_ptr() and set_xyz(const char*).

C. WxRemoteObject by Value

This creates two functions: XYZ* get_xyz_val() const and void set_xyz(const XYZ*).

D. WxRemoteObject by Reference

This also creates two functions: XYZ* get_xyz_ref() const and void set_xyz(XYZ*). This also assumes that the "set" function will retain a copy of the object.

Meta Compiler

Usage

This is used by running the command:

rme2c <classname>

where the classname is the base class (such as Account). The Account.rme file must be in the same directory as well as other parent definitions.

The Object Router TCL files are found under Wx/Util. These files parse the description file and produce the six C++ output files.

Syntax

In the preferred embodiment, the syntax must be adhered to closely. Blank lines and lines beginning with a pound sign "#" are considered comments.

The following words are reserverd:

1. include [c*,h*,j*,*base,*skel,*stub]

This adds verbose code to be added to one of six files: cbase, hbase, cstub, hstub, cskel or hskel.

The programmer can also specify all "c" files or "h" files. The programmer can also specify "base", "skel" or "stub" files.

2. endinclude

The ends the verbose inclusion.

3. header <class>

This indicates that the compiler will wait for the header section to complete.

4. beginclass <class> [<parent>]

This indicates the class name and any parent.

5. begindata

This signals the begin of the data section

6. data [<perm>] <type> <name>

This is a data definition statement.

7. enddata

This is the end of the data section.

8. beginmethod

This is the beginning of the method section.

9. method [const] [<perm>] <return_type> <name> [{<arg1_type> <arg1_name>}, . . .]

This is a method definition.

10. endmethod

This is the end of the method section.

11. endclass

This is the end of the class definition. It should be the last statement in the file.

Operational Overview

The simplest way to introduce the operation of this object layer is through an example.

Scenario

Referring to FIGS. **11** and **12**, consider a sample bank in which there is a main bank object "BofA" and several bank Accounts "bob" and "joe". On the remote client side, the programmer first must establish contact with the remote server and then receive the Bank object on the client side as a stub. Once the stub is received, the client can then look up accounts just as if the programmer was local to the skeleton on the server side. In fact, the programming code is the same and can be reused on either the client or server side. In addition, the programmer does not need to know where the actual object resides: locally or remote.

Object Router Data

The programmer must first create a definition file describing the WxBank and WxBankAccount objects. This is written in a simple language, which is parsed by the rme2c meta compiler of the present invention.

WxBankAccount

The WxBankAccount file is written as:

```
include cskel
# INCLUDE ANY ADDITIONAL C CODE FOR THE
  SKELETON
#include "WxBankAccountSkel.cc"
endinclude
beginclass WxBankAccount
begindata
# PUT MEMBER DATA HERE
data int balance
end data
beginmethod
# PUT MEMBER METHODS HERE
method void deposit {int x}
method void withdraw {int x}
endmethod
endclass
```

This class contains not only methods but also data. In this case, the data is simply an integer describing the amount of money the account holds. The two methods: deposit and withdraw, simply increment and decrement this amount, respectively.

```
void WxBankAccountSkel::deposit(int x) {
 WriteLockGuard lock(wxlock());
 __balance +=x;
}
void WxBankAccountSkel::withdraw(int x) {
 WriteLockGuard lock(wxlock());
 __balance -=x;
}
```

Notice that the programmer should add the lines:
WriteLockGuard lock(wxlock());

to provide thread locking on this object. It is as simple as copy and paste to provide this feature for each method. If the method is const and only accesses data, use
WriteLockGuard lock(wxlock());

and of course, if the programmer locks the method, he/she cannot call any other locked method which includes any object-layer defined data access components.

The above file (WxBankAccountSkel.cc) defines the skeleton methods. All the stub methods are defined by the rme2c meta compiler.

WxBank

For the Bank object, the Bank.rme file would look like:

```
# INCLUDE ANY ADDITIONAL C CODE FOR THE SKELETON
include cskel
#include "WxBankSkel.cc"
endinclude
# INCLUDE ALL H FILES FOR DATA TYPES USED IN ALL C FILES
include c
#include "WxBankAccountSkel.h"
#include "WxBankAccountStub.h"
endinclude
beginclass WxBank
# PUT MEMBER DATA HERE
begindata
enddata
# PUT MEMBER METHODS HERE
beginmethod
method const WxBankAccount* getAccount {int id} {int pin}
endmethod
endclass
```

This file, when processed by rme2c will create six files: WxBank.h, WxBank.cpp, WxBankStub.h WxBankStub.cpp, WxBankSkel.h and WxBankSkel.cpp. These six files describe the operation and remote execution of the WxBank object. Since there is no data, there will be no data access components generated. The method "getAccount" requires some explanation. The line below defines this method:

```
method const WxBankAccount* getAccount {int id} {int pin}
```

The first keyword "const" identifies that this method will not change the object data. The next keyword is the returned object "WxBankAccount*". The asterisk identifies that the object will be passed by reference. The method name is next. Then, all of the parameters follow. In this case there are two. Each parameter is enclosed in braces and contains the data type and the parameter name. In this case there are two integer parameters with name id and pin. The programmer, however, must still describe any additional functionality for the operation of this object and the definitions of the skeleton methods (contained in the WxBankSkel.cc file). The WxBankSkel.cc file would contain:

```
WxBankAccount* bob = 0;
WxBankAccount* WxBankSkel::getAccount_ref(int id, int pin) const {
 if(bob) bob->ol_reference();
 return bob;
}
```

This example is a very simple one in which getAccount will always return bob's account. There are two things to notice.

1. The actual method name is "getAccount_ref" with "ref" appended since this method will return an object by reference.
2. But before simply returning the global variable bob, we must increment its reference count since getAccount is passing a new reference.

Example Client/Server

The client and server are now simple. The server side creates the skeletons and registers them in the name server:

```
extern WxBankAccount* bob; // global used by Bank::getAccount()
```

```
void main(int argc, char* argv) {
 RWWinSockInfo winsock; // initialize the socket library
 WxRemoteConnectionServer s; // create the socket server
 WxBank bofa; // create a bank
 WxBankAccount joe; // create joe's account
 joe.set_balance(0); // with a $0 balance
 bob = new WxBankAccount(); // create bob's account
 bob->set_balance(10000); // with a $100 balance
 bob->deposit(20000); // then, deposit $200.
 // register bofa with a global name --- after everything else is done!
 bofa.set_ol_name(new String("BofA"));
 // start the connection server receiver
 RWThread server =
  rwMakeThreadFunction(
   s,&WxRemoteConnectionServer::run,(RWBarrier*)0);
 serverstart();
 server.join();
}
```

The client program is just as simple:

```
// create a global function which is called from a Rogue Wave thread.
void async() {
 WxRemoteConnectionMonitor monitor;
 WxRemoteClient* local = monitor.client("localhost");
 WxBank* bofa = LOOKUP(WxBank,"BofA",local);
 WxBankAccount* bob = bofa->getAccount(10,20); // arguments are
 dummy cout << "bob's account balance is (should be 30000):"
   <<bob->get_balance() <<endl;
 bob->withdraw(5000); // withdraw $50.
 cout << "bob's new balance is " << bob->get balance() << endl;
}
void main(int argc, char* argv) {
 WxRemoteObject: :initializeStringExecutionTables();
 WxBankSkel::wxClassHierarchy();
 WxBankStub::wxClassHierarchy();
 WxBankAccountSkel::wxClassHierarchy();
 WxBankAccountStub::wxClassHierarchy();
 RWWinSockInfo winsock;
 // start the Rogue Wave thread - and wait until it exits
 RWThread thread = rwMakeThreadFunction(async);
 thread.start();
 thread.join();
}
```

The major benefit demonstrated above is that the programmer does not have to know, nor care, whether the object "bob" is local or remote.

Architecture Details

Design Considerations

There are a number of trade-offs and considerations taken into account during the creation of the object layer. These are briefly described below:

Class Overview

Referring to FIG. 13, this section describes in more detail the function and purpose of each class in the Object Router of the preferred embodiment of the present invention.

WxRemoteObjectInt

This is the interface that all remote objects must have. It contains several abstract class definitions, including WxStringExecutionInt, WxLock and WxFlags. Plus, it defines several methods used for all remote objects.

RWBoolean ol_isValid() const

This should be tested if the programmer does not know if this object is local or if the connection is established. This will return TRUE if the object is local or it has a connection to the remote object.

unsigned get_ol_referenceCnt() const

This returns the number of pointers outstanding for this object. Typically, if garbage collection is enabled, this object will automatically be destroyed when the referenceCnt reaches zero.

WxReferenceId get_ol_referenceId() const

This is the remote referenceId for this object. This WxReferenceId uniquely tags an object instance on the server for the established connection. This is not a well-known name in the sense that it is not guaranteed to be the same with a difference connection.

unsigned ol_reference() const

This increments the number of references outstanding. This needs to be performed whenever a new copy of the pointer is stored.

void ol_unreference() const

This decrements the reference count and should be called instead of delete.

Object* ol_object() const

This simply type casts this instance to an RWCollectable pointer.

WxRemoteStub* ol_stub() const

This will return a stub for this object. If this object is local, it will create a stub. If this is already a stub, it will increment the reference count.

WxRemoteSkel* ol_skel() const

This will return a skeleton for this object. If this is a skeleton, it simply increments the reference count. If this is a stub, it will create a new skeleton, copy the data and return it.

WxRemoteReference* ol_remoteReference() const

This will create a WxRemoteReference object which is only used for serialization.

WxRemoteObject

This is the actual first level implementation of the above interface and adds String Execution to the above functions. From this object, all of the Object Router objects are derived.

WxRemoteReference

This is a type of network "pointer" which indicates where the actual skeleton object resides. It contains the following data:

RWInetHost host

intport

RWClassID classID

WxReferenceId referenceId

Port and host are used to uniquely specify the socket for the WxRemoteConnectionServer. The referenceId is used to uniquely specify which object on the WxRemoteConnectionServer this points to. The classID is simply used to construct a local stub object.

WxRemoteStub

All stubs are derived from this object and the abstract base object for the class. This object provides some interfaces to the Object Router library, which is used by the meta compiler.

WxRemotePDU*_ol_execute(WxRemotePDU* pdu) const

This will block until ol execution is finished. It will take the pre-formatted PDU.

WxMarshalId_ol_send(WxRemotePDU* pdu) const

This is a non-blocking remote execution, which returns a WxMarshalId, which can be used to receive the result.

WxRemotePDU*_ol_peek(WxMarshalId id) const

This checks if the PDU id is returned from execution.

WxRemotePDU*_ol_receive(WxMarshalId id) const

This blocks until the PDU is returned.

WxRemoteClient*_ol_connect() const

This ensures the connection to other side is established.

WxRemoteSkel

All skeletons are derived off this object and the abstract base for the class. This object provides the interface ol_methodPDU() for the meta compiler to the Object Router.

WxRemotePDU

This is the actual data packet sent across the network. The data in this are:

WxMarshalId id

This is the PDU packet number. This is a monotonically increasing integer to uniquely identify the packet.

unsigned flags

Option flags to modify the execution of this protocol. The current options are:

Syn—this will perform synchronous execution of the packet at the server (no threads).

NoMarshal—this is a unconfirmed execution similar to UDP.

Log—this will (in the future) log this request

Response—this indicates that the PDU is a response

Val—this indicates that the result should be a value, not a reference.

unsigned type

This is one of several known protocol operations:

Disconnect—close the connection between WxRemoteClient and WxRemoteServer

Error—an error occurred in processing the request

Result—a packet containing the result of a request

Lookup—a request to find a WxRemoteReference based on a well-known name in the WxRemoteNameServer

Ping—a request for the server to send a Pong back.

Pong—a response from the server to the client to a Ping

Method—a request to execute the command on the server

Unreference—a request to decrement a reference count.

Reference—a request to increment a reference count.

RWCString cmd

This is the ASCII string command to execute on the remote server. This is the "name" in a Name-Value pair.

WxReferenceId referenceId

This is the object WxReferenceId on the server to uniquely the object of this PDU.

Vector* data

This is the data for a method execution. This is the "value" in a Name-Value pair.

WxRemoteConnectionServer

This is the main class on the server side of the Object Router. This is the entry point into the system for a WxRemoteClient requesting access. This is similar to a phone operator who directs incoming calls to the correct person. That person, in this analogy, is the WxRemoteServer.

WxRemoteConnectionMonitor

This is the main class on the client side of the Object Router. This is the entry point into the system for a connection and to create a WxRemoteClient for a particular destination. This is similar to a phone operator who directs outbound calls to the correct person. That person, in this analogy, is the WxRemoteConnectionServer.

WxRemoteServer

This is server side of the client-server communication channel that processes each inbound request. The WxRemoteServer will spawn a new thread for each WxRemotePDU method packet. There is always one WxRemoteServer for a

WxRemoteClient.

WxRemoteClient

This is the client side of the client-server communication channel that sends each outbound request and rendezvous' with the inbound response. There is always one WxRemoteClient for a WxRemoteServer.

WxRemoteError

This is the class, which is thrown by both the client and server side whenever an error is detected. All of these are fatal and are non-recoverable.

WxRemoteException

This is the class, which is thrown by both the client and server side whenever an exception is detected. None of these are fatal and the user should provide recovery code.

Timing

Referring to the timing diagram illustrated in FIG. **14**, the timing diagram shows the timing between all of the objects and the threads operating on them in the preferred embodiment. This timing diagram represents the startup of the object-layer on both the client and server sides. It also demonstrates a lookup("root") call from the client and a later get_name() method call on the root object.

The vertical lines in FIG. **14** represent the objects (data and functions) used in the client and server sides. The horizontal lines represent the threads in the operating system—different threads have different line patterns. There are two threads shown on the client side (left) and there are three on the server side (right). Also, the dotted "write" line represents TCP traffic between the two machines (client and server).

System Requirements

In the preferred embodiment of the present invention, the following conventional computer components may be used to implement the present invention.

Hardware

Intel brand personal computer (PC), Unix workstation or Unix server, or network appliance or device on a DSP chipset.

Software

In the preferred embodiment, the software is compiled under Windows NT3.51 and 4.0 using the Microsoft Visual C++ compiler 2.1. This requires the wx.lib, and the Rogue Wave libraries: Tools++, Net++and Threads++, Unix, WinNT server, or DSP chipset. It will be apparent to those of ordinary skill in the art that the system items defined above represent an implementation of the preferred embodiment. Other equivalent embodiments will be apparent to those of ordinary skill in the art.

DOLSIB Architecture Introduction

The present invention also includes a dynamic DOLSIB, the Distributed Online Service Information Base of the Object Router. The following sections cover an overview of the architecture for the DOLSIB, as the enterprise needs to be customized for each Merchant. To accomplish this, a uniform interface is provided which will allow the Merchant, to provide a customized interface for the business or management objects using a simple Extended Finite State Machine (ESFM) or DOLSIB language. A library provides the core parser and interpreter for the DOLSIB. It also is the base class for business and management objects, which will interface with the enterprise interface.

1. Glossary

Item Definition

EFSM Extended Finite State Machine

EISM Enterprise Interface State Machine

FSM Finite State Machine

DOLSIB Distributed On-Line Service Information Base

state The set of values describing the current position of the machine if it has memory.

transition The action and state change performed by the machine after receiving an event

event The inbound trigger which causes the machine to perform some transition

action The output of the machine as a result of a transition

diagram A complete finite state machine description containing states and transitions.

2. Architectural Overview

The architecture of a business object in the preferred embodiment requires four parts:

1. The Extended Finite State Machine (EFSM) DOLSIB in the CoreBusinessObject or Management Object (C++)
2. The Object Router interface for the business or management object to the DOLSIB. (C++)
3. The enterprise interface protocol (specification)
4. The DOLSIB instructions for the business or management object (EISM)

The first item (DOLSIB and CoreBusinessObject or management object) is built once. This is part of the Object Router library. The second item is built as a common business object and should be generic enough to be configurable for different merchants. The third item is a specification that must be written by the merchant for his own enterprise interface. The fourth is configurable during runtime for different business or management objects.

The following sections discuss the DOLSIB and CoreBusinessObject or management object, as well as show an example BankAccount object which has two different merchant enterprise interfaces.

DOLSIB

FSM

As shown in FIG. **15**, the simple Finite State Machine (FSM) consists of two items: a set of states S and their possible transitions T. A transition t consists of an initial state ts, and event e which triggers an action a, and a final state tf. The example illustrated in FIG. **15** consists of two states S={A, B} and two transitions T={t1, t2 }. The transitions can be described as t1=(A, X, Y, B) and t2=(B, U, V, A). The transition t1 from state A to state B is triggered by an event X and causes an action Y. Likewise, the transition t2 from state B to state A is triggered by an event U and causes an action V. If the finite state machine is in state A and receives any event besides X, it will remain in state A. Only when a valid event occurs (by having a defined transition) does the FSM change its state.

In a formal FSM, the set of states is indeed finite. However, often an extended FSM, called the dynamic DOLSIB, is needed. Here the states are not finite, per se; there exists a number of finite state "blocks" on the diagram, but there are also global variables which may store values which may take on an infinite number of possibilities. This adds another dimension to the FSM and makes the whole system have an infinite number of "states".

As an example, consider a simple counter that receives an input trigger and outputs an integer one greater than the previous output. The input event is anything. The output action is the integer "count". In the formal FSM, there would need to be a state for each possible integer value. If the counter had a limit of 10, then there would be eleven states: an initial state and one for each output value (see FIG. **16**). However, if an infinite counter were needed, an infinite number of states would be needed.

However, if an extended FSM or DOLSIB is used, there would only need to be one state. This is the idle (or initial) state. The counter would simply output the count value and increment this value (see FIG. **17**).

The following section describes the architecture of a DOLSIB C-like language and its use in the present invention. For example, the above counter would be written in the DOLSIB language, as shown below:

```
state Idle;
event count;
var value=0;
var str="",
diagram Counter;
Counter (Idle) {
 Idle : count ? value++ -> Idle;
}
```

This allows the user of the DOLSIB to simply describe the events and actions of the machine in an easy to write and read language. The above describes the Counter diagram beginning in the Idle state. If the Counter is in the Idle state (which it always is), then given the count event, it will increment the variable value and output this as an action. The arrow "->" signifies that the new state is to be Idle.

Language

The Enterprise Interface State Machine (EISM) DOLSIB language of the preferred embodiment is a C-style DOLSIB EFSM language similar to ISO's Estelle. The major difference between the two is the language on which they are based. DOLSIB EISM is based on C style conventions, while Estelle is a Pascal language extension.

The language of the DOLSIB EISM has several features listed below.

ESTELLE-like

The ISO Estelle language is defined as a superset of the Pascal language and fits nicely with the language. The DOLSIB EISM language is similar in that it conforms to the syntax of C.

Multiple Diagram

The DOLSIB EISM language provides for more than one state machine diagram (and thus, more than one state machine) to be described and operated by the same script.

Interpreted

The state machine is interpreted allowing for bytecode compilation into a stack machine opcode.

Stack-based machine

The stack machine allows for a simple implementation and compilation of the parsed state machines. This also allows for other languages in the future.

C style language

The C-style language is easy for the vast majority of programmers. But as mentioned above, if this is not useful, another interface may be designed on the stack machine.

Simple ASCII language

The programming language includes a simple ASCII English language equivalent of the C-style convention.

Machine

Referring to FIG. **18**, the state machine described in the preferred embodiment uses the scheme shown in FIG. **18** to create an intermediate bytecode, which is interpreted by the stack machine.

As shown in FIG. **18**, the ASCII input file is written in the DOLSIB EISM language and passed off to the parser which converts this to the byte code. The byte code can then be used to run the stack machine (as a state machine) or to the dump program which creates an object file dump of the instructions and the symbol table.

Parser

The parser will take the input ASCII, parse it for syntax and syntactical errors, and then create a stack machine instruction set for the resulting state machine. The conversion of the state machine diagram into a stack machine saves time for the run time interpreter and does the preprocessing of the symbol table.

Interpreter

The interpreter is now just a simple stack machine, which does receive input events and sends out actions. The stack machine contains a very limited set of instructions, which simply perform the basic arithmetic and conditional chores at run time.

Dump

The dump program is the only debugging tool, which simply prints out the symbol table and the instructions for each stack op code.

Symbols

Symbol names may be any alphanumeric string ([A-Za-z_][A-Za-z_0-9]*) which does not match a keyword. Symbols are always case sensitive.

All symbols must be declared before use and before the first diagram body. Symbols of the same type may be declared within the same command. Any symbols may be a scalar or vector. The length of the vector must be declared with the symbol type. Valid declarations would be:

state xyz[4];

event test;

var x=0, y=1, z=4;

var a="apple",b="banana";

However, the following are not valid symbols:

V/21 contains an illegal symbol "/"

diagram this is a keyword

There are five types of symbols: diagram, action, event, state, var. All symbols have a value when evaluated, and all have a particular function used as an input to the DOLSIB EISM. The following table shows their respective evalutions:

TABLE 1

Evaluation of Symbol types

Type	Evaluates to
diagram	current state id
action	value of the action; default is zero
event	value of the event if current event; zero otherwise
state	non-zero if this is the current state.
var	an integer or string variable

The next table shows their action when entered into the DOLSIB EISM. This is quite different from their action within a program. Every symbol type, which enters the DOLSIB EISM also carries an assignment value which it uses in the following way:

TABLE 2

Assignment of Symbol Types

Type	Assignment Action
diagram	changes the current state of this diagram to the assigned state
action	sets the action value
event	sets the event value
state	ignored
var	sets the variable

The above assignments within the DOLSIB EISM program are valid with the exception of state. This is because the state is a read-only condition of the current system and cannot be modified. If the programmer wants to change the state of the diagram within DOLSIB EISM, one may use the diagram assignment.

For each declared symbol, there exists an integer ID. The integer ID begin with zero (for the variable "nil"). This variable is always declared and may be used as an event, which does nothing but trigger transitions. All other symbols are assigned starting with one in the order in which they are declared in the program. Hence, if one changes the order, the symbol Ids will change. Also, an integer is assigned to each element of a vector. The previous example would have Ids as:

TABLE 3

Symbol Table

Name	Type	ID
nil	Int	0
xyz[0]	state	1
xyz[1]	state	2
xyz[2]	state	3
xyz[3]	state	4
test	event	5
x	var	6
y	var	7
z	var	8
a	var	9
b	var	10

Statements

This section outlines a typical program and the possible statements in DOLSIB EISM. Every program must have all of its symbols declared as one of the five types followed by at least one diagram.

Consider as an example the simple state diagram that counts the number of incoming count events and dumps the count upon receiving the event dump. This state diagram could be written in DOLSIB EISM as:

```
state Idle;
event count,dump;
var n=0;
diagram Counter;
Counter (Idle) {
   Idle: count ? n++→Idle
      |dump ? n(), n=0→Idle;
}
```

Notice that all of the symbols are declared. The state diagram is named Counter and the line

Counter (Idle) {

begins the diagram defintion. The state Idle is placed in parentheses to show that it is the initial state. If no state is declared as the default, the first state in the state diagram is considered the default.

The state transition may be described in many different formats. The one shown above would have the following meaning:

If in state Idle

and event count is seen, then increment n and go back to state Idle

else if event dump is seen, then output n and n to zero, then go back to state Idle.

This state transition may also be written in DOLSIB EISM as:

if Idle and count then n++ enter Idle

else dump then n(), n=0 enter Idle;

This form is more understandable initially and may be preferred. In either case, the keyword "if" is optional and the following table shows which keywords and symbols are interchangable.

TABLE 4

Interchangable Symbols

Symbol	Meaning
: with and	introduces first arc
? then	follows are conditional expression

TABLE 4-continued

Interchangable Symbols

Symbol	Meaning
-> begin	enter signifies which state to ener if conditional is true is true
! else elsewith	introduces next arc

A final variation on the command structure is the ability to specify outputs without the parentheses. The normal meaning of n() would be to output an action which has an ID of n and a value equal to n. One could specify, for example, n(5) which would set the value of n to five and then output n. One may also explicitly output a symbol as:

Idle with dump then enter Idle output n;

instead of the first arc. Notice, however, that the proceeding two statements are interchangable.

BNF Grammar

The grammar is specified similar to BNF form. Brackets "[ ]" surround those items which are optional. A vertical bar is used to show alternatives. Bold symbols are actual keywords

program	:decl diagrams
decls	:decl | decls decl
decl	:type defs;
type	:diagram | state | event | int | action
defs	:def | defs, def
def	:lvalue | lvalue = const
diagrams	:diagram | diagrams diagram
diagram	:diagram_init { lines }
diagram_init:	:diagram_symbol (state_symbol) diagram_symbol
lines	:line | lines line
line	:[if] states with cmds;
with	:with | and | :
states	:state_symbol | states, state_symbol
cmds	:cmd | cmds else cmd
else	:elsewith | else |
cmd	:exprs then acts begin state_symbol [output outs]
then	:then | ?
acts	:act | acts, act
act	:lvalue ([expr]) | expr
begin	:begin | enter | ->
outs	:lvalue | outs , lvalue
exprs	:const | state_symbol | & symbol | lvalue | asgn (expr) | expr ompop expr | expr logop expr | expr arthop expr | NOT expr | -expr
ompop	:LT | LE | EQ | NE | GT | GE
logop	:AND | OR
arthop	:+|-|*|/|%|.
asgn	:lvalue ASSIGN expr | lvalue ++ | ++ lvalue | lvalue -- | --lvalue
const	:"-"[0-9][0-9]*
lvalue	:symbol | symbol [expr]

Core Business Object or Management Object

All business or management objects must interface with the back-end channel to communicate with the enterprise computer. Hence, the core business or management object, off which all of the business or management objects derive, must have embedded in it the FSM to be able to parse the EISM DOLSIB configured with diagrams.

The core business or management object must be remotely accessible and therefore must be integrated within the Object Router. It has the interfaces for the enterprise interface and the FSM. FIG. **19** illustrates the description of an example of a CoreBusinessObject or Management Object.

Example Application (BankAccount)

The back end communication channel must be customized for each customer for their given application. Since this back end channel will be different for each customer, a program capable of handling the different capabilities and customizable features per customer is needed. As illustrated below, consider a simple bank account class, which has a balance query and withdraw and deposit methods to change the account balance:

```
class BankAccount {
  int balance() const;
  void withdraw(int amount);
  void deposit(int amount);
  }
```

Given this object, the programmer can query the account balance from within the object-oriented C++ environment. However, the actual mechanics of talking to the back end may vary from merchant to merchant. Therefore, an intermediate machine needs to connect with the back-end to communicate using a name-value pair protocol, which is modifiable.

Consider two banks, B1 and B2. B1 may query the back-end for a balance of an account by sending the account number and then the word "balance:query". This can be summarized as follows:

1. send("account_number")
2. send(<eid>)
3. send("balance:query")
4. expect("amount")
5. expect(amount)
6. return amount

The other bank B2 may require the account number being set, receiving this confirmation and then send the "balance" query. This can be summarized as follows:

1. send("account-number")
2. send(<account_number>)
3. expect(status)
4. send("balance")
5. expect(amount)
6. return amount

The second bank B2 contains more steps and can have more error conditions. As illustrated in FIG. **20**, both of these cases can also be described using an FSM and therefore can be written using the DOLSIB EISM language to configure the BankAccount class.

The FSM diagram illustrated in FIG. **20** shows B1 being a little more complicated. This is due to the added error transitions. One can view the state diagram of FIG. **20** as an expect script for a modem which sends out requests and expects back responses matching a particular string. Using the DOLSIB EISM language, the diagram of FIG. **20** can be written as follows:

```
* the following are automatically declared by DOLSIB interface:
 state Idle; // default initial state
 state Expect; // waiting on receive == expect_value
 state Found; // default state after receiving expect_value
 state Error; // default error state
 event receive; // indicates the enterprise interface has data
 event timeout; // indicates the timer has expired
 event method; // indicates a method call has started
 action return; // returns from the method call
 action send; // sends data to the enterprise interface
 action throw; // returns from the method call with a throw
 var max_wait =1000; // the default timer value
 var eid=0;  // the enterprise id
var expected_value=""; // waited value
*/
diagram Balance;
Balance(Idle) {
    Error : true ? -> Idle
    Idle
      : method == "balance"?
            timeout(max_wait),
            send(eid),
            send("balance: query")
            expected_value = "amount"
          -> Expect;
    Expect
      : receive == expected_value ?
            timeout(max_wait)
          -> Found
      | receive != expected_value ?
            throw("expected" .. expected_value
       .. "but received"
            .. receive)
          -> Error
      | timeout?
            throw("timeout while waiting for" .. expected_value)
          -> Error;
    Found
      :receive ?
            return(receive)
          -> Idle
      | timeout ?
            throw("timeout while waiting value")
          -> Idle;
}
```

As shown in the above description, often the merchant will be expecting something from the enterprise interface. Instead of creating a new state for each such "expect" string, one can make use of a predefined set of states "Expect", "Found" and "Error". The only state transitions defined are for the "Expect" state. Referring to FIG. **21**, the user must therefore provide the arcs for the Error and Found states.

The diagram illustrated in FIG. **21** shows the defined arcs of the Expect state. The program for this was the same as shown in the previous example for Bank B1;

```
Expect
    :receive == expected_value ?
            timeout(max_wait)
        ->Found
    | receive != expected_value ?
            throw("expected".. expected_value
      .. "but received"
                .. receive)
        ->Error
    | timeout ?
            throw("timeout while waiting for".. expected_value)
        ->Error;
```

For a single expected string, the use is straightforward. However, to use the expect state for more than one string (more than one expected response), one must make use of the "extended" nature of the DOLSIB EFSM. Namely, one must use global variables to store the other dimension of state. This is shown with the second Bank example using the variable "step".

For the other bank (B2), the finite state diagram would be a little different as shown in FIG. **22**. This diagram (see FIG. **22**) is very similar, but does produce a different "name-value" pair protocol. This is described below in the DOLSIB EISM program:

```
diagram Balance;
//two steps:
//1) wait for status.
//2) wait for "balance".
var step = 1;
Balance(Idle) {
    Error : true ? step = 1 ->Idle;
    Idle
      : method == "balance" ?
            timeout(max_wait),
            send("account_number")
            send(eid),
            expected_value = 0
          ->Expect;
    Found
      : step == 1 ?
            expected_value = "balance",
            step = 1
          ->Expect
      | step == 2 && receive ?
            return(receive),
            step = 1
          ->Idle
      | timeout?
            throw("timeout while waiting balance"),
          ->Error;
}
```

These two bank examples show how different merchant back-ends can make use of the same business or management object.

The bank object class structure was shown above. However, since this will be derived off of the CoreBusinessObject or Management Object (FIG. **19**), it follows that this BankAccount object needs an Object Router definition too. This definition is shown below assuming these three methods: balance, withdraw and deposit:

```
beginclass BankAccount CoreBusinessObject
begindata
enddata
beginmethod
method const int balance
method void deposit {int amount}
method void withdraw {int amount}
endmethod
endclass
```

After compiling this Object Router object, one can then add the hooks to provide the method connections to the FSM for this particular business object.

```
int BankAccountSkel::balance() const {
        fsm_event("balance","");
        RWCString result = fsm_result();
        return atoi(result);
    }
```

The balance method simply calls the FSM event "balance" which starts the diagram transition from Idle to Expect

(see FIG. 20). Since all results in the FSM are performed using strings, the string return type must be converted to an integer.

```
void BankAccountSkel::withdraw(int amount) {
        char a[20];
        sprintf(a,"%d",amount);
        fsm_event("withdraw",a);
        fsm_result();
}
void BankAccountSkel::deposit(int amount) {
        char a[20];
        sprintf(a,"%d",amount);
        fsm_event("deposit",a);
        fsm_result();
}
```

These two examples show that the method fsm_result() is called even when not expecting a result. The reason for this is twofold: 1) the thread will block until result is actually called inside of the FSM and 2) an error result will throw an exception within this fsm_result.

Thus, a configurable value-added network switching and object routing method and apparatus is disclosed. These specific arrangements and methods described herein are merely illustrative of the principles of the present invention. Numerous modifications in form and detail may be made by those of ordinary skill in the art without departing from the scope of the present invention. Although this invention has been shown in relation to a particular preferred embodiment, it should not be considered so limited. Rather, the present invention is limited only by the scope of the appended claims.

We claim:

1. A method for providing a service over a digital network, the method comprising:
- sending first display information from a first computer system to a user device, wherein the first display information includes a control associated with a commercial service;
- accepting a first signal in response to a user input to activate the control;
- associating an object identity with information entries and attributes, wherein the object identity represents a networked object;
- storing said information entries and said attributes in a virtual information store;
- assigning a unique network address to said object identity;
- initiating, in response to the first signal, communication between the user device and a second computer system;
- using the second computer system for
  - sending second display information to the user device, wherein the second display information includes a list of at least one commercial service;
  - accepting a second signal in response to a user input to select a commercial service from the list; and
  - completing a commercial transaction relating to the selected commercial service.

2. The method of claim 1, wherein the second computer system includes an object router.

3. The method of claim 1, wherein the second computer system includes a virtual information store.

4. The method of claim 1, wherein the second computer system includes a value-added network switch.

5. The method of claim 1, wherein the second computer system includes a legacy database.

6. The method of claim 1, wherein the control includes a web page control.

7. The method of claim 1, wherein the first computer system is in a first corporate network and wherein the second computer system is in a second corporate network.

8. The method of claim 7, wherein the first corporate network is operated by a first business entity and wherein the second corporate network is operated by a second business entity.

9. The method of claim 1, wherein the second computer system includes a legacy computing system.

10. The method of claim 1, wherein the user device includes a web browser.

11. The method of claim 10, wherein the display information includes web page content.

12. The method of claim 1, wherein the list of at least one commercial service includes a payment service.

13. The method of claim 12, wherein a user is provided with access to a payment electronic back office system.

14. An apparatus for providing a service over a digital network, the apparatus comprising:
- a processor;
- a machine-readable storage device including one or more instructions executable by the processor for
- sending first display information from a first computer system to a user device, wherein the first display information includes a control associated with a commercial service;
- accepting a first signal in response to a user input to activate the control; and
- initiating, in response to the first signal, communication between the user device and a second computer system, wherein the second computer system acts to send second display information to the user device, wherein the second display information includes a list of at least one commercial service; wherein the second computer system further acts to accept a second signal in response to a user input to select a commercial service from the list; and to complete a commercial transaction relating to the selected commercial service;
- associating an object identity with information entries and attributes, wherein the object identity represents a networked object;
- storing said information entries and said attributes in a virtual information store; and
- assigning a unique network address to said object identity.

15. A machine-readable storage device including instructions executable by a processor for providing a commercial service over a digital network, the machine-readable storage device including one or more instructions for:
- sending first display information from a first computer system to a user device, wherein the first display information includes a control associated with a commercial service;
- accepting a first signal in response to a user input to activate the control;
- initiating, in response to the first signal, communication between the user device and a second computer system, wherein the second computer system acts to send second display information to the user device, wherein the second display information includes a list of at least one commercial service; wherein the second computer system further acts to accept a second signal in response to a user input to select a commercial service from the list; and to complete a commercial transaction relating to the selected commercial service;

associating an object identity with information entries and attributes, wherein the object identity represents a networked object;

storing said information entries and said attributes in a virtual information store; and

assigning a unique network address to said object identity.

16. A method for providing a commercial service over a digital network, the method comprising:

using first computing resources operated by a first business entity to send first display information to a user device, wherein the first display information includes first and second controls;

accepting a first signal in response to a user input to activate the first control;

first performing, in response to the first signal, a first commercial service associated with the first control, wherein the first performing is accomplished under the control of the first business entity interacting with the user device;

accepting a second signal in response to a user input to activate the second control;

second performing, in response to the second signal, a second commercial service associated with the second control, wherein the second performing is accomplished under the control of a second business entity operating second computing resources and interacting with the user device;

associating an object identity with information entries and attributes, wherein the object identity represents a networked object;

storing said information entries and said attributes in a virtual information store; and

assigning a unique network address to said object identity.

17. The method of claim 16, further comprising:

exchanging information between the first and second business entities in order to complete a transaction in response to one or more of the signals.

18. The method of claim 16, wherein a three-way transaction is achieved between the first computing resources, the second computing resources and the user device.

19. The method of claim 16, wherein a plurality of computing resources are used, each on a separate corporate network, and each operated by one of a plurality of different business entities.

* * * * *

Exhibit B1



US007340506C1

(12) INTER PARTES REEXAMINATION CERTIFICATE (970th)

United States Patent

Arunachalam

(10) **Number: US 7,340,506 C1**

(45) **Certificate Issued: Oct. 15, 2014**

(54) **VALUE-ADDED NETWORK SWITCHING AND OBJECT ROUTING**

(75) Inventor: **Lakshmi Arunachalam**, Menlo Park, CA (US)

(73) Assignee: **WebXchange, Inc.**, Menlo Park, CA (US)

Reexamination Request:
No. 95/001,129, Dec. 19, 2008

Reexamination Certificate for:
Patent No.: **7,340,506**
Issued: **Mar. 4, 2008**
Appl. No.: **09/792,323**
Filed: **Feb. 23, 2001**

Related U.S. Application Data

(63) Continuation of application No. 09/296,207, filed on Apr. 21, 1999, now Pat. No. 6,212,556.

(51) **Int. Cl.**
G06F 13/00 (2006.01)

(52) **U.S. Cl.**
USPC **709/219**; 709/203; 709/227; 719/313

(58) **Field of Classification Search**
None
See application file for complete search history.

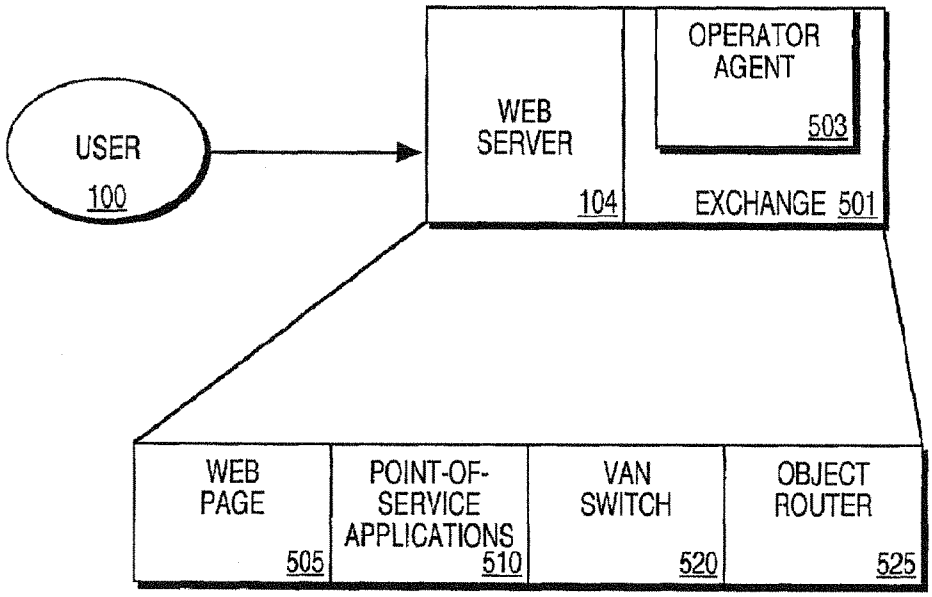
(56) **References Cited**

To view the complete listing of prior art documents cited during the proceeding for Reexamination Control Number 95/001,129, please refer to the USPTO's public Patent Application Information Retrieval (PAIR) system under the Display References tab.

Primary Examiner — Zoila Cabrera

(57) **ABSTRACT**

The present invention provides a method and apparatus for providing real-time, two-way transactional capabilities on the network. Specifically, one embodiment of the present invention discloses a configurable value-added network switch for enabling real-time transactions on the network. The configurable value added network switch comprises means for switching to a transactional application in response to a user specification from a network application, means for transmitting a transaction request from the transactional application, and means for processing the transaction request. Additionally, a method for enabling object routing is disclosed, comprising the steps of creating a virtual information store containing information entries and attributes associating each of the information entries and the attributes with an object identity, and assigning a unique network address to each of the object identities. Finally, a method is disclosed for enabling service management of the value-added network service, to perform OAM&P functions on the services network.



INTER PARTES REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 316

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW.

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

Claims **1-19** are cancelled.

New claims **20** and **21** are added and determined to be patentable.

20. The apparatus of claim 14, wherein the transaction is handed over to an exchange, wherein the exchange manages the connection between the user and the commercial service, wherein the commercial service is an online service operating across the digital network, wherein the digital network is a value-added service network atop the Web.

21. The apparatus of claim 14, wherein the first computer system offering the commercial service comprising access to employee payroll information on a service network atop the Web.

* * * * *