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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 August 2004 (26.08.2004)

PCT

(10) International Publication Number
WO 2004/072831 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number:
PCT/US2004/004519

(22) International Filing Date: 12 February 2004 (12.02.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/365,856 12 February 2003 (12.02.2003) US

(71) Applicant *(for all designated States except US)*: **IPASS, INC.** [US/US]; 3800 Bridge Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **SUNDER, Singam** [IN/US]; 539 Isaac Court, San Jose, CA 95136 (US). **EDGETT, Jeff** [US/US]; 151 S. Bernardo #24, Sunnyvale, CA 94086 (US).

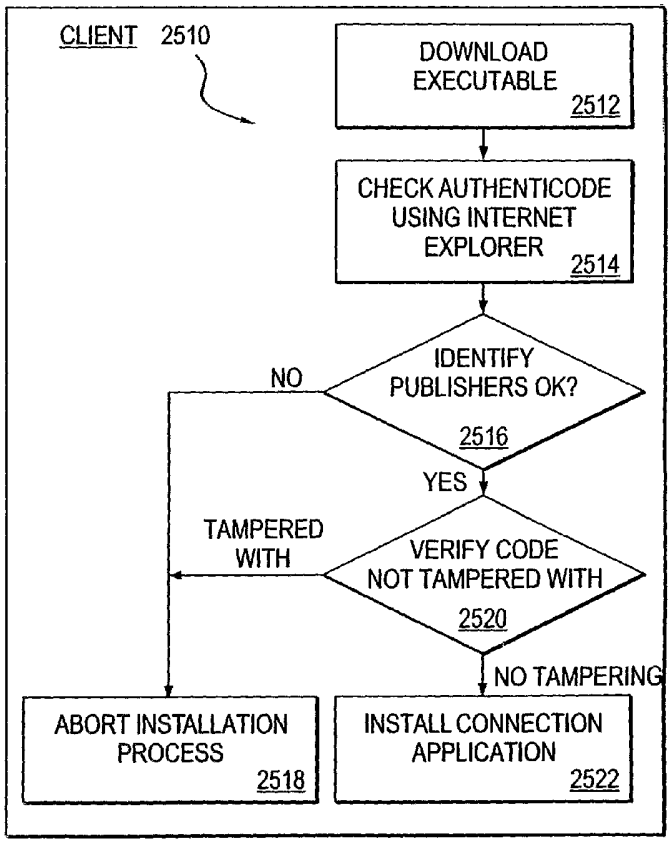
(74) Agent: **MARAIS, Andre**; Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, Seventh Floor, Los Angeles, CA 90025 (US).

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

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

[Continued on next page]

(54) Title: A METHOD AND SYSTEM TO SECURE A CONNECTION APPLICATION FOR DISTRIBUTION TO MULTIPLE END-USERS



(57) Abstract: A method and system is provided to secure a connection application in a multi-party access environment including a plurality of service providers. The method includes generating a customized connection application, and cryptographically signing the customized connection application, the customized connection application being for installation on a client machine. In certain embodiments, the customized connection application is in the form of a self-extracting executable to automatically install the connection application on the client machine. The customized connection application may be signed using Authenticode technology.

WO 2004/072831 A2

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

Published:

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

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

A METHOD AND SYSTEM TO SECURE A CONNECTION APPLICATION FOR DISTRIBUTION TO MULTIPLE END-USERS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part of U.S. Application Serial No. 09/921,959, filed August 2, 2001.

FIELD OF THE INVENTION

[0002] The present invention relates to the field of remote network connections and more particularly to securing a customized connection application.

BACKGROUND OF THE INVENTION

[0003] With the technological developments of the last decade and growing popularity of online commerce, e-mail, online chatting and the Internet in general, the demand to have constant access to these innovative technological ways of communication is rapidly increasing. Some users cannot imagine their lives without the Internet and email; some cannot imagine their lives without being able to buy groceries online. Constant desire to be connected to the informational highway increases demand for reliable, fast, convenient network connection.

[0004] Anyone using current technology has dealt with networks at some point. Being connected to a local network where users share files and data on one server is a common scheme in workplaces. Companies would like their employees to use network connection tools that reflect company's policies, such as session duration limits, usage of particular Point-to-Point locations, etc. However, companies usually obtain network access tools from outside providers. The security of these network access tools may however become compromised.

SUMMARY OF THE INVENTION

[0005] According to one aspect of the invention, there is provided a method to secure a connection application in a multi-party access environment including a plurality of service providers. The method may include generating a customized connection application, and cryptographically signing the customized connection application. The customized connection application may be for installation on a client machine.

[0006] The customized connection application may be in the form of a self-extracting executable automatically to install the connection application on the client machine. In one embodiment, the method cryptographically signs the customized connection application using Authenticode™ technology available from Microsoft™ Corporation. The connection application may then be distributed to the plurality of service providers via a computer network for downloading by the client machine.

[0007] The invention extends to a machine-readable medium for executing any of the method described herein.

[0008] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

FIGURE 1A is a diagram of centralized customization system architecture according to one embodiment of the present invention;

FIGURE 1B is a block diagram illustrating domains of a data model utilized by a customization tool and a phonebook generation tool of the customization system, according to one embodiment of the present invention;

FIGURE 2 is a flow diagram illustrating operation of a back-end of a centralized customization system according to one embodiment of the present invention;

FIGURES 3A and **3B** are flow diagrams illustrating a customization process of building a customized dialer according to one embodiment of the present invention;

FIGURE 4 is a graphical end-user interface presented to a customer to allow the selection of a customer account according to one embodiment of the present invention;

FIGURE 5 is a graphical end-user interface presented to the customer to create or edit a dialer profile according to one embodiment of the present invention;

FIGURE 6 is a graphical end-user interface presented to the customer to allow an input of basic settings according to one embodiment of the present invention;

FIGURES 7A and 7B show graphical end-user interfaces presented to the customer to allow addition of a logo to the customized dialer according to one embodiment of the present invention;

FIGURE 8 is a graphical user-interface presented to the customer to allow specification of dialer connection actions according to one embodiment of the present invention;

FIGURE 9 is a graphical user-interface presented to the customer to allow addition of customer POPs to a dialer phonebook according to one embodiment of the present invention;

FIGURE 10 is a graphical user-interface presented to the customer to allow making of the dialer phonebook according to one embodiment of the present invention;

FIGURE 11 is a graphical user-interface presented to the customer to allow specification of POP filter rules according to one embodiment of the present invention;

FIGURE 12 is a graphical user-interface presented to the customer to allow specification of pricing rules according to one embodiment of the present invention;

FIGURE 13 is graphical user-interface presented to the customer to allow review of customized information according to one embodiment of the present invention;

FIGURE 14 is a graphical user-interface presented to the customer to allow downloading of files according to one embodiment of the present invention;

FIGURE 15 is a flow chart detailing a phonebook generation process performed by a phonebook generation tool;

FIGURE 16 is diagram of system architecture according to one embodiment of the present invention;

FIGURE 17 is a graphical end-user interface presented on a client machine that constitutes a main dialog box of a dialer according to one embodiment of the present invention;

FIGURE 18 is a graphical end-user interface presented on the client machine that allows an end-user to specify dial properties according to one embodiment of the present invention;

FIGURE 19 is a graphical end-user interface presented on the client machine that prompts the end-user for end-user information according to one embodiment of the present invention;

FIGURE 20 is a graphical end-user interface presented on the client machine that allows the end-user to specify settings of the dialer according to one embodiment of the present invention;

FIGURES 21 and 22 show graphical end-user interfaces presented on the client machine that allows the end-user to add and modify bookmarks according to one embodiment of the present invention;

FIGURE 23 is a diagrammatic representation of a number of exemplary protocols and/or components that may be utilized to support various access methods that may be performed utilizing a network connection application, according to exemplary embodiments of the present invention;

FIGURE 24 is a diagrammatic representation of a machine, in an exemplary form of a computer system, for executing a sequence of instructions stored on a machine-readable medium, the sequence of instructions causing the machine to perform any of the methodologies described herein;

FIGURE 25 is a schematic flow diagram of a method, in accordance with one embodiment of the invention, to secure a customized connection application; and

FIGURE 26 is a schematic flow diagram of authentication of the connection application at a client machine.

DETAILED DESCRIPTION

[0010] Although the present invention is described below by way of various embodiments that include specific structures and methods, embodiments that include alternative structures and methods may be employed without departing from the principles of the invention described herein.

[0011] In general, embodiments described below feature a system and a method that secure a connection application (e.g., a dialer) for distribution to multiple end-users. One embodiment of the present invention features a centralized network dialer customization system.

NETWORK-RELATED TECHNOLOGY

[0012] Before describing embodiments of the present invention in detail, it may be helpful to discuss some of the concepts on which the present invention is

based. A component of one embodiment of the present invention is a computer server. Servers are computer programs that provide some service to other programs, called clients. A client and server communicate by means of message passing often over a network, and use some protocol, (i.e., a set of formal rules describing how to transmit data), to encode the client's requests and/or responses and the server's responses and/or requests. The server may run continually waiting for client's requests and/or responses to arrive or it may be invoked by some higher level continually running server that controls a number of specific servers. Client-server communication is analogous to a customer (client) sending an order (request) on an order form to a supplier (server) dispatching the goods and an invoice (response). The order form and invoice are part of the protocol used to communicate in this case.

[0013] Another component of one embodiment of the present invention is Microsoft Foundation Class (MFC), a collection of software structures written in C++ language and which are Microsoft Windows-based classes and which can respond to messages, make windows, and from which application specific classes can be derived. The current invention also utilizes the Remote Access Service (RAS) API, which provides an abstraction layer between the application and the underlying hardware that provides the Point-To-Point Protocol (PPP) connection. RAS is a feature built into Windows NT that enables users to log into an NT-based Local Area Network (LAN) using a modem, X.25 connection or Wide Area Network (WAN) link. RAS works with several major network protocols, including TCP/IP, IPX, and Netbeui.

[0014] Another component of one embodiment of the present invention is a Point-to-Point Tunnel Protocol (PPTP), a new technology for creating Virtual Private Networks (VPN), developed jointly by Microsoft Corporation, U.S. Robotics and several remote access vendor companies, known collectively as the PPTP forum. A VPN is a private network of computers that uses the public Internet to network processing locations. Because the Internet is essentially an

open network, PPTP is used to ensure that messages transmitted from one VPN node to another are secure.

[0015] Yet, another component of one embodiment of the present invention is a Telephony Application Programming Interface (TAPI), an Application Programming Interface (API) for connecting personal computers running Windows operating system to telephone services. TAPI was introduced in 1993 as the result of joint development by Microsoft Corporation and Intel Corporation. The standard supports connections by individual computers as well as Local Area Networks (LAN) connections serving many computers. Within each connection type, TAPI defines standards for simple call control and for manipulating call content.

[0016] Another component of one embodiment the present invention is an Internet Service Provider (ISP). An ISP is a service that provides access to the Internet. For a monthly fee, a service provider gives a customer a software package, username, password and Internet access phone number. Equipped with a modem (e.g., a dial-up, DSL, ISDN or wireless), a customer can then log onto the Internet and browse the World Wide Web (WWW) and USENET, send and receive e-mail, and access a particular network. In addition to serving individuals, ISPs also serve large companies, providing a direct connection from the company's networks to the Internet. ISPs themselves are connected to one another through Network Access Points (NAPs). NAP is a public network exchange facility where ISPs can connect with one another in peering arrangements. The NAPs are a key component of the Internet backbone because the connections within them determine how traffic is routed. They are also the points of most Internet congestion.

[0017] ISPs generally provide a plurality of Point of Presence gateways (POP) in order for a customer to gain an Internet access by making a local call. A POP (point-of-presence) is an access point to the Internet that is associated with a phone number. A connection established via such a POP causes a unique IP

address to be assigned to a machine that accesses the Internet utilizing the established connection. The number of POPs that an ISP has and the number of subscribers are usually used as a measure of its size or growth rate.

[0018] Yet another component in one embodiment of the present invention is a servlet. Servlets are Java applications, which run on a Web server or application server and provide server-side processing, typically to access a database. It is a Java-based alternative to Common Gateway Interface (CGI) scripts, interface programs, usually written in C or Perl, that enables an Internet server to run external programs to perform a specific function. The most important difference between servlets and CGI scripts is that a Java servlet is persistent. This means that once it is started, it stays in memory and can fulfill multiple requests. In contrast, a CGI script disappears once it has fulfilled a request.

ARCHITECTURE

[0019] With these concepts in mind, an embodiment of a system architecture of the present invention can be explored. In one embodiment, the present invention includes customization system 10 and an end-user tool that allows a user to establish a network connection. **Figure 1A** illustrates an exemplary customization system 10 that includes a web server 100, database server 105, a build server 110, and an update server 113. The web server 100 contains a phonebook generation tool 115, responsible for phonebook generation update and customization, and a customization tool 120, responsible for customization of a dialer application (hereinafter "the dialer"). While the exemplary embodiment of the present invention describes the generation, distribution and updating of a customized dialer, it will be appreciated that the dialer is merely an example of a connection application with purposes of establishing a connection between a client and a server computer, or between peer computers within a network. Accordingly, the present invention should not be construed as being limited to the generation, distribution and updating of an application

for establishing a dialed connection over the Public Switched Telephone Network (PSTN), and extends to the generation, distribution and updating of any customized connection application that operates to establish a connection between machines coupled via a network.

[0020] The database server 105 contains a customer database 125, a phonebook database 130, a profile database 135, a phonebook customization database 140, and customer phonebook database 145. It will be appreciated that databases may not be stored at the server machine and the database data may be uploaded to the server machine when necessary.

[0021] **Figure 1B** is a diagrammatic representation of domains of a data model accessed and maintained by the phonebook generation tool 115 and the customization tool 120, according to an exemplary embodiment of the present invention. Specifically, the data model is shown to include the primary components of the customer database 125, the phonebook database 130, and the profile database 135. The data model is also shown to include an access points database 136, and a pricing database 137, which will be described in further detail below.

METHODOLOGY

[0022] A flow chart detailing a method 190, according to an exemplary embodiment of the present invention, of generating and distributing a customized dialer is illustrated in **Figure 2**. At 200 the customization occurs, during which the customer utilizing, in one embodiment, a series of web pages, generated by the web server 100, specifies information (or parameters) for the customization of a dialer that will incorporate the customer's needs. At 205 of **Figure 2**, generation of an executable file takes place upon the customer completing the customization process. The executable file is generated by the build server 110, the description of which follows. At 210 the distribution of the executable file to the end-users or to the distributor, which in turn distributes it

to the end-users, takes place. The above-mentioned back-end processes of the method 190 are described in detail below.

METHODOLOGY: Customization by the Customization Tool 120

[0023] In one exemplary embodiment, the customization tool 120 is a web application developed utilizing HTML, JavaScript, and JavaServlets.

[0024] The customization tool 120 presents a customer of the system 10 with a sequence of web pages that guide the customer through a process of building a customized dialer incorporating the customer's needs. The output of the customization process implemented by the customization tool 120 is a "profile" that defines a customization of a network connection application. Utilizing the customization process, a customer may define multiple customized network connection applications (e.g., dialers), each customized network connection application being described in terms of a profile.

[0025] An exemplary embodiment of a customization process that takes place at block 200 in Figure 2, implemented by the customization tool 120, is described with the reference to **Figure 3A** and **Figure 3B**. Upon the customer logging onto the customization system 10 system, the customer is presented with a sequence of web pages, generated by the web server 100 that facilitate the input of customization information specifying preferences of the customer. A first customization page 301, an example of which is illustrated in **Figure 4**, is generated and presented at 300. The page 301 prompts the customer to select a customer account name under which all the customization information is stored. A partner code, representing an account number, may be automatically displayed after the customer login process. More specifically, the page 301 is utilized only for "channel" customers of a primary customer. The selected customer account name is a coded name for the channel customer for which a customer's dialer is to be generated, and the customization system 10 stores all

customization information entered during the process under the relevant customer account name.

[0026] At 305 the second web page 302, an example of which is illustrated in **Figure 5**, is presented to the customer where the customer is prompted to select between the options of creating a new profile, or editing an existing profile.

[0027] A profile consists of all the files needed by the customization system 10 to create a complete, self-installing package distributable to a customer of the system 10, a distributor or directly to a customer's end-users. Customers may maximize or minimize the identification of the service or organization depending on what is included in a dialer profile. For example, the following features that are described in detail below may be included into a dialer profile: custom corporate logos, connection actions, addition and removal of access points (POPs), and pricing setting.

[0028] The customer is presented with the third web page 303, an example of which is illustrated in **Figure 6**, at 310 giving the customer an option to enter a default authentication domain, which will allow the end-users to enter only a end-user name and password in order to be connected to the network, without specifying a domain name. At the third web page 303, the customer may be prompted for the back-up Domain Name System (DNS) server IP address. The back-up DNS server may be used where a Point-of-Presence (POP), which an end-user has dialed into, does not automatically assign an IP address. In one embodiment of the present invention all POPs in the phonebook database 130 have dynamic DNS addressing. The customer may specify if he/she desires the end-users to have an option of saving their password in order to avoid entering it every time an end-user logs into the system.

[0029] The third web page 303 may also prompt the customer to specify if prices will be displayed next to each dial-in number when the dialer is invoked by the end-user. The customer may also desire to display prices in particular currencies. According to one embodiment of the present invention, the customer

may enter a conversion rate in order for the dialer to display pricing in currency applicable to the end-user's geographical location.

[0030] Phonebook updates are uploaded to the end-user's machine upon establishment of a network connection through the dialer. The customer may, via the third web page 303, specify if it desires the end-users to choose a manual phonebook update instead of an automatic one.

[0031] Some customers may desire to limit network connection sessions of the end-users. The third web page 303 allows customers to specify the maximum connect time that the customer desires the end-users to have. In one embodiment, an unlimited option may be available for the customers to select.

[0032] In one embodiment of the present invention the dialer will be installed on end-users' machines with a default shortcut name. Via the third web page 303, the customer may specify its own shortcut name, for example, the name of the company.

[0033] Upon selection of the options displayed at the third web page 303, the customer at 315 may be presented with the fourth web page 304, an example of which is illustrated in **Figure 7A**, allowing the customer to add a personalized logo to the dialer application distributed to the end-users.

[0034] **Figure 7B** illustrates an exemplary end-user interface 702, generated by a dialer, that displays a selected logo to an end-user when the dialer is invoked.

[0035] In one embodiment, at 320 the customer is presented with a fourth web page 800, an example of which is illustrated in **Figure 8**, allowing the customer to specify the dialer connect actions. Dialer connect actions are additional programs that may be executed at various points when the end-users connect to the Internet utilizing the customized dialer. For example, a connect action may be an automatic establishment of a VPN connection after the end-user connects to the Internet. According to one embodiment of the present invention, the customer may specify connect actions to execute at six different points during the end-user's connection process. Those actions may be a PostConnect action

specifying programs to be executed after the connection is established; a PreConnect action specifying programs to execute prior to the establishment of the network connection; a PreDial action specifying programs to run immediately prior to dialing a point of access number; an OnError action specifying programs to run in case an error occurs; an OnCancel action specifying programs to run when the end-user decides to cancel a connection session; and Disconnect action specifying programs to run when the end-user disconnects from the Internet.

[0036] In box 810 of **Figure 8** the customer is presented with a drop-down menu to select an action type from the list described above to be added to a dialer profile. The description box 815 allows the customer to enter a short description of the programs that the customer wants to be executed. At box 820 the customer may specify the sequence in which the connect action to be executed. In a case where the connect actions are asynchronous, or there is only one action, the sequence of execution is not important. In box 825 the customer may specify the name of the program to be launched at a particular connect action. The customer is presented with a browse feature in order to specify the exact name of the program. The customer may specify the parameters, including the command line parameters, necessary to run the program in box 830. At box 835 the customer may specify that a program does not need to be loaded with the dialer to the end-users' machines. In one embodiment, the programs that need to be run at particular connect actions may be already installed on the end-users' machines. In one embodiment, the customers may select a sequence of connect action to run at the same time (asynchronous mode), or one after the other (synchronous mode) at 840. If the programs are running in synchronous mode, one program must completely finish executing before the next one can be launched in synchronous mode. In one embodiment if an error occurred while executing one of the programs, the connect action to be executed after the program may not be launched. At box 845, the customer may identify the POPs for which the connect

actions should run. In one embodiment, the customer is presented with an option to create additional connect actions or to delete the existing ones.

[0037] In one embodiment of the present invention, the customized dialer may be configured to launch Microsoft's VPN (PPTP) after a successful connection is established. PPTP support may be built into the customized dialer and not require any additional client software.

[0038] At 325 of **Figure 3A**, the customer may add POPs to a phonebook, stored in the phonebook database 130, utilizing the sixth web page 900, an example of which is illustrated in **Figure 9**. In one embodiment, the list of POPs to be added to the phonebook of the customization center 10 may be created through a text editor. Each POP to be added may be identified by the following parameters: a country code that may be represented in a 2-letter code; the POP's region identification number or state identification number; the city in which the POP is located; the area code of the phone number for the POP; the phone number for the POP, without the area code; the maximum analog speed supported by the POP; identification of whether one channel or two channel ISDN is available or if no ISDN is available for the POP to be added; identification of whether Password Authentication Protocol (PAP) is available; identification of whether Challenge Handshake Authentication Protocol (CHAP) is available; the price to be charged for the utilization of the POP; the prefix used for routing the authentication request; the suffix to be used for routing the authentication request; a script name of a file containing a series of commands, parameters and expressions required to establish the connection via the POP.

[0039] At 330 of **Figure 3A**, the tool 120 presents a list of phonebooks that are valid for the customer as per the pricing plan associated to the customer. The list of phonebooks may be presented via a drop-down menu of a web page 1000, an example of which is illustrated in **Figure 10**. These phonebooks contain all the POPs in a service provider network, excluding the POPs filtered as per the filtering value associated to the pricing plan (see block 1540 in **Figure 15**). The

customer can further apply custom filtering and pricing rules to the phonebooks to arrive at their customized phonebooks. For some plans, the tool 120 may generate phonebooks that have price markups. The example web page 1000, shown in **Figure 10**, provides examples of such markups. At 335 the customer is presented with an eighth web page 1102, an example of which is illustrated in **Figure 11**, through which the customer may specify filter rules for various POPs. In box 1100 the customer is presented with a list of the attributes that may be used in filtering the list of POPs presented to the end-users. In one embodiment, the filter rules may be the Structures Query Language (SQL) where clauses. The filtering rules may be generated utilizing a list of the attributes including, but not limited to: country code; the region or state identification of a POP; the city in which the POP is located; the phone number of the POP without an area code; the maximum analog speed supported by the POP; the price of the POP; identification if one channel or two channel ISDN is available or if no ISDN is available for the POP to be added. For example, in order to filter all POPs located in the Russian Federation, a filter rule may specify: Country Code = 'RU', where 'RU' is the 2-letter code for the Russian Federation.

[0040] At 340 the customer is presented with a ninth web page 1200, an example of which is illustrated in **Figure 12**, that allows the customer to specify pricing rules to be applied to the prices of the POPs in the customization system phonebook. Two types of the pricing rules may be available to the customer according to one embodiment of the present invention: the percentage markup or slab pricing. If percentage markup pricing is selected, the system 10 applies a specified markup percentage to the POP price listed in the customization system phonebook. The slab pricing applies a pricing formula specified by the customer to the listed prices in the customization system phonebook. For example, the customer may specify a particular amount to be added to a listed POP price if the listed price is within the customer-specified price range and a different amount to be added if the listed price is outside the customer-specified range. In

one embodiment, the customer may also specify different rules for the POPs currently listed in the phonebook and the POPs that are going to be added to the phonebook in the future. In another embodiment of the present invention, the customer may specify different pricing rules for different countries.

[0041] At 345 of **Figure 3B** the customer is presented with a review web page 1300, an example of which is illustrated in **Figure 13,** that shows the details of the customization process that was performed by the customer. If the customer is not satisfied with the details he or she may edit a dialer profile to make desired changes to the customization. If the customer is satisfied with the dialer profile he/she may click on the Build Dialer button 1302 in order to build a dialer according to the customer-specified customization information. Upon the customer requesting to build the customized dialer, the customization information is sent to the build server 110. The build server 110 generates a self-extracting (or self-installing) executable file that is capable of being distributed to the customer, a specified distributor or directly to the customer's end-users in order to provide the end-users with the Internet access through the customized dialer. In one embodiment of the present invention, upon the end-users connection to the system 10 utilizing the dialer, the build server 110 dynamically adds new files and removes outdated files utilizing the version numbers associated with each file and dynamically generates a self-extracting executable that replaces an outdated end-user's dialer file. This update process is described in more detail below.

[0042] At 350 the customer is presented with the download web page 1400, an example of which is illustrated in **Figure 14.** The web page 1400 contains the list of files that are necessary to publish the customized dialer to the end users. In one embodiment those files are executable installation file generated by the build server 110, a phonebook file containing all the POPs in the customized phonebook, a zip phonebook file containing Perl scripts and data files necessary to generate smaller HTML files per each country, a phonebook file containing

phonebooks in CSV and ASCII format and a Macintosh phonebook file which is in a format compatible with the Macintosh dialers.

[0043] In one embodiment, the customization system 10 utilizes the pricing and access point data maintained by a settlement system that described in detail in a co-pending patent application 09/791,239, titled "A Method and System to Facilitate Financial Settlement of Service Access Between Multiple Parties". The pricing data maintained by the settlement system specifies the method of pricing of a POP according to a particular pricing plan. The customization system 10, in one embodiment, retrieves a contract of a customer and the list of available phonebooks for the retrieved customer pricing plan.

[0044] In one embodiment, the customer may specify the rules for the termination of a connection if it is determined to be idle. The decision to terminate the connection may depend on the specified allowed duration of the idle connection before its termination, on the allowed minimum data transfer rate before the connection is terminated (this may be used to discount certain background traffic, which does not represent real end-user activity), on the allowed time to respond to a dialog box to renew the connection by the end-user before the connection is terminated. In one embodiment the absolute limit may be set on the length of sessions, regardless of the connection activity as described above.

[0045] In one embodiment of the present invention, the customer may require the customized dialer to support foreign languages through the use of external language resources and help files. In one embodiment at runtime, the customized dialer may determine the language of the operating system installed on the end-user's machine and load the associated language resource and help files stored at the end-user's machine. If external files are not found, the customized dialer may use the default language, i.e. English.

[0046] In one embodiment security information, such as end-user password, VPN password, calling card PIN, stored locally on the end-user's system may be encrypted using standard encryption algorithms well know in the art.

[0047] It will be appreciated that the above-described customization process need not be implemented utilizing a series of web pages. In one embodiment the customization may be performed through a software application and the customization information may then be uploaded to the centralized customization tool through a network. Further, the customization process need not be performed by the customer but by any other person authorized to do so.

METHODOLOGY: Update

[0048] In one embodiment of the present invention, the customization tool 120 updates multiple copies of a network connection application (e.g., a dialer) distributed by the customer to the end-users automatically upon each end-user connecting to a network access point. In an alternative embodiment, an end-user may manually invoke the update feature of the customized dialer distributed to him/her by the customer. During the update process, the client dialer contacts the update server 113 and retrieves the list of files and their latest version numbers. The dialer compares the list of files stored locally with the list retrieved from the server 113. If the list and/or the version numbers don't match, the dialer retrieves the affected files from the update server 113. In one embodiment of the present invention, the new build executable and DLL files are downloaded to the client machine and stored in temporary locations due to inefficiency of updating dialer files when the dialer is running. Upon the end-user exiting the dialer, the files on the client machine are updated to the files containing newer information.

[0049] In one embodiment the customer may not want the end-users to have access to the latest changes until, for example, the testing of all the new POPs is

performed. In such a case the customer may instruct the customization system 10 not to update the dialer automatically unless instructed otherwise.

METHODOLOGY: Phonebook generation

[0050] **Figure 15** is a flow chart illustrating a method 1500, according to an exemplary embodiment of the present invention, that is performed by the phonebook generation tool 115 to create a phonebook and phonebook delta files 175 and 180, illustrated in **Figure 16**. In one embodiment, the phonebook generation tool 115 is a Java application that uses a database to store and manipulate phonebook data. The tool 115 may communicate with the database utilizing the JDBC protocol. The tool 115 furthermore publishes the generated phonebook and phonebook delta files 175 and 180 to the file system on the web server 100 for publication.

[0051] The generated phonebook files 175 may be customized according to the needs of a customer, (e.g., a particular POPs may be filtered or removed, and rules may be established for the pricing of POPs).

[0052] A phonebook management system (not shown) maintains a current "open" phonebook version number and tags changes with this version number. Each run of the phonebook generation tool 115 increases this phonebook version by one. When the phonebook generation tool 115 runs, it closes the current "open" phonebook version number, and opens a new "open" phonebook version. All subsequent changes to the phonebook database are tagged with the new "open" phonebook version number.

[0053] The phonebook generation tool 115 determines changes to the phonebook database since the last run of the tool 115, and generates phonebook and phonebook delta files 175 and 180.

[0054] A more detailed description will now be provided with reference to **Figure 15**.

[0055] In one embodiment, the phonebook generation tool 115 generates delta files that contain cumulative changes to the phonebook database 130 since the last version of the phonebooks was published. In one embodiment if the size of the delta files is greater than 75% of the size of the whole phonebook, the delta files are not generated.

[0056] Referring to **Figure 15**, at 1510 the phonebook generation tool 115 creates the next open version phonebook number and updates the current phonebook version to publishing and creates a new open version phonebook. At 1520, the phonebook generation tool 115 retrieves the complete list of POPs from the server. Upon retrieval of the complete POP list the phonebook generation tool 115 at 1530 retrieves the latest customized phonebook. Application of the default filters to the list of POPs (for example, customer location filters) occurs at 1540. At 1550 the phonebook generation tool 115 applies customer-specified filters to the list of POPs (e.g., eliminates some of the countries that the customer specifically requested to be excluded from the available POPs). At 1560 the phonebook generation tool 115 determines if the pricing plans for particular POPs have changed. If positive, then the necessary corrections are made to the list of POPs. In some instances the customer may specify his/her own pricing rules, for example, to charge end-users 10% more than the price iPass charges the customer. These customer pricing rules are applied at 1570. Upon application of the above-described rules, the phonebook generation tool 115 determines the new, modified and deleted POPs at 1580. At 1590, the new POPs list is printed to a full phonebook tree with the new open version phonebook number, and at box 1595 the delta files 180 are printed into a delta files tree. In one embodiment the phonebook and delta trees are stored at the web server 100.

[0057] All the files are associated with a version number in order to facilitate a more efficient update process described above.

[0058] The phonebook generation tool 115 utilizes "pricing" and "access point" data maintained in the access point and pricing databases 136 and 137 illustrated

in **Figure 1B**. The pricing data includes buy and sell prices for all access points. Sell prices for access points combined with a number of other pricing parameters constitute a "pricing plan". Each phonebook, for which a record is maintained within the phonebook database 130, has a pricing plan associated therewith. Access point information includes all POP related information. When access point information is modified, this data is tagged with the latest "open" version number.

METHODOLOGY: Customization by the end-users of the customers

[0059] In one embodiment of the present invention, the end-user invokes a customized network connection application in the form of a dialer 150 on the client machine 101 of **Figure 16**. **Figure 17** illustrates a main dialog box 1700 of the customized dialer 150, according to one embodiment, that is presented to the end-user upon invocation of the dialer 150. To establish a dial-up connection the end-user may select an access point from the list of all the available access points presented to him/her in box 1710. In order not to display the list of all available access points, most of which will be long distance calls, the end-user may enter his/her location in box 1720. The customized dialer 150, in one embodiment of the present invention, filters the list of access points based on the end-user's location and displays only the closest points of access in box 1710. Upon selection of an access point, the end-user may click on a connection button 1715 in order to instruct the customized dialer 150 to establish a network connection via the selected access point. In one embodiment of the present invention, the access points displayed in box 1710 may be sorted by city name. Alternatively, the end-user may sort the access points list by phone numbers, connection speed, or price by clicking on the corresponding column headings. For example, to sort by price the end-user may click on box 1725.

[0060] The end-user may specify the dialing settings to use by the customized dialer 150 when establishing a remote network connection. **Figure 18** illustrates

an exemplary dial properties dialog box 1800 that is presented to the end-user. Facilities using private branch exchange (PBX) (e.g., a private telephone network users of which share a number of outside lines), usually require an access code to obtain an outside line. Thus, in box 1810, the end-user is prompted to enter an outside line code. Some phone lines are setup with a call waiting feature, which in one embodiment may need to be disabled prior to establishing a data connection. The end-user may enter in box 1815 a phone number to dial in order to disable the call waiting feature. In box 1820 the end-user may enter the country and area code from which the end-user is dialing; this information is used by the customized dialer 150 to determine if an area code, a country code or an access code need to be dialed in order to establish a network connection via the end-user-selected access point. In one embodiment, if check box 1825 is checked by the end-user, the selected number will automatically be dialed as a local number. Calling card information may be entered in box 1830 to be used when dialing the end-user-selected access point number. Each calling card may be defined by a name, access number, PIN and a dialing rule.

[0061] In order for the customized dialer 150 to establish the connection with the Internet, the end-user's information such as username, domain and password should be available. End-user information dialog box 1900 illustrated in **Figure 19** prompts the end-user for such information. In one embodiment the end-user information dialog box 1900 is automatically displayed if the end-user dials an access point without providing all the required end-user information.

[0062] The settings dialog box 2000 illustrated in **Figure 20** allows the end-user, in one embodiment of the present invention, to specify settings used in establishing the remote connection. The end-user may specify in box 2010 the number of redial attempts to be made by the customized dialer 150 when the network connection may not be established from the first dialing attempt. Alternatively, in box 2015 the end-user may specify the duration of an attempt to establish the connection before redialing. For example, the end-user may desire

for the customized dialer 150 to redial the same or different access point number if connection is not established within 90 seconds. Depending on the device used for the dial-up connection, particular features of the customized dialer 150 need to be invoked. Thus, the end-user may specify the dialing-up device that he/she may select from the drop down menu 2020.

[0063] In one embodiment, the end-user may select an option of automatic update of the phonebook upon establishment of the network connection by check box 2025. This will ensure that the latest network access numbers are used next time the end-user invokes the customized dialer 150. A "smart redial" option, when enabled by the end-user check box 2030, directs the customized dialer 150 to dial another number in the same city when the dial-up attempt failed using the first network access number. In one embodiment the end-user may wish to run particular applications upon the establishment of the network connection, for example a Web browser, such as Internet Explorer™ (Microsoft Corporation). Instead of opening desired applications manually, the end-user may direct the customized dialer 150 automatically to launch specified applications when the network connection is established by adding software applications to box 2035 utilizing Add 2040, Modify 2045 and Delete 2050 buttons illustrated in **Figure 20**. In another embodiment of the present invention, the end-user may select an option of launching a default web browser once the connection is established by checking on the Default Web Browser box 2055.

[0064] In one embodiment of the present invention, the end-user may bookmark the access points that are most often used. **Figure 21** illustrates an exemplary dialog box 2100 that the end-user may use in order to compile a list of favorite network access points. Window 2110 allows the end-user to add a bookmark by entering the location of the access point. **Figure 22** illustrates an exemplary dialog box 2200 that an end-user may utilize to modify a list of favorite network access points. Window 2220 allows the end-user to modify the list of bookmarks

by providing a Modify option 2225 to change the properties of a bookmark and a Delete option 2230 to remove a bookmark from the list.

[0065] In one embodiment of the present invention, the end-user may access an online help feature from any dialog boxes described above by clicking on a Help button.

[0066] Some settings may be saved in the configuration files on the client machine 101 when the end-user exits the customized dialer 150. The saved settings may be location filters (country, state, city, area code), connection type (modem, ISDN), selected access points, dial properties including dialing prefixes, the location of the end-user and calling card information, end-user information including end-user name, domain name and password and modem settings including redial attempts, redial timeout, modem device, update phonebook selected options, SmartRedial, bookmarks and programs to launch after the connection is established.

[0067] Certain area codes in the Unites States require 10/11-digit dialing when placing calls within the area code. These dialing requirements are very regional and are constantly changing. In one embodiment of the present invention, a dialing rule file is downloaded to the client machine 101 along with the distribution of the customized dialer 150, containing all the area codes that require 10/11 digit dialing.

[0068] **Figure 23** is a diagrammatic representation of three exemplary protocols and hardware components of three exemplary access methods, supported by network connection applications according to respective exemplary embodiments of the present invention. Specifically, a modem dialup access method is illustrated at 2300, a wireless broadband access method is illustrated at 2302 and a wired broadband access method is illustrated at 2304. As mentioned above, the present invention is not restricted to the generation, updating and distribution of a dialer for establishing a modem dialup connection, and extends to a method and system for generating, updating

and/or distributing a network connection application for establishing a network connection between the two machines.

Build server

[0069] As mentioned above, the customized connection application in the exemplary form of the dialer 150 may be in the form of a self-extracting executable. In one embodiment of the invention, the self-extracting executable that is generated by the customization tool 120 is signed using Microsoft's Authenticode technology.

[0070] Authenticode verifies who published the file and provides an indication of whether or not the file has been tampered with since the publisher signed it by incorporating a digital signature directly into an executable file. Installers are often downloaded using Internet Explorer, which can validate signatures generated with Authenticode.

[0071] To sign a file with Authenticode, an encryption module including a SignCode utility is installed on the build server 110 (see Figure 1A), and is executed after each build of a connection application executable. The SignCode utility is available from Microsoft as part of their Authenticode SDK. A certificate signed by a well-known authority, such as Verisign, may be obtained to be used for Authenticode signing. Accordingly, the web server 100, database server 105, update server 113 and build server may be provided behind a firewall. The secure dialer 150 may be replicated to servers accessible to the public across the globe.

[0072] Referring in particular to Figure 25, reference numeral 2500 generally indicates a method, in accordance with one embodiment of the invention, of securing a self-extracting executable for installing a connection application in the exemplary form of the dialer 150. As shown at block 2502, the customization tool 120 builds a customized dialer 150 that is arranged in the form of a self-extracting installer file. The installer file includes all the components of the

customized dialer 150 so that, upon execution of the installer file, a complete dialer 150 is installed on the client machine 101. Once the customized dialer 150 and its self-extracting installer has been built, at block 2504, it is cryptographically signed with SSH credentials obtained from a trusted server (see block 2506). Thereafter, the secure self-extracting installer is distributed to customers, for example, service providers for downloading via the network as discussed above.

[0073] Referring to Figure 26, reference numeral 2510 generally indicates a method for verifying a self-extracting connection application on the client machine 101. A user desirous of installing the dialer 150 on his or her client machine 101 may download the self-extracting executable from any one of the service providers, as shown at block 2512, via the Internet. When using a browser such as Internet Explorer to download a file signed with Authenticode technology, the browser may automatically generate a pop-up window asking the user if the file is to be saved or opened. If the user selects to save the file, Authenticode verification may not take place. However, if the user chooses to open the file (see block 2514) Authenticode then identifies the publisher of the software and verifies that the file has not been tampered with. As shown at block 2516, the user may thus check to ensure that the connection application is from a trusted source and identify the publisher of the software. If the dialer 150 is not from a trusted source, then the user may abort the installation process as shown at block 2518.

[0074] If the dialer 150 is from a trusted source, then the user may verify that the dialer 150 was not tampered with since it was published by the trusted source as shown at block 2520. If the dialer 150 has been tampered with, the user may abort the installation process as shown at block 2518. If, however, the dialer 150 has not been tampered with, then the installation of the dialer 150 may then be completed as shown at block 2522.

[0075] Thus, if the security of any server from which the dialer 150 is downloaded becomes compromised, a user downloading the dialer 150 may be made aware of such a security breach. If the web server 100 with the updated files is compromised, the dialer 150 may then abort installation as the updated files may, for example, contain viruses.

[0076] The dialer 150, when installed on the client computer 101, can be used by the end-users to connect to the Internet from anywhere in the world. After connecting to the Internet, the dialer 150 may check for configuration, data and software updates and apply these updates to the installed client if appropriate updates are found.

[0077] A user attempting to install a connection application where the Authenticode signature is not recognized by Internet Explorer may receive one of the following messages:

Internet Explorer 3.x:

A Windows application is attempting to open or install the following software component:

<filename.exe> from

<example.microsoft.com>

Please be aware that some files may contain viruses or otherwise harm your computer. This component has not been digitally "signed" by its publisher. Do you wish to continue?

Internet Explorer 4.x or 5:

Security Warning

Warning: The authenticity of this software can not be verified, therefore this software cannot be trusted.

Problem/s listed below: <description of problem>

Are you sure that you want to install and run <program> distributed by <Publisher>?

[0078] In the foregoing specification the present invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made to the specific exemplary embodiments without departing from the broader spirit and scope of the invention as set forth in the appended claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

Computer System

[0079] **Figure 24** is a diagrammatic representation of a machine in the form of computer system 2400 within which software, in the form of a series of machine-readable instructions, for performing any one of the methods discussed above may be executed. The computer system 2400 includes a processor 2402, a main memory 2404 and a static memory 2406, which communicate via a bus 2408. The computer system 2400 is further shown to include a video display unit 2410 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)). The computer system 2400 also includes an alphanumeric input device 2412 (e.g., a keyboard), a cursor control device 2414 (e.g., a mouse), a disk drive unit 2416, a signal generation device 2418 (e.g., a speaker) and a network interface device 2420. The disk drive unit 2416 accommodates a machine-readable medium 2422 on which software 2424 embodying any one of the methods described above is stored. The software 2424 is shown to also reside, completely or at least partially, within the main memory 2404 and/or within the processor 2402. The software 2424 may furthermore be transmitted or received by the network interface device

2420. For the purposes of the present specification, the term "machine-readable medium" shall be taken to include any medium that is capable of storing or encoding a sequence of instructions for execution by a machine, such as the computer system 2400, and that causes the machine to perform the methods of the present invention. The term "machine-readable medium" shall be taken to include, but not be limited to, solid-state memories, optical and magnetic disks, and carrier wave signals.

[0080] If written in a programming language conforming to a recognized standard, the software 2424 can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a machine, such as the computer system 2400, the machine to perform an action or a produce a result.

[0081] The preceding description of **Figure 24** is intended to provide an overview of computer hardware and other operating components suitable for implementing the invention, but is not intended to limit the applicable environments. One of skill in the art will immediately appreciate that the invention can be practiced with computer architectures and configurations other than that shown in **Figure 24**, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor. The invention can also be practiced in distributed

computing environments where tasks are performed by remote processing devices that are linked through a communications network.

CLAIMS

What is claimed is:

1. A system to secure a connection application in a multi-party access environment including a plurality of service providers, the system including:

a server adapted to generate a customized connection application; and

an encryption module adapted to cryptographically sign the customized connection application, the customized connection application being for installation on a client machine.

2. The system of claim 1, in which the customized connection application is in the form of a self-extracting executable to automatically install the connection application on the client machine.

3. The system of claim 1, in which cryptographically signing the customized connection application includes signing the file using Authenticode technology.

4. The system of claim 1, which distributes the connection application to the plurality of service providers via a computer network for downloading by the client machine

5. A system to secure a connection application in a multi-party access environment including a plurality of service providers, the system including:

means to generate a customized connection application; and

means to cryptographically sign the customized connection application, the customized connection application being for installation on a client machine.

6. A method to secure a connection application in a multi-party access environment including a plurality of service providers, the method including:

generating a customized connection application; and

cryptographically signing the customized connection application, the customized connection application being for installation on a client machine.

7. The method of claim 6, in which the customized connection application is in the form of a self-extracting executable to automatically install the connection application on the client machine.

8. The method of claim 6, in which cryptographically signing the customized connection application includes signing the file using Authenticode technology.

9. The method of claim 6, which includes distributing the connection application to the plurality of service providers via a computer network for downloading by the client machine.

10. A machine-readable medium, embodying a sequence of instructions that, when executed by a machine cause the machine to execute a method to secure a connection application in a multi-party access environment including a plurality of service providers, the method including:

generating a customized connection application; and

cryptographically signing the customized connection application, the customized connection application being for installation on a client machine.

11. The machine-readable medium of claim 10, in which the customized connection application is in the form of a self-extracting executable to automatically install the connection application on the client machine.

12. The machine-readable medium of claim 10, in which cryptographically signing the customized connection application includes signing the file using Authenticode technology.

13. The machine-readable medium of claim 10, which includes distributing the connection application to the plurality of service providers via a computer network for downloading by the client machine

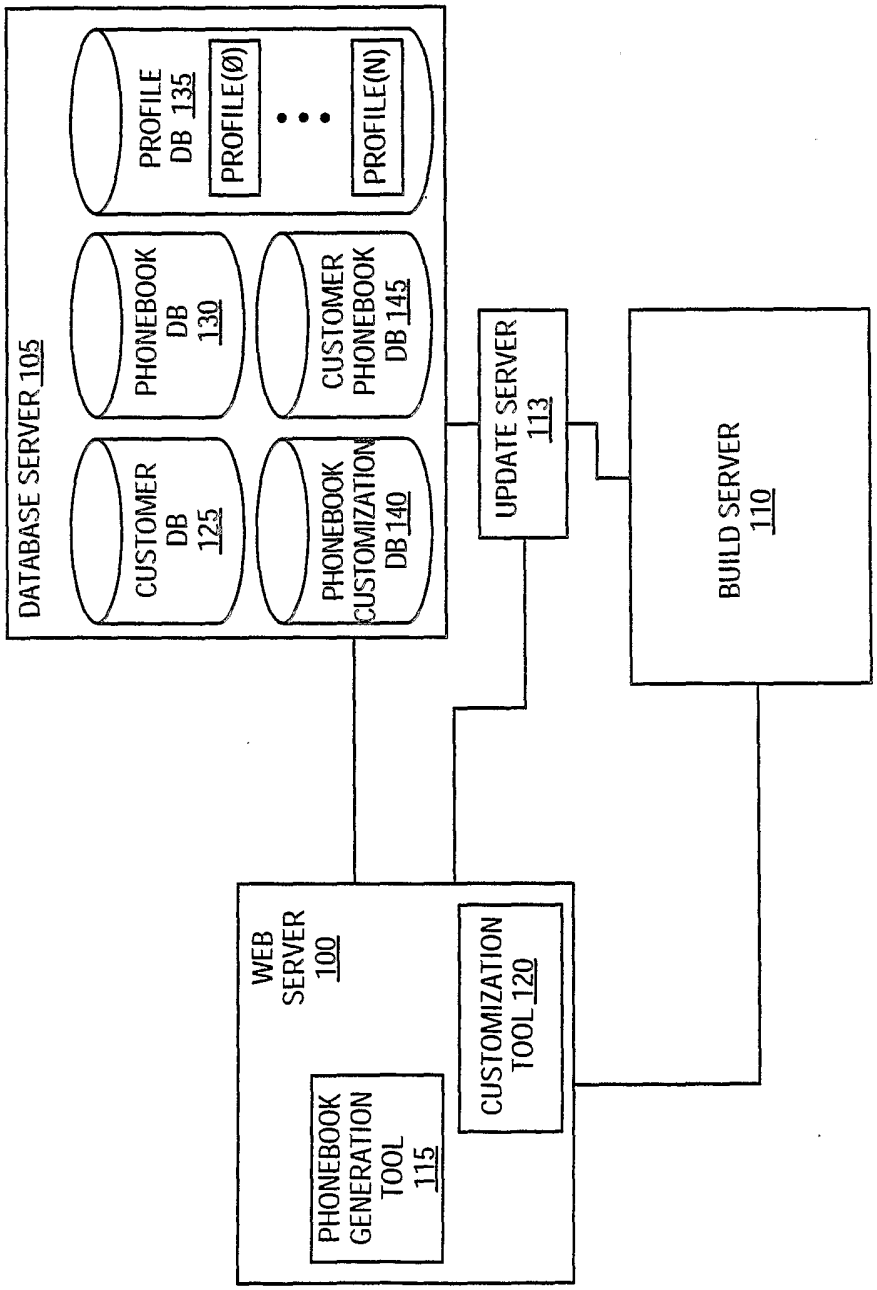

DATABASE SERVER 105
CUSTOMER DB 125
PHONEBOOK CUSTOMIZATION DB 140
PHONEBOOK DB 130
CUSTOMER PHONEBOOK DB 145
PROFILE DB 135
PROFILE(Ø)
PROFILE(N)
UPDATE SERVER 113
BUILD SERVER 110
WEB SERVER 100
PHONEBOOK GENERATION TOOL 115
CUSTOMIZATION TOOL 120
10

FIG. 1A

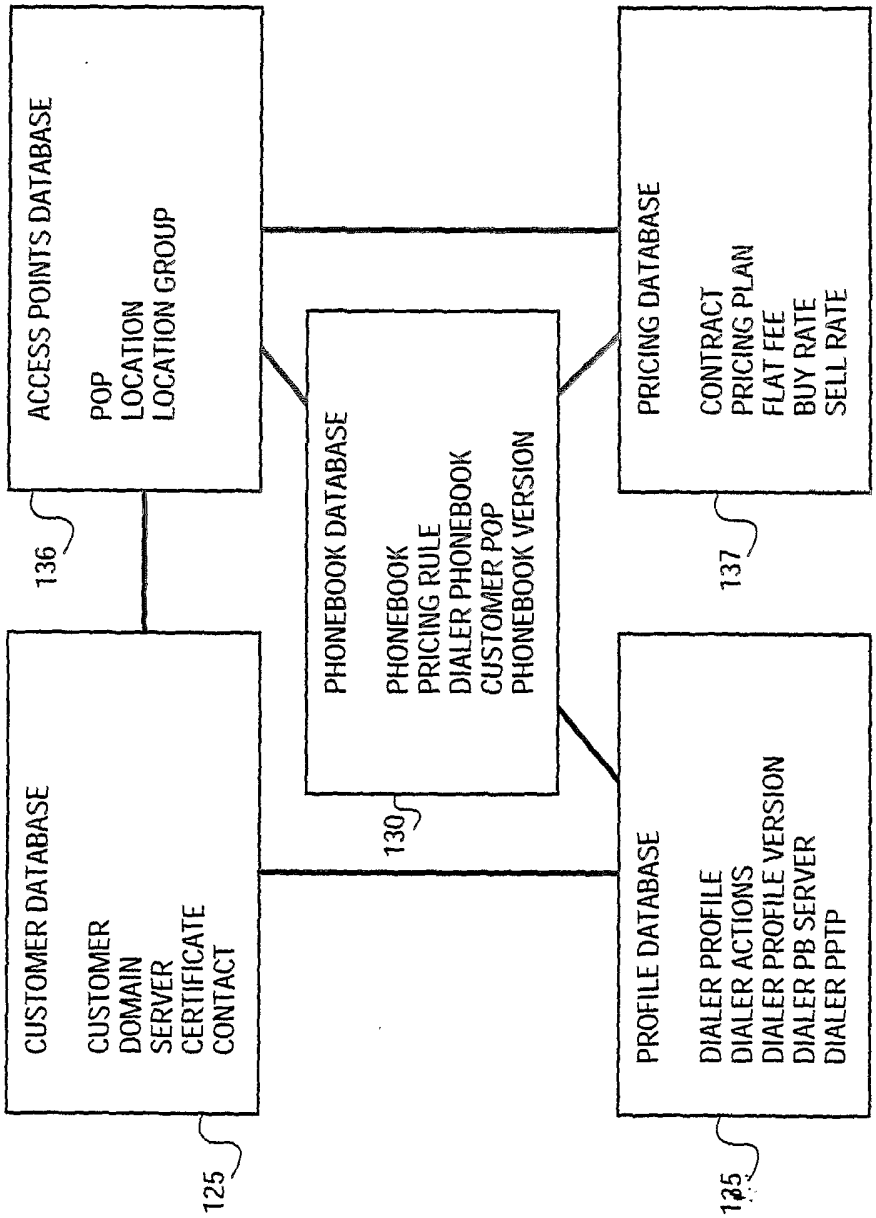

ACCESS POINTS DATABASE
POP
LOCATION
LOCATION GROUP
136
CUSTOMER DATABASE
CUSTOMER
DOMAIN
SERVER
CERTIFICATE
CONTACT
125
PHONEBOOK DATABASE
PHONEBOOK
PRICING RULE
DIALER PHONEBOOK
CUSTOMER POP
PHONEBOOK VERSION
130
PRICING DATABASE
CONTRACT
PRICING PLAN
FLAT FEE
BUY RATE
SELL RATE
137
PROFILE DATABASE
DIALER PROFILE
DIALER ACTIONS
DIALER PROFILE VERSION
DIALER PB SERVER
DIALER PPTP
135

FIG. 1B

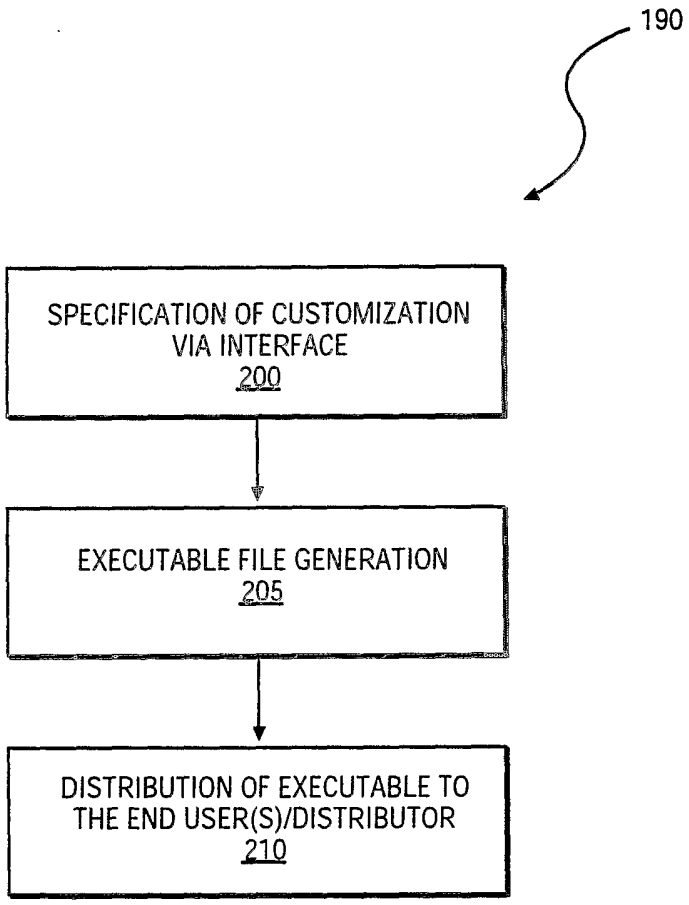

190
SPECIFICATION OF CUSTOMIZATION VIA INTERFACE
200
EXECUTABLE FILE GENERATION
205
DISTRIBUTION OF EXECUTABLE TO THE END USER(S)/DISTRIBUTOR
210

FIG. 2

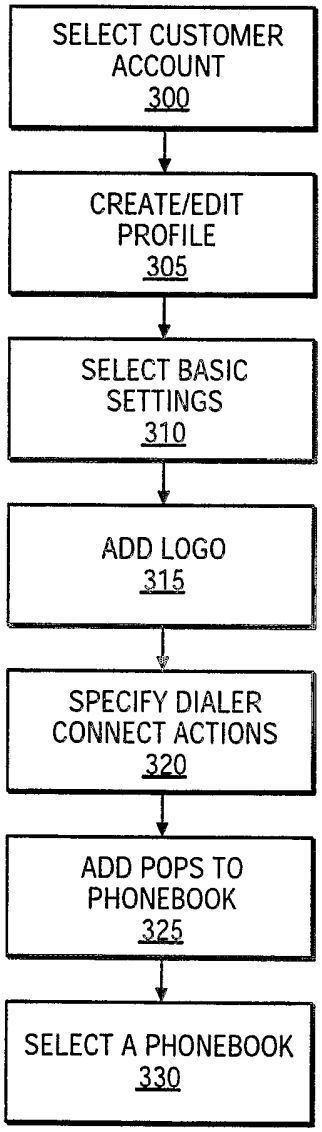

SELECT CUSTOMER ACCOUNT
300
CREATE/EDIT PROFILE
305
SELECT BASIC SETTINGS
310
ADD LOGO
315
SPECIFY DIALER CONNECT ACTIONS
320
ADD POPS TO PHONEBOOK
325
SELECT A PHONEBOOK
330

FIG. 3A

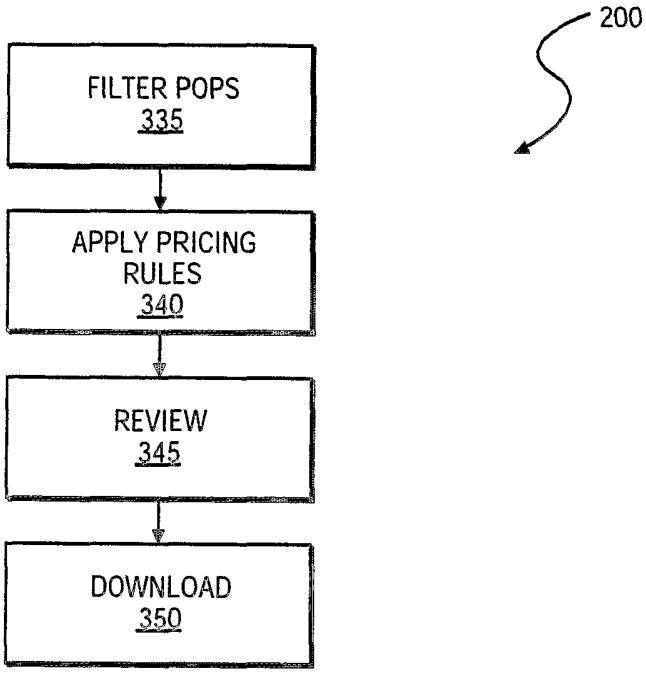

200
FILTER POPS
335
APPLY PRICING
RULES
340
REVIEW
345
DOWNLOAD
350

FIG. 3B

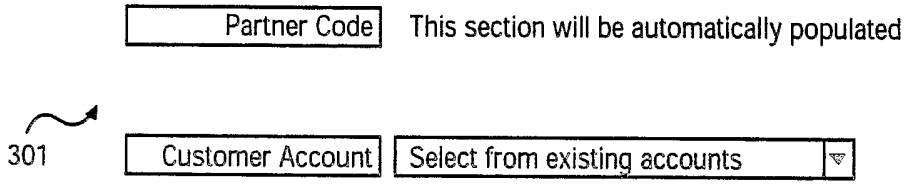

Partner Code
This section will be automatically populated
301
Customer Account
Select from existing accounts

FIG. 4

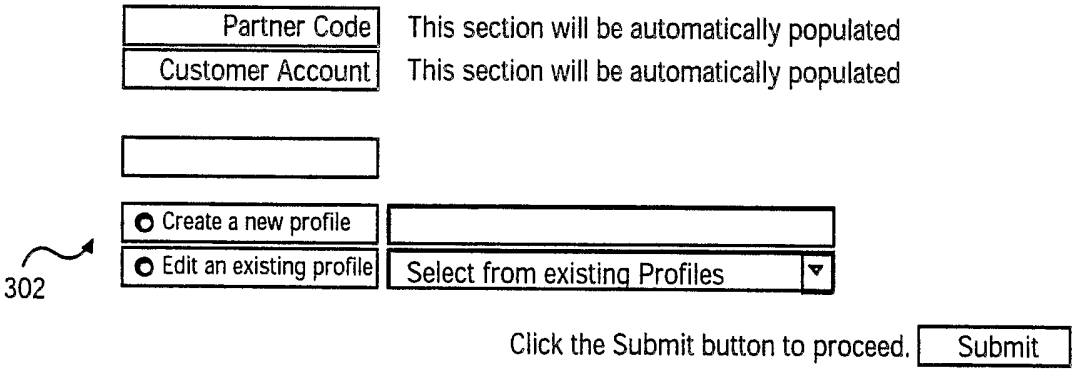

Partner Code
This section will be automatically populated
Customer Account
This section will be automatically populated
Create a new profile
Edit an existing profile
Select from existing Profiles
302
Click the Submit button to proceed.
Submit

FIG. 5

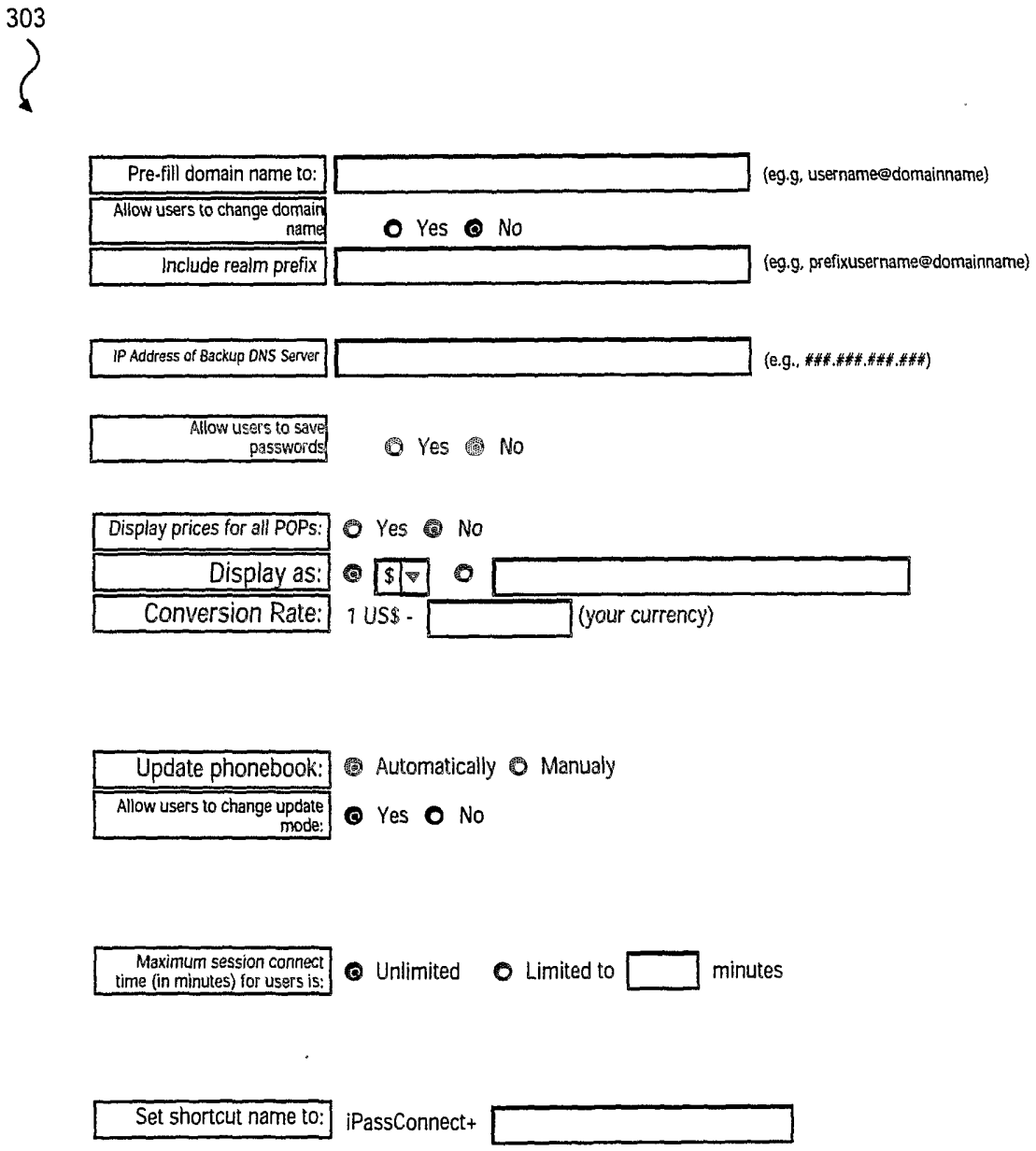

303
Pre-fill domain name to:
(eg.g, username@domainname)
Allow users to change domain name
Yes
No
Include realm prefix
(eg.g, prefixusername@domainname)
IP Address of Backup DNS Server
(e.g., ###.###.###.###)
Allow users to save passwords
Yes
No
Display prices for all POPs:
Yes
No
Display as:
$
Conversion Rate:
1 US$ -
(your currency)
Update phonebook:
Automatically
Manualy
Allow users to change update mode:
Yes
No
Maximum session connect time (in minutes) for users is:
Unlimited
Limited to
minutes
Set shortcut name to:
iPassConnect+

FIG. 6

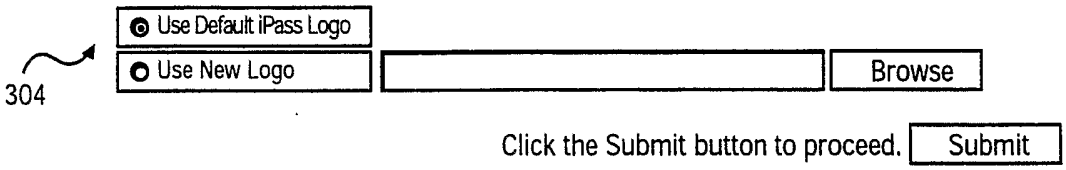


FIG. 7A

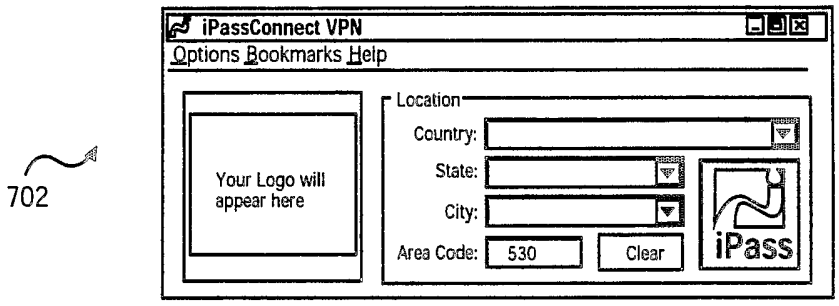


FIG. 7B

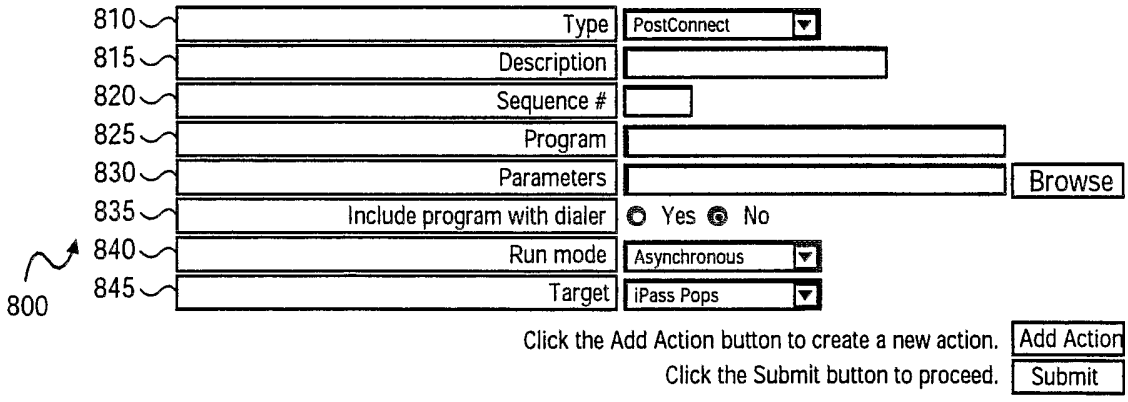


FIG. 8

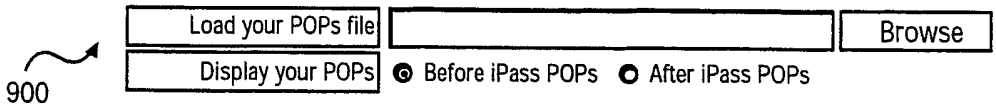


FIG. 9

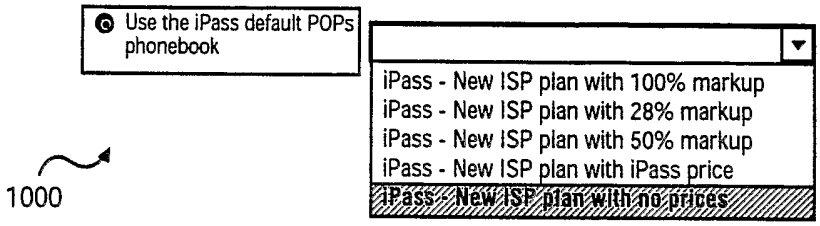


FIG. 10

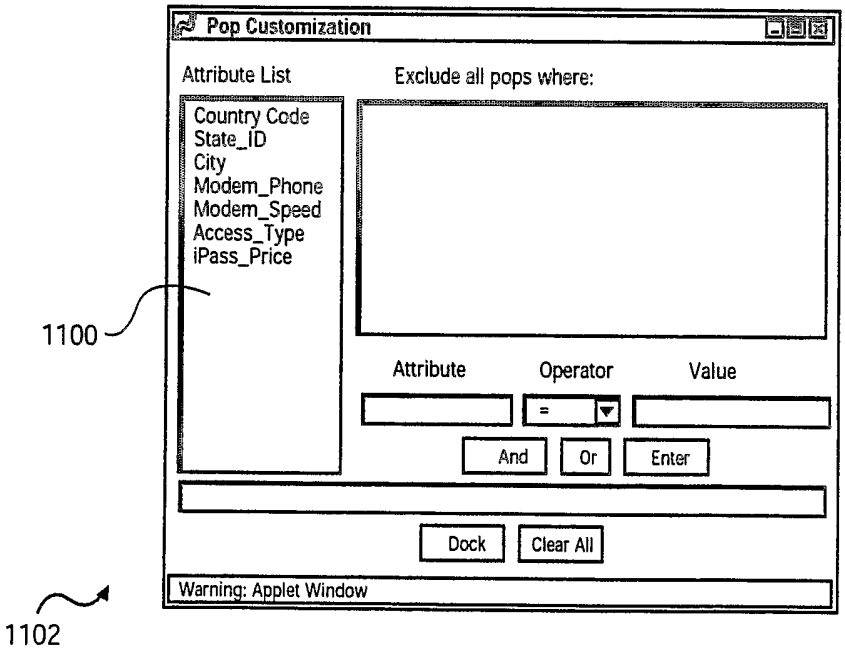
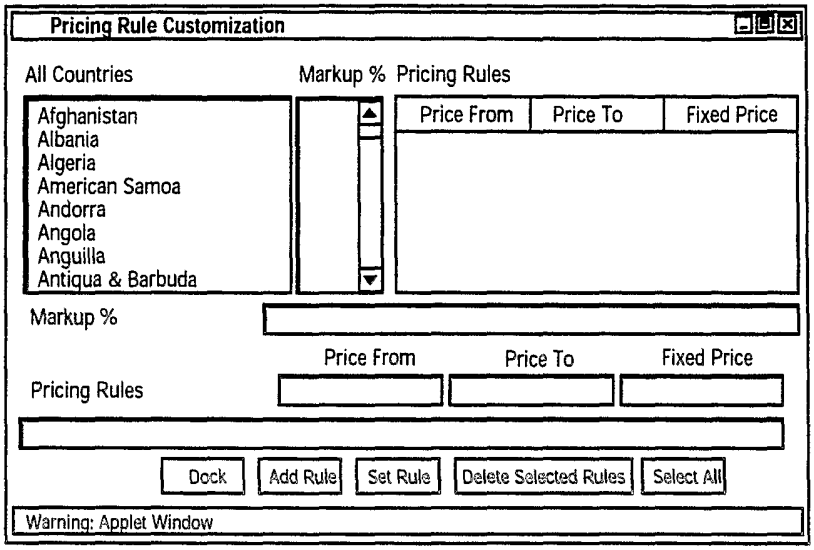


FIG. 11



1200

FIG. 12

Default Authentication Domain	
Backup DNS IP Address	254.254.254.254
Save Password	No
POP Pricing Display	Shown, Currency: $
Phonebook Updates	Default: automatic, User can change: yes
Session Limit	Unlimited
Dialer Shortcut Name	iPassConnect COMPANY
Custom Logo	Default Logo
Connect Actions	PreConnect (2),PreDial (1),PreTunnel (1),OnCancel (1),OnError (1),
Custom POPs added	Yes
iPass POPs Phonebook	ISP plan with no prices

Click the Edit button to make changes. Edit
Click the Build Dialer button to proceed. Build Dialer

1300

FIG. 13

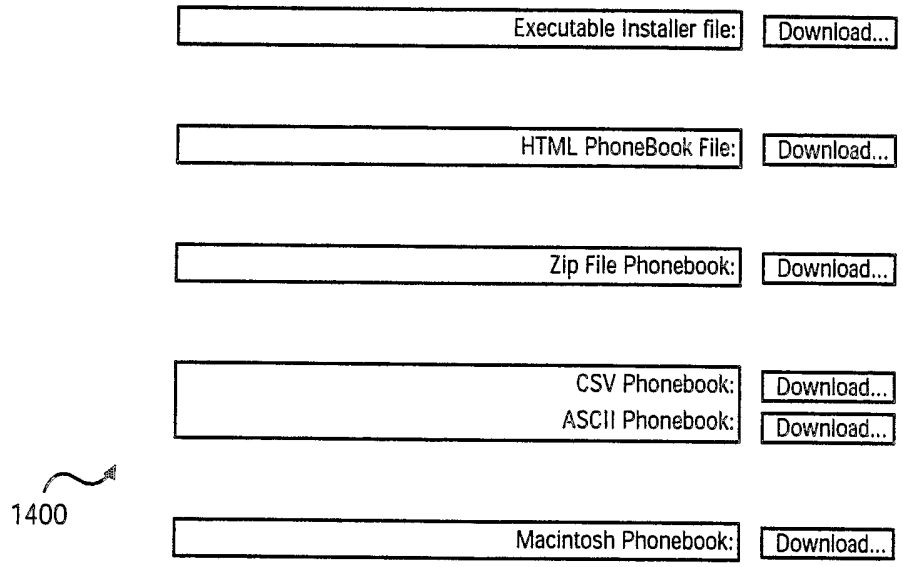

Executable Installer file:
Download...
HTML PhoneBook File:
Download...
Zip File Phonebook:
Download...
CSV Phonebook:
Download...
ASCII Phonebook:
Download...
1400
Macintosh Phonebook:
Download...

FIG. 14

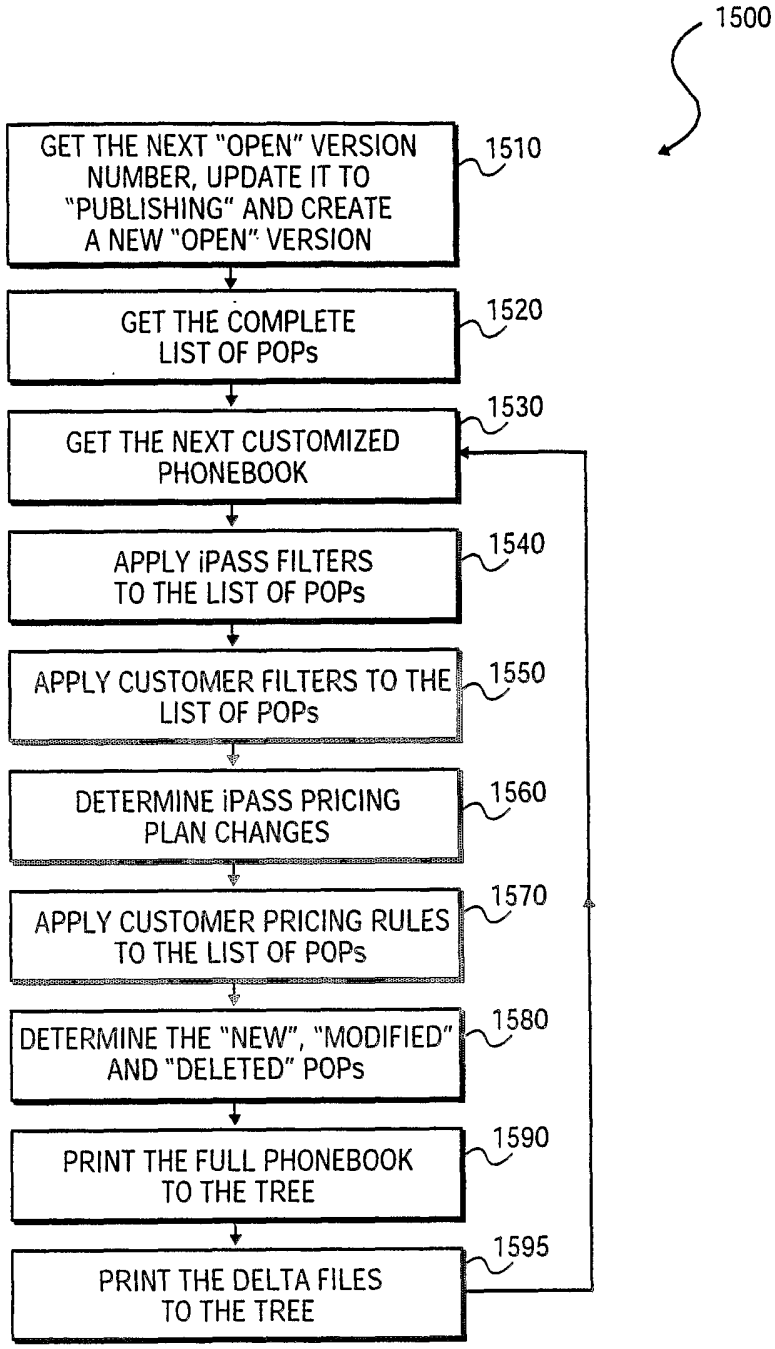

1500
GET THE NEXT "OPEN" VERSION NUMBER, UPDATE IT TO "PUBLISHING" AND CREATE A NEW "OPEN" VERSION
1510
GET THE COMPLETE LIST OF POPs
1520
GET THE NEXT CUSTOMIZED PHONEBOOK
1530
APPLY iPASS FILTERS TO THE LIST OF POPs
1540
APPLY CUSTOMER FILTERS TO THE LIST OF POPs
1550
DETERMINE iPASS PRICING PLAN CHANGES
1560
APPLY CUSTOMER PRICING RULES TO THE LIST OF POPs
1570
DETERMINE THE "NEW", "MODIFIED" AND "DELETED" POPs
1580
PRINT THE FULL PHONEBOOK TO THE TREE
1590
PRINT THE DELTA FILES TO THE TREE
1595

FIG. 15

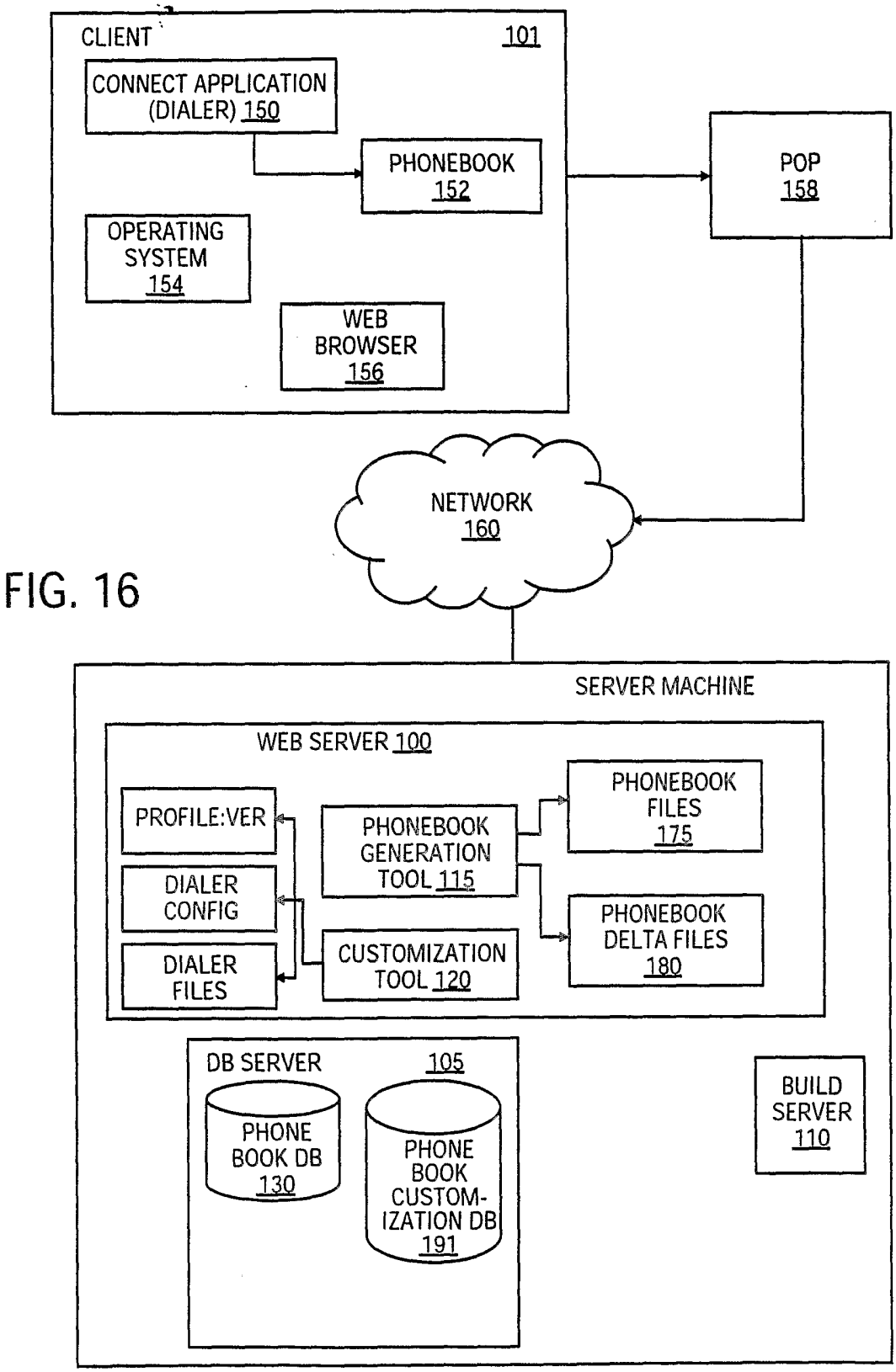

CLIENT
101
CONNECT APPLICATION (DIALER) 150
PHONEBOOK 152
OPERATING SYSTEM 154
WEB BROWSER 156
POP 158
NETWORK 160
SERVER MACHINE
WEB SERVER 100
PROFILE:VER
DIALER CONFIG
DIALER FILES
PHONEBOOK GENERATION TOOL 115
CUSTOMIZATION TOOL 120
PHONEBOOK FILES 175
PHONEBOOK DELTA FILES 180
DB SERVER
105
PHONE BOOK DB 130
PHONE BOOK CUSTOM-IZATION DB 191
BUILD SERVER 110

FIG. 16

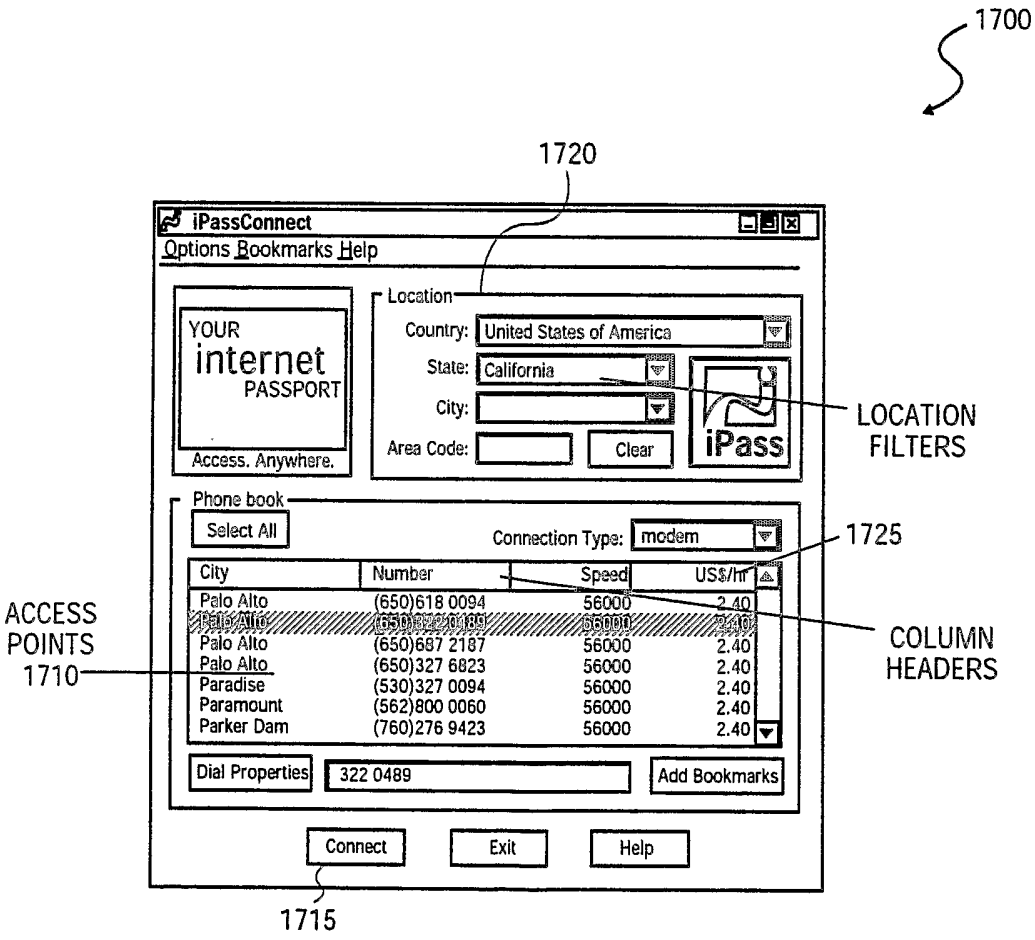

1700
1720
iPassConnect
Options Bookmarks Help
YOUR
internet
PASSPORT
Access. Anywhere.
Location
Country: United States of America
State: California
City:
Area Code:
Clear
iPass
LOCATION FILTERS
Phone book
Select All
Connection Type: modem
1725
City
Number
Speed
US$/hr
Palo Alto (650)618 0094 56000 2.40
Palo Alto (650)687 2187 56000 2.40
Palo Alto (650)327 6823 56000 2.40
Paradise (530)327 0094 56000 2.40
Paramount (562)800 0060 56000 2.40
Parker Dam (760)276 9423 56000 2.40
ACCESS POINTS 1710
COLUMN HEADERS
Dial Properties
322 0489
Add Bookmarks
Connect
Exit
Help
1715

FIG. 17

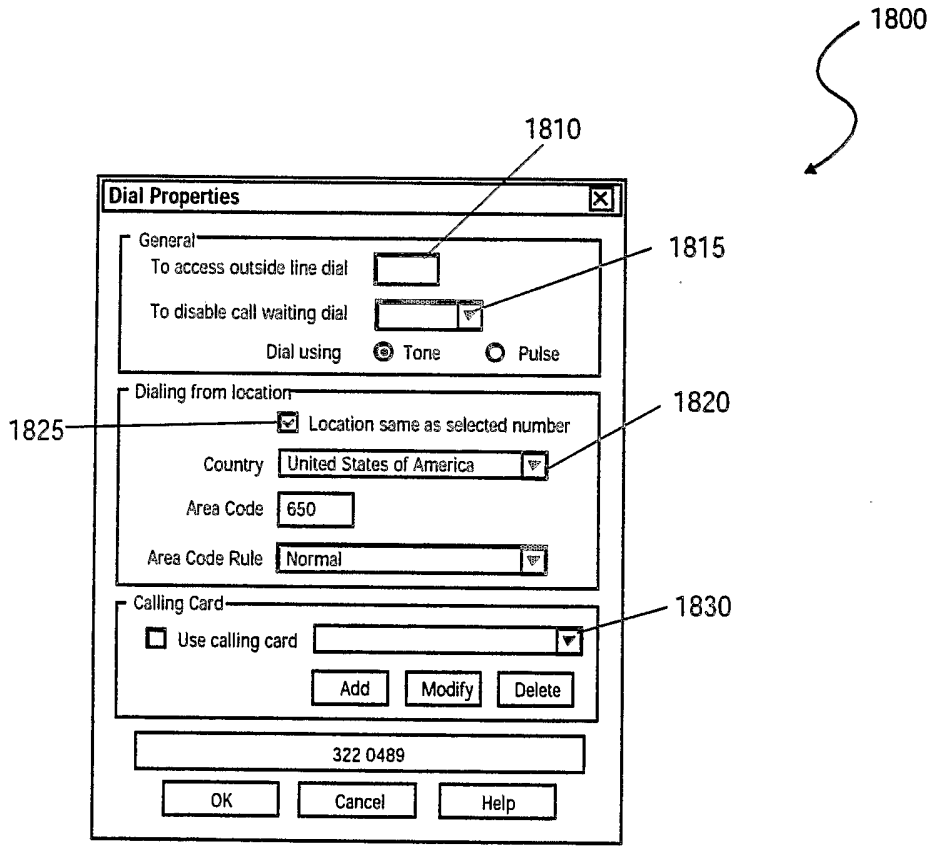

1800
1810
Dial Properties
General
To access outside line dial
1815
To disable call waiting dial
Dial using
Tone
Pulse
Dialing from location
1825
Location same as selected number
1820
Country
United States of America
Area Code
650
Area Code Rule
Normal
Calling Card
1830
Use calling card
Add
Modify
Delete
322 0489
OK
Cancel
Help

FIG. 18

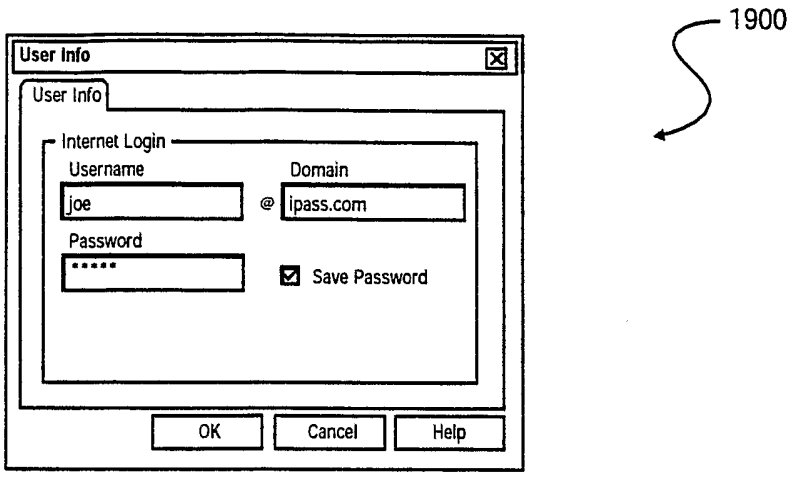

1900
User Info
User Info
Internet Login
Username
Domain
joe
@
ipass.com
Password
Save Password
OK
Cancel
Help

FIG. 19

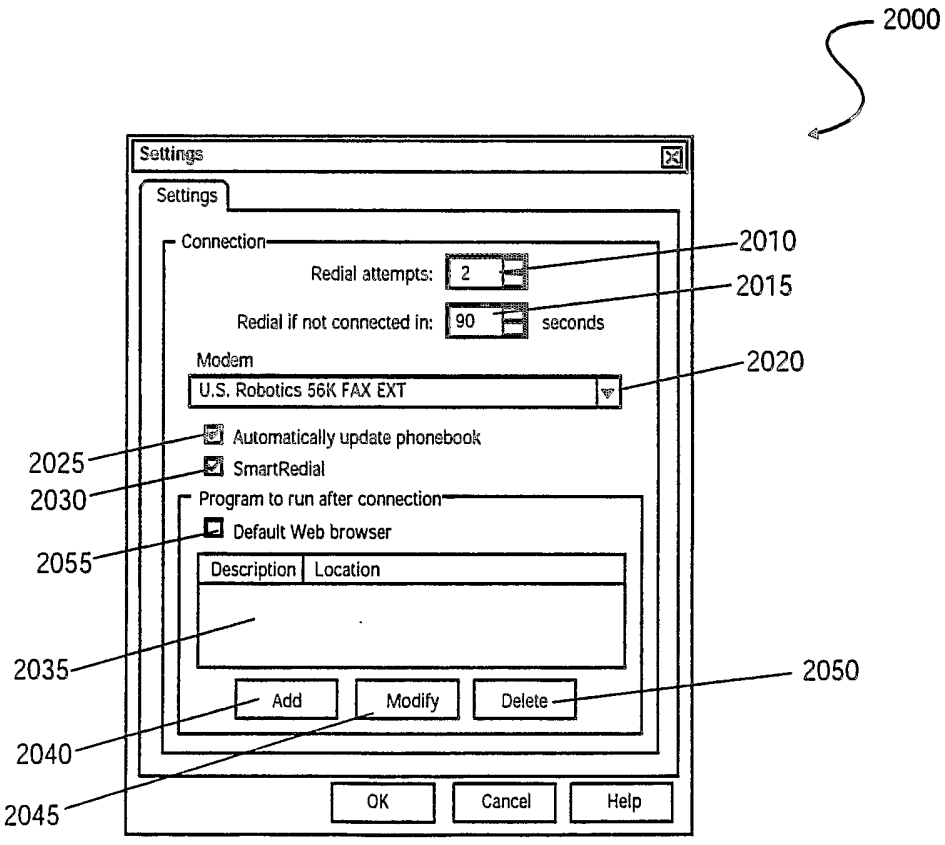

2000
Settings
Settings
Connection
Redial attempts:
2
2010
Redial if not connected in:
90
seconds
2015
Modem
U.S. Robotics 56K FAX EXT
2020
Automatically update phonebook
SmartRedial
Program to run after connection
Default Web browser
Description
Location
2025
2030
2055
2035
2040
2045
Add
Modify
Delete
2050
OK
Cancel
Help

FIG. 20

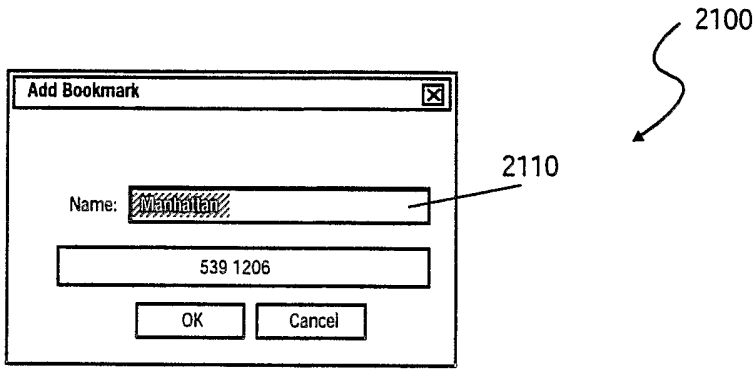

2100
Add Bookmark
2110
Name:
Manhattan
539 1206
OK
Cancel

FIG. 21

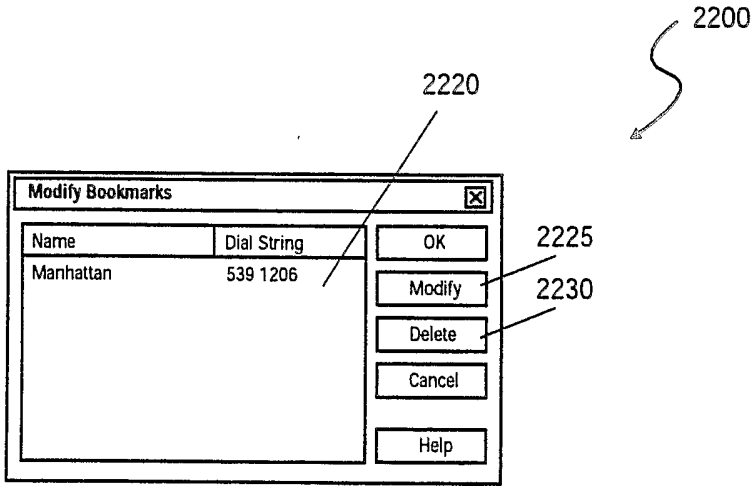

2200
2220
Modify Bookmarks
Name
Dial String
Manhattan
539 1206
OK
2225
Modify
2230
Delete
Cancel
Help

FIG. 22

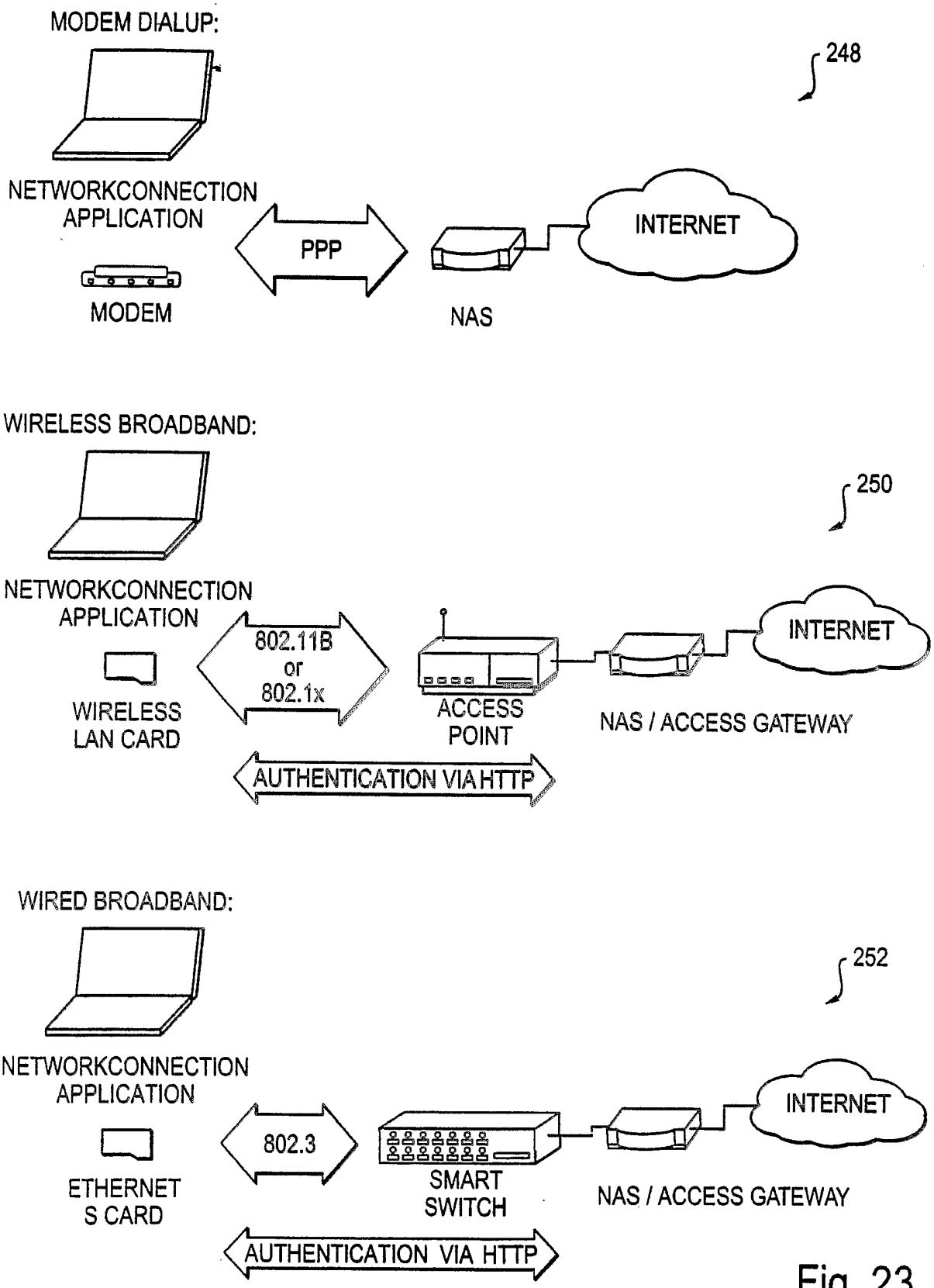

MODEM DIALUP:
248
NETWORKCONNECTION APPLICATION
PPP
INTERNET
MODEM
NAS
WIRELESS BROADBAND:
250
NETWORKCONNECTION APPLICATION
802.11B or 802.1x
INTERNET
WIRELESS LAN CARD
ACCESS POINT
NAS / ACCESS GATEWAY
AUTHENTICATION VIA HTTP
WIRED BROADBAND:
252
NETWORKCONNECTION APPLICATION
802.3
INTERNET
ETHERNET S CARD
SMART SWITCH
NAS / ACCESS GATEWAY
AUTHENTICATION VIA HTTP

Fig. 23

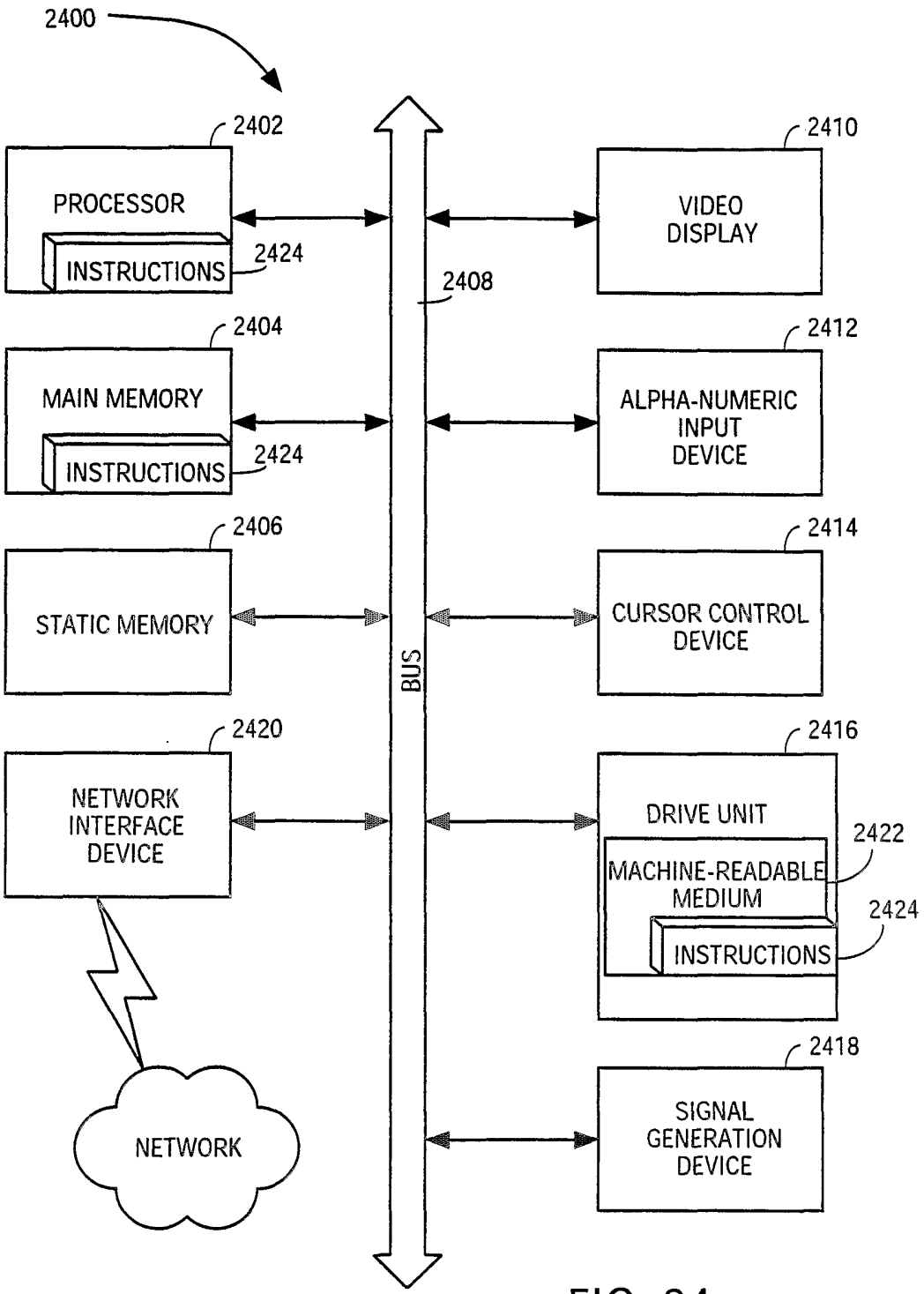

2400
2402
PROCESSOR
INSTRUCTIONS
2424
2404
MAIN MEMORY
INSTRUCTIONS
2424
2406
STATIC MEMORY
2420
NETWORK INTERFACE DEVICE
NETWORK
BUS
2408
2410
VIDEO DISPLAY
2412
ALPHA-NUMERIC INPUT DEVICE
2414
CURSOR CONTROL DEVICE
2416
DRIVE UNIT
MACHINE-READABLE MEDIUM
2422
INSTRUCTIONS
2424
2418
SIGNAL GENERATION DEVICE

FIG. 24

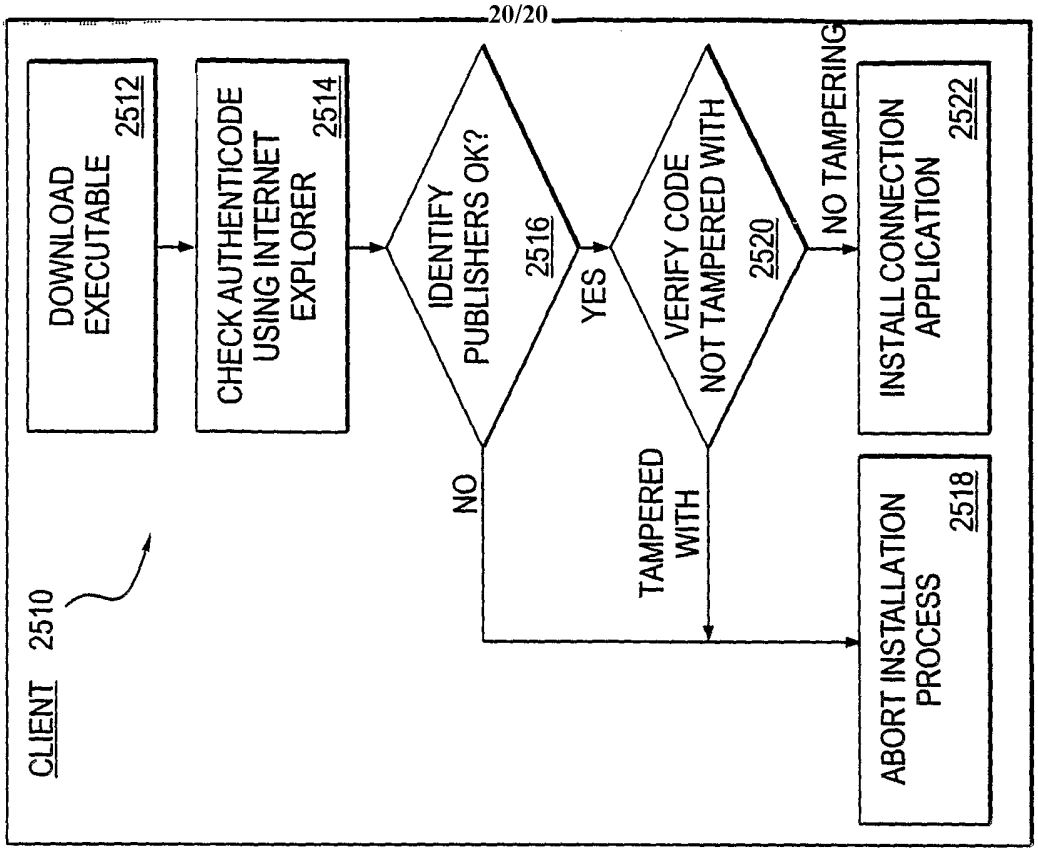

CLIENT 2510
DOWNLOAD EXECUTABLE 2512
CHECK AUTHENTICODE USING INTERNET EXPLORER 2514
IDENTIFY PUBLISHERS OK? 2516
NO
YES
VERIFY CODE NOT TAMPERED WITH 2520
TAMPERED WITH
NO TAMPERING
INSTALL CONNECTION APPLICATION 2522
ABORT INSTALLATION PROCESS 2518

Fig. 26

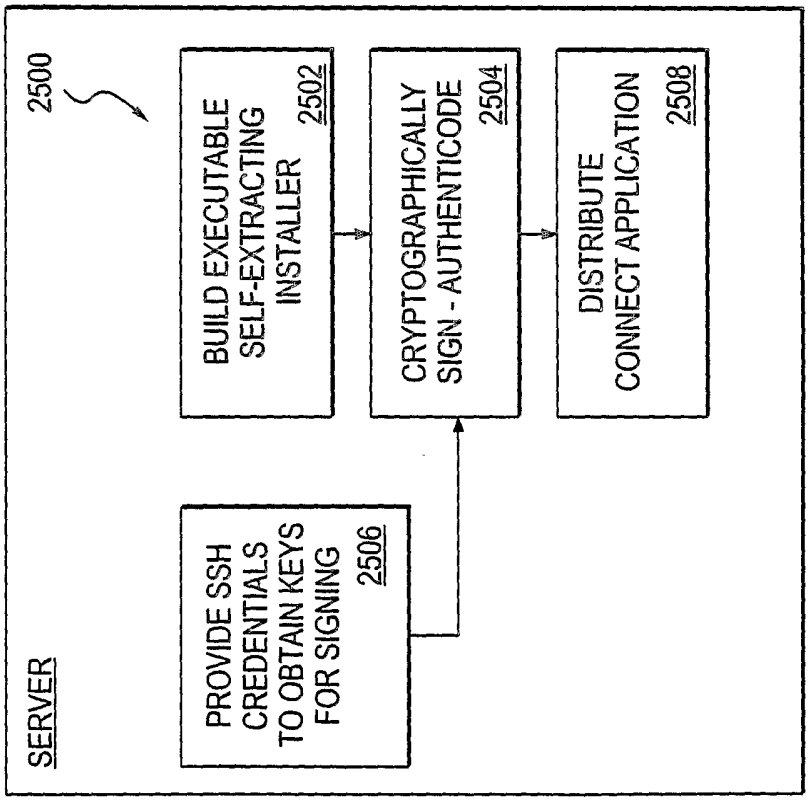

SERVER
2500
BUILD EXECUTABLE SELF-EXTRACTING INSTALLER 2502
CRYPTOGRAPHICALLY SIGN - AUTHENTICODE 2504
DISTRIBUTE CONNECT APPLICATION 2508
PROVIDE SSH CREDENTIALS TO OBTAIN KEYS FOR SIGNING 2506

Fig. 25