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

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2501566 C 2012/07/10

(11)(21) 2 501 566

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2003/10/30
(87) Date publication PCT/PCT Publication Date: 2004/05/13
(45) Date de délivrance/Issue Date: 2012/07/10
(85) Entrée phase nationale/National Entry: 2005/04/06
(86) N° demande PCT/PCT Application No.: IB 2003/004843
(87) N° publication PCT/PCT Publication No.: 2004/040918
(30) Priorités/Priorities: 2002/10/31 (US10/284,135);
2003/09/16 (US10/662,407)

(51) Cl.Int./Int.Cl. *H04W 84/18* (2009.01)

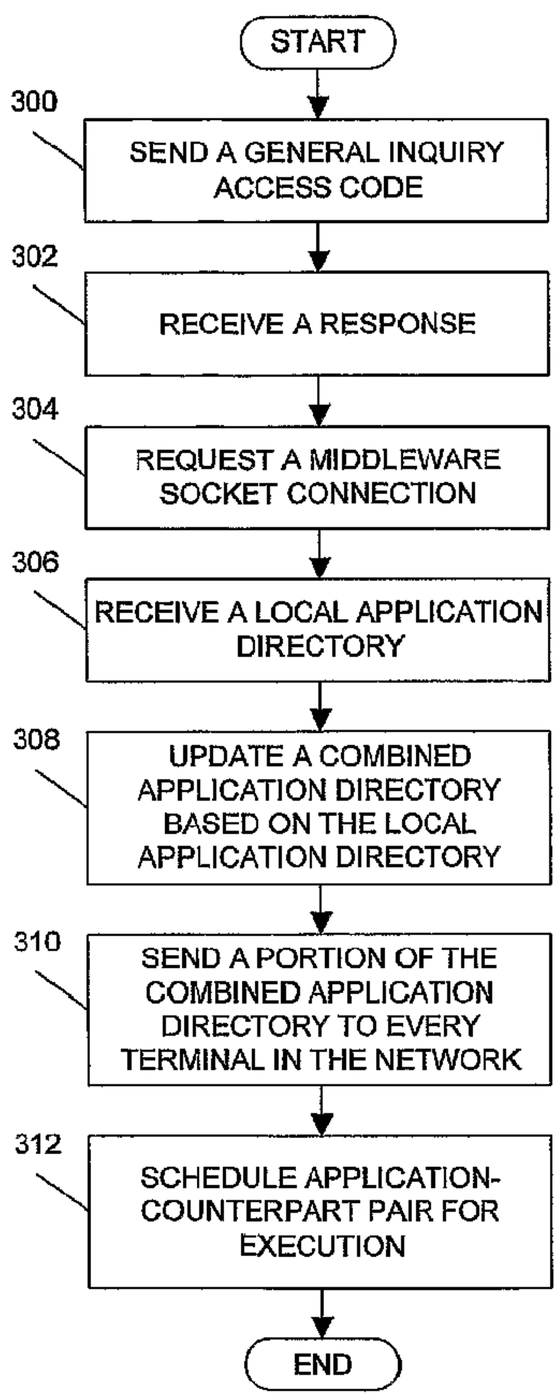
(72) Inventeurs/Inventors:
EKBERG, JAN-ERIK, FI;
LAHTINEN, PEKKA, FI;
LIPASTI, JAAKKO, FI

(73) Propriétaire/Owner:
NOKIA CORPORATION, FI

(74) Agent: SIM & MCBURNEY

(54) Titre : SYSTEME DE DECOUVERTE DE SERVICE ET DE DETECTION DE DISPOSITIF ET METHODE POUR UN RESEAU DE TELECOMMUNICATIONS AD HOC MOBILE

(54) Title: DEVICE DETECTION AND SERVICE DISCOVERY SYSTEM AND METHOD FOR A MOBILE AD HOC COMMUNICATIONS NETWORK




(57) Abrégé/Abstract:
A computer system, method, and computer program product for performing device detection and service discovery in a mobile ad hoc communications network. The method comprises conducting an inquiry of the mobile ad hoc communications network to

CA 2501566 C 2012/07/10

(11)(21) 2 501 566

(13) C

(57) Abrégé(suite)/Abstract(continued):
discover nearby devices. If the inquiry indicates that the nearby devices may include a middleware layer, the method further comprises creating a connection to each of the nearby devices and confirming whether each of the nearby devices include the middleware layer. For each of the nearby devices that include the middleware layer, the method further comprises executing the middleware layer to perform application and service discovery, and to launch applications and services.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 May 2004 (13.05.2004)

PCT

(10) International Publication Number
WO 2004/040918 A2

(51) International Patent Classification[7]: H04Q

(21) International Application Number:
PCT/IB2003/004843

(22) International Filing Date: 30 October 2003 (30.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

| | | |
|---|---|---|
| 10/284,135 | 31 October 2002 (31.10.2002) | US |
| 10/662,407 | 16 September 2003 (16.09.2003) | US |

(71) Applicant: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(71) Applicant *(for LC only)*: **NOKIA INC.** [US/US]; 6000 Connection Drive, Irving, TX 75039 (US).

(72) Inventors: **EKBERG, Jan-Erik**; Seljatie 1 A 5, FIN-00320 Espoo (FI). **LAHTINEN, Pekka**; Melkonkatu 7B 50, FIN-00210 Helsinki (FI). **LIPASTI, Jaakko**; Vasikkahaantie 2B, FIN-02420 Jorvas (FI).

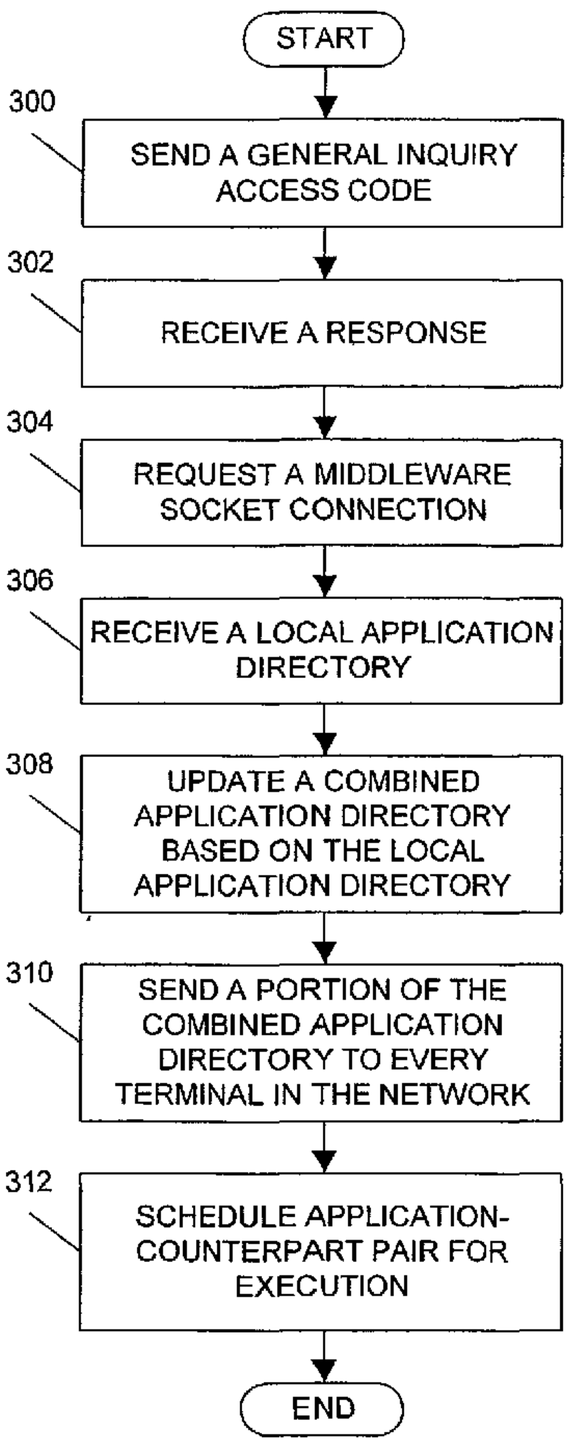
(74) Agent: **WASZKIEWICZ, Kenneth, P.**; Morgan & Finnegan, LLP, 345 Park Avenue, New York, NY 10154 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, 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, 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, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

*[Continued on next page]*

(54) Title: DEVICE DETECTION AND SERVICE DISCOVERY SYSTEM AND METHOD FOR A MOBILE AD HOC COMMUNICATIONS NETWORK




(57) Abstract: A computer system, method, and computer program product for performing device detection and service discovery in a mobile ad hoc communications network. The method comprises conducting an inquiry of the mobile ad hoc communications network to discover nearby devices. If the inquiry indicates that the nearby devices may include a middleware layer, the method further comprises creating a connection to each of the nearby devices and confirming whether each of the nearby devices include the middleware layer. For each of the nearby devices that include the middleware layer, the method further comprises executing the middleware layer to perform application and service discovery, and to launch applications and services.

WO 2004/040918 A2

WO 2004/040918 A2

European patent (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 patent (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.*

# DEVICE DETECTION AND SERVICE DISCOVERY SYSTEM AND METHOD FOR A MOBILE AD HOC COMMUNICATIONS NETWORK

## FIELD OF THE INVENTION

**[0002]** The disclosed invention relates, in general, to communication between devices connected to a wireless communications network. In particular, the disclosed invention is a system and method for performing device detection and service discovery in a mobile ad hoc communications network.

## BACKGROUND OF THE INVENTION

**[0003]** Short-range wireless systems have a range of less than one hundred meters, but may connect to the Internet to provide communication over longer distances. Short-range wireless systems include, but are not limited to, a wireless personal area network (PAN) and a wireless local area network (LAN). A wireless PAN uses low-cost, low-power wireless devices that have a typical range of ten meters. An example of a wireless PAN technology is the Bluetooth Standard. The Bluetooth Standard operates in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band and provides a peak air-link speed of one Mbps and a power consumption low enough for use in personal, portable electronics such as a personal digital assistance or mobile phone. A description of the Bluetooth communication protocol and device operation principles is in *Bluetooth Special Interest Group, Specification of the Bluetooth Standard*, version 1.0B, volumes 1 and 2, December 1999. A wireless LAN is more costly than a wireless PAN, but has a longer range. An example of a wireless LAN technology is the IEEE 802.11 Wireless LAN Standard and the HIPERLAN Standard. The HIPERLAN Standard operates in the 5 GHz Unlicensed-National Information Infrastructure (U-NII) band and provides a peak air-link speed between ten and one hundred Mbps.

**[0004]** An ad hoc network is a short-range wireless system comprising an arbitrary collection of wireless devices that are physically close enough to exchange information. An ad hoc network is constructed quickly with wireless devices joining and leaving the network as they enter and leave the proximity of the remaining wireless devices. An ad hoc network also may include one or more access points, that is, stationary wireless devices operating as a stand-alone server or as gateway connections to other networks.

**[0005]** In the future, the Bluetooth Standard will likely support the interconnection of multiple piconets to form a multi-hop ad hoc network, or scatternet. In a scatternet, a connecting device forwards traffic between different piconets. The connecting device may serve as a master device in one piconet, but as a slave device or a master device in another piconet. Thus, the connecting devices join the piconets that comprise a scatternet by adapting the timing and hop sequence to the respective piconet and possibly changing the roles that they serve from a master device to a slave device.

**[0006]** A Bluetooth device includes, but is not limited to, a mobile telephone, personal or laptop computer, radio-frequency identification tag, and personal electronic device such as a personal digital assistant (PDA), pager, or portable-computing device. Each Bluetooth device includes application and operating system programs designed to find other Bluetooth devices as they enter and leave the communication range of the network. The requesting Bluetooth device in a client role and the responding Bluetooth device in a server role establish a link between the two devices. The requesting and responding Bluetooth device use the link and a service discovery protocol to discover the services offered by the other Bluetooth device and how to connect to those services.

**[0007]** Prior art systems follow similar patterns of behavior for service discovery protocols. A service description, created using a description language and an appropriate vocabulary, is advertised or made available for query matching. Some prior art systems advertise the service description by pushing the description to a directory and requiring the advertisers to discover the directory. Other prior art systems advertise the service description by making the descriptions available for peer-to-peer discovery. A client device that needs to discover the service description composes a query using a query language and a matching vocabulary and uses either a query protocol or a decentralized query-processing server to deliver the query.

**[0008]** Service discovery protocols in the prior art systems require sending and replying to inquiry messages. If no other device is present, the inquiry messages are sent in vain. To avoid excessive power consumption, the prior art systems typically require a human user to manually initiate device detection when another device of interest is present. For example, a human user manually initiates device detection when connecting a cellular telephone to a laptop computer to handle data communications or when connecting a wireless headset to a laptop computer to deliver digital audio. These prior art systems rely upon three assumptions. First, an application can be freely started because the presence of its services is guaranteed. Second, an application performs service discovery when it first needs a service. Third, the composition of the network does not change during the lifetime of the application.

**[0009]** Thus, there is a need for a device detection and service discovery protocol that will avoid excessive power consumption and allow an application resident in one

device to automatically find a counterpart application or some other resource resident in any of the remaining devices within the ad hoc communications network. The protocol does not require a human user to manually initiate device detection to find the counterpart application or other resource. Furthermore, the protocol will accommodate a network environment in which the presence of a particular service is not guaranteed and in which the composition of the network is dynamic because devices frequently enter and leave the network. The disclosed invention addresses this need.

## SUMMARY OF THE INVENTION

**[0010]** A computer system, method, and computer program product for performing device detection and service discovery in a mobile ad hoc communications network. The method comprises conducting an inquiry of the mobile ad hoc communications network to discover nearby devices. If the inquiry indicates that the nearby devices may include a middleware layer, the method further comprises creating a connection to each of the nearby devices and confirming whether each of the nearby devices include the middleware layer. For each of the nearby devices that include the middleware layer, the method further comprises executing the middleware layer to perform application and service discovery, and to launch applications and services.

**[0011]** In one embodiment, the mobile ad hoc communications network is a Bluetooth network. Conducting the inquiry includes sending a Bluetooth inquiry command and receiving a Bluetooth inquiry result command that includes an indication that the device may include the middleware layer. Creating the connection to a device that may include the middleware layer includes sending a Bluetooth paging request message to the device and receiving a Bluetooth paging accept message. Confirming that the device includes the middleware layer includes sending a recognition request message to the device and receiving a recognition response message. Executing the middleware layer to perform application and service discovery includes receiving a notification message from a device with a copy of a local application directory, storing an update to a combined application directory based on a comparison of the local and combined application directory, and sending the update to the combined application directory to each device in the Bluetooth network. In addition, executing the middleware layer includes launching a local application based on a reference in the combined application

directory, and connecting the local application to a counterpart application executing on the device.

**[0011A]** Accordingly, in one aspect of the present invention there is provided a method of performing device detection and service discovery in a mobile ad hoc communications network including at least one network node, each network node storing a local application directory, comprising:

selecting a directory server node from said at least one network node, the directory server node having a coverage area and storing a combined application directory;

sending an inquiry message to a listening node that enters the coverage area;

receiving a notification message from the listening node, the notification message comprising the local application directory stored in the listening node;

storing, in the directory server node, an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory; and

sending an update message to each network node communicating with the mobile ad hoc communications network, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network.

**[0011B]** According to another aspect of the present invention there is provided a method of performing device detection and service discovery in a mobile ad hoc communications network including at least one network node, each network node storing a local application directory, comprising:

selecting a directory server node from said at least one network node, the directory server node having a coverage area and storing a combined application directory;

receiving an inquiry message at a listening node that enters the coverage area;

sending a notification message to the directory server node, the notification message comprising the local application directory stored in the listening node; and

receiving an update message from the directory server node, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network;

wherein the directory server node stores an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory.

**[0011C]** According to yet another aspect of the present invention there is provided an apparatus, comprising:

at least one processor;

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

select a directory server node from at least one network node, the directory server node having a coverage area and storing a combined application directory;

cause sending an inquiry message to a listening node that enters the coverage area;

receive a notification message from the listening node, the notification message comprising the local application directory stored in the listening node;

cause storing, in the directory server node, an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory; and

cause sending an update message to each network node communicating with the mobile ad hoc communications network, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network.

**[0011D]** According to still yet another aspect of the present invention there is provided an apparatus, comprising:

at least one processor; and

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

select a directory server node from at least one network node, the directory server node having a coverage area and storing a combined application directory;

receive an inquiry message at a listening node that enters the coverage area;

cause sending a notification message to the directory server node, the notification message comprising the local application directory stored in the listening node; and

receive an update message from the directory server node, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network;

wherein the directory server node stores an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** The accompanying figures best illustrate the details of the device detection and service discovery system and method for a mobile ad hoc communications network, both as to its structure and operation. Like reference numbers and designations in these figures refer to like elements.

**[0013]** Figure **1** is a network diagram that illustrates the interaction of the devices that comprise a mobile ad hoc communications network, in accordance with one embodiment of the present invention.

**[0014]** Figure **2A** is a block diagram that illustrates the hardware and software components comprising server 110 shown in Figure **1**, in accordance with one embodiment of the present invention.

**[0015]** Figure **2B** is a block diagram that illustrates the hardware and software components comprising terminal **120** shown in Figure **1**, in accordance with one embodiment of the present invention.

**[0016]** Figure **3A** is a flow diagram of an embodiment of server **110** performing device detection and service discovery for a mobile ad hoc communications network.

**[0017]** Figure **3B** is a flow diagram of an embodiment of terminal **120** performing device detection and service discovery for a mobile ad hoc communications network.

**[0018]** Figure **4A** is an exemplary block diagram of the data flow before a terminal enters a mobile ad hoc communications network.

**[0019]** Figure **4B** shows the exemplary block diagram of Figure **4A** after the terminal enters the mobile ad hoc communications network.

**[0020]** Figure **5** is a flow diagram of an embodiment of a process that illustrates the message flow during establishment of a communication session between terminal X and terminal Y in a mobile ad hoc communications network.

## DETAILED DESCRIPTION OF THE INVENTION

**[0021]** Figure **1** is a network diagram that illustrates the interaction of the devices that comprise a mobile ad hoc communications network, in accordance with one embodiment of the present invention. In one embodiment, the mobile ad hoc communications network is a Bluetooth piconet that includes one master device and up to seven active slave devices. As shown in Figure **1**, piconet **100** includes server **110** and five instances of terminal **120**. Server **110** maintains the network clock and is the communication manager for each instance of terminal **120**. Server **110** typically initiates an exchange of data with an instance of terminal **120**. Two instances of terminal **120** typically communicate through the server **110** however, if two instances of terminal **120** communicate directly, one instance will assume the role of server, or master, and the other instance will assume the role of client, or slave.

**[0022]** Each device in the mobile ad hoc communications network will either assume the role of a terminal device or a server device. A terminal device is a consumer of services that a single user operates. A terminal device includes devices such as a mobile phone or PDA. A server is typically a stationary device and only produces services. A server device creates a hotspot around them for using their services. "Hotspot" refers to the radio coverage area provided by the server device for detecting devices and discovering services offered by the applications hosted in the server. If the server device is not stationary, one of the terminal devices in the network will assume the role of application directory server and perform device detection and service discovery functions for the remaining terminal devices in the network. The disclosed invention introduces two roles among such terminal devices, application directory servers and terminals, where application directory servers serve terminals in device detection and service discovery. If stationary servers with hotspots exist, servers typically act as application directory servers however, device detection and service discovery is possible without such a stationary server because one of the terminals will assume the application directory server duties.

**[0023]** The disclosed invention categorizes an application as a server-based application, terminal-to-terminal application, foreground application, background application, or generic application component. A server-based application requires a

server to produce a service. A terminal-to-terminal application requires at least two terminal devices to implement a service without the presence of a server device. A foreground application is an application resident in a terminal device that a user accesses via the user interface of the terminal device. A background application is an application resident in a terminal device that may start without any intervention by the user. A generic application component can be used either as a standalone application or as a component of another application.

**[0024]** An application may be further categorized as either active, passive, new, or rejected. An active application is a foreground or background application that is resident in (i.e., stored in memory) the terminal. A passive application is resident in the terminal, but has not yet been started. In another embodiment, the passive application is started, but is not actively looking for other instances of the same application. A new application is not yet resident in the terminal, but might be in the future. A rejected application is not resident in the terminal and has been marked by the user as an application that should never be resident in the terminal. In another embodiment, the rejected application was once resident in the terminal, but was subsequently deleted and marked as rejected. In yet another embodiment, the rejected application never resided in the terminal, but is of a type of application that the user has marked as rejected.

**[0025]** Service discovery in a mobile ad hoc communications network differentiates between a resident application and an unloaded application. A resident application is stored in the terminal memory and loaded as either a foreground application or a background application. An unloaded application is not yet stored or loaded in the terminal, but has been accepted by the user. Typically, when an application was previously used, but has been overwritten to reclaim space, the application is considered unloaded. Thus, starting an unloaded application may require first downloading the application.

**[0026]** Service discovery from the perspective of the terminal device requires categorizing the status of an application as either an active resident application, active unloaded application, passive resident application, passive unloaded application, rejected application, or new application. An active resident application is loaded in the terminal and looking for peers, servers, or clients. An active unloaded application is not loaded in

the terminal, but is still looking for such counterpart applications that could be automatically downloaded if found interesting. A passive resident application is loaded in the terminal, but is not looking for counterpart applications. A passive unloaded application is not loaded in the terminal, but was once accepted by the user. A rejected application is an application that a user has requested to exclude from the terminal device. A new application is not loaded in the terminal device, but the user might have seen an application in an earlier server for instance.

**[0027]** Figure **2A** is a block diagram that illustrates the hardware and software components comprising server **110** shown in Figure **1**, in accordance with one embodiment of the present invention. Server **110** is a general-purpose wireless device. Bus **200** is a communication medium that connects keypad **201**, display **202**, central processing unit (CPU) **203**, and radio frequency (RF) adapter **204** to memory **210**. RF adapter **204** connects via a wireless link to terminal **120** and is the mechanism that facilitates network traffic between server **110** and terminal **120**.

**[0028]** CPU **203** performs the methods of the disclosed invention by executing the sequences of operational instructions that comprise each computer program resident in, or operative on, memory **210**. Memory **210** includes operating system software **211**, application programs **212**, and middleware software **220**. Operating system software **211** controls keypad **201**, display **202**, RF adapter **204**, and the management of memory **210**. Application programs **212** control the interactions between a user and server **110**. Middleware software **220** includes an application program interface (API) **221** that help an application program running on server **110** find and communicate with a counterpart application running on terminal **120**. To quickly locate each application, middleware software **220** also includes application directory **230** to track the role assumed by each application that is resident in each device in piconet **100**.

**[0029]** Figure **2B** is a block diagram that illustrates the hardware and software components comprising terminal **120** shown in Figure **1**, in accordance with one embodiment of the present invention. Terminal **120** is a general-purpose wireless device. Bus **250** is a communication medium that connects keypad **251**, display **252**, CPU **253**, and RF adapter **254** to memory **260**. RF adapter **254** connects via a wireless link to server

**110** or another terminal **120** and is the mechanism that facilitates network traffic between server **110** and terminal **120**.

**[0030]** CPU **253** performs the methods of the disclosed invention by executing the sequences of operational instructions that comprise each computer program resident in, or operative on, memory **260**. Memory **260** includes operating system software **261**, application programs **262**, and middleware software **270**. Operating system software **261** controls keypad **251**, display **252**, RF adapter **254**, and the management of memory **260**. Application programs **262** control the interactions between a user and terminal **120**. Middleware software **270** includes an API **271** that help an application program running on terminal **120** find and communicate with a counterpart application running on server **110** or another terminal **120**. To quickly locate each application, middleware software **270** also includes application directory **280** to track the role assumed by each application that is resident in each device in piconet **100**.

**[0031]** In one embodiment, the configuration of memory **210** and memory **260** is identical. In another embodiment, the configuration of memory **210** and memory **260** only includes the software necessary to perform the essential tasks of server **110** and terminal **120**, respectively. For example, if terminal **120** needs to receive a general inquiry access code, but does not need to send a general inquiry access code message, only the software that receives this message will reside in memory **260**.

**[0032]** An application executing on a terminal is constantly searching for a counterpart application, that is, another instance of the same application that can communicate with the application. Each instance of an application assumes a particular role. Communication between an application and a counterpart application is only meaningful if the roles are complementary. For example, an application that assumes the role of "client" can communicate with a counterpart application that assumes the role of "server". Middleware software is a software layer with an API that negotiates the communication between two applications to help an application find a counterpart application with the correct role. Thus, an application installed in a terminal and activated, will query the API for a continuous stream of new counterpart applications that are of interest.

**[0033]** A new application is installed by "installer" applications that use middleware for finding counterparts and installing the new application into the local storage of a terminal. The actual finding and selection of new applications takes place in the application level. Initially, the installer application will be a dedicated "browser-supplier" (i.e., client-server) application that accesses counterpart applications in servers, browses their available application databases, allows a user to pick the applications to install, and downloads and installs the new applications. Later, the corresponding functionality may be added to a wireless access protocol (WAP) and hypertext markup language (HTML) browsers.

**[0034]** Service discovery is viewed as a three-step process. First, new potential applications are found and will be considered for installation. Second, active installed applications begin to search for counterpart application. Third, active installed applications begin searching for common resources such as printers (i.e., resource discovery). The disclosed invention relies upon the applications to perform resource discovery. Typically, a terminal application communicates with its counterpart application and use local (i.e., server) resources. If an application uses a private resource, the associated service discovery is implemented by the application in a standard (e.g., Bluetooth or Bluetooth/Java) way not supported by the terminal middleware software.

**[0035]** Figure **3A** is a flow diagram of an embodiment of server **110** performing device detection and service discovery for a mobile ad hoc communications network. The process begins when server **110** sends a general inquiry access code message to terminal **120** (step **300**). Terminal **120** receives the message and sends an acknowledgment response message to server **110** (step **302**). Server **110** accesses middleware software **220** to request a socket connection with terminal **120** (step **304**). In response to establishing the socket connection, server **110** receives a message from terminal **120** that includes a local application directory listing all of the applications that are locally resident on terminal **110** (step **306**). Server **110** compares the list of applications resident on terminal **120** to a combined application directory resident on server **110**. Server **110** updates the combined application directory by adding to the combined application directory each entry in the local application directory that does not appear in the combined application directory (step **308**). Server **110** sends a portion of the updated combined application directory to each terminal **120** in piconet **100** (step

310). The portion may vary for each terminal **120** and includes each entry in the combined application directory that is a counterpart application to an application resident in terminal **120**. In another embodiment, server **110** sends the entire combined application directory to each terminal **120** in piconet **100** and relies upon terminal **120** to retain the pertinent entries. Instances of middleware software in terminal **120** and server **110** begin to schedule the newly found counterpart application pairs for execution (step **312**). In one embodiment, the scheduled applications make use of any other Bluetooth profile and protocol. In another embodiment, an application that is an installer application may suggest to the user other applications that the user should download. Once server **110** downloads and starts a new application, counterpart matching repeats and the new application becomes a part of the middleware scheduling.

**[0036]** Figure **3B** is a flow diagram of an embodiment of terminal **120** performing device detection and service discovery for a mobile ad hoc communications network. The process begins when terminal **120** receives a general inquiry access code message from server **110** (step **320**). Terminal **120** generates and sends an acknowledgment response message to server **110** (step **322**). Terminal **120** sends a message to server **110** that includes a local application directory that includes all of the applications that are locally resident on terminal **110** (step **324**). Server **110** compares the list of applications resident on terminal **120** to a combined application directory resident on server **110**. Server **110** updates the combined application directory by adding to the combined application directory each entry in the local application directory that does not appear in the combined application directory. Terminal **120** receives from server **110** a portion of the updated combined application directory (step **326**). Server **110** customizes the portion for terminal **120** to include each entry in the combined application directory that is a counterpart application to an application resident in terminal **120**. In another embodiment, server **110** sends the entire combined application directory to terminal **120** and relies on terminal **120** to retain the pertinent entries. Instances of middleware software in terminal **120** and server **110** begin scheduling these newly found counterpart application pairs for execution (step **328**).

**[0037]** Figures **4A** and **4B** are exemplary block diagrams showing the content of the application directory before and after terminal X and terminal Y enter a mobile ad hoc communications network served by server S. Figure **4A** shows the configuration of

application directory **404**, application directory **415**, and application directory **425** before terminal X and terminal Y enter a communication network managed by server S, a master device. Application C **401** resides in server S memory **400** and accesses middleware software **403** via API **402**. Middleware software **403** registers application C **401** with application directory **404** by adding a table entry to indicate that application C resides in the local device (i.e., server S) and assumes the role of server. Application A **411** and application B **412** reside in terminal X memory **410** and access middleware software **414** via API **413**. Middleware software **414** registers application A **411** and application B **412** with application directory **415** by adding a table entry to indicate that application A resides in the local device (i.e., terminal X) and assumes the role of client and that application B resides in the local device (i.e., terminal X) and assumes the role of peer. Application B **421** and application C **422** reside in terminal Y memory **420** and access middleware software **424** via API **423**. Middleware software **424** registers application B **421** and application C **422** with application directory **425** by adding a table entry to indicate that application B resides in the local device (i.e., terminal Y) and assumes the role of peer and that application C resides in the local device (i.e., terminal Y) and assumes the role of client.

**[0038]** Figure **4B** shows the configuration of application directory **404**, application directory **415**, and application directory **425** after terminal X and terminal Y enter the communication network managed by server S, a master device. Server S assumes the role of an application directory server (ADS) and mediates the registration of the applications residing in each device in piconet **100**. Server S adds a table entry to application directory **404** for each application residing in a device on piconet **100**. Thus, server S adds an entry for application A residing in terminal X in a client role, application B residing in terminal X in a peer role, application B residing in terminal Y in a peer role, and application C residing in terminal Y in a client role. Server S also updates application directory **415** in terminal X and application directory **425** in terminal Y with application registrations that may be interesting to those terminal devices. As shown in Figure **4B**, terminal X and terminal Y both host application B in a peer role. Since, a peer-to-peer communication session between application B on terminal X and application B on terminal Y is likely, server S adds an entry to application directory **415** for application B residing in terminal Y in a peer role and an entry to application directory

**425** for application B residing in terminal X in a peer role. Also, since a client-server communication session between application C on terminal Y and application C on server S is likely, server S adds an entry to application directory **425** for application C residing in server S in a server role. Finally, there is no counterpart in piconet **100** for application A on terminal X.

**[0039]** As shown in Figures **4A** and **4B**, the disclosed data items for each entry in the middleware software application directory server include a device identifier (e.g., "local", an address, or other unique identifier), an application identifier (e.g., application name or other unique identifier), and a role for the application (e.g., "client", "server", "peer", etc.). In another embodiment, the data items can be expanded to include fields for the local applications (i.e., device = "local") and fields for remote applications in other terminals or servers. The fields for the local applications include:

- Name – An identifier for the application (e.g., supplier name and data to compare different versions and hardware variants);
- My_role – The role that the application takes in the local device;
- Partner_role – The role that the application assumes from interesting counterparts (e.g., peer, client, and server are the most common roles);
- Residency – Either RESIDENT (installed and currently in memory), UNLOADED (installed once, not currently in memory, but can be re-downloaded automatically), REJECTED (indicates to the new application installer that it should ignore the application), or NEW (the application is not installed or rejected);
- State – Either RUNNING (has communications, is now working with its remote counterparts, but there may be either only one, or more, applications that can use the communications at a time), WAITING (in execution but does not have any communications), STARTABLE (active, if a matching peer with the right partner_role is found, the middleware software starts this application, downloading the software first if needed), COMPLETE (all counterpart applications are aware), or PASSIVE (user must do something to start application);

- Type – Either FOREGOUND (when the application terminates, the state will be PASSIVE), or BACKGROUND (if the application terminates, the state will be STARTABLE);
- Unload – Either AUTOMATIC (middleware may remove code when the application has terminated), or UNINSTALL (user must confirm removals);
- Icon – Creates a visual image of the application for the user; and
- Timeout – Sets a time limit that the middleware software uses to detect, for example, when the application is in an unproductive software loop.

**[0040]** The fields for the remote applications include:

- Device – An address for establishing communications with the terminal or server storing the application instance;
- Name – An identifier for the application; and
- My_role – The role that the application takes in the remote device.

**[0041]** The client-server roles of the applications are independent of the roles of the devices as a terminal device and an application directory server. Typically, the device acting as an application directory server hosts applications acting in a server role and the terminal devices act in the client role for the same application. In another embodiment, two terminal devices each send a general inquiry access code message and listen for a reply. The terminal device that receives a response first will assume the server role and proceed according to the procedure in Figure **3A**. Another terminal device that receives the inquiry message will assume the terminal role, and proceed according to Figure **3B**. Thus, the disclosed invention supports terminal-to-terminal scenarios (e.g., one of identical handheld devices automatically becoming an ADS) and does not require predetermined application directory servers.

**[0042]** Figure **5** is a flow diagram of an embodiment of a process that illustrates the message flow during establishment of a communication session between terminal X and terminal Y in a mobile ad hoc communications network. In one embodiment, terminal X and terminal Y are mobile devices such as terminal **120** shown in Figure **1** and Figure **2B**. In another embodiment, terminal X is a mobile device such as terminal **120** shown in Figure **1** and Figure **2B** and terminal Y is a mobile device such as server **110** shown in Figure **1** and Figure **2A**.

**[0043]** As shown in Figure **5**, terminal X initiates the communication by sending an inquiry request message to the mobile ad hoc communications network. Since terminal Y is a nearby device, terminal Y receives the inquiry request message and sends an inquiry response message to terminal X. In one embodiment, the inquiry request message is a Bluetooth inquiry command and the inquiry response message is a Bluetooth inquiry result command. In another embodiment, the inquiry request message is a Bluetooth inquiry command and the inquiry response message is a Bluetooth inquiry result command modified to indicate that the terminal sending the Bluetooth inquiry result command includes a middleware layer. In one embodiment, the middleware layer includes dedicated middleware software providing advanced application and service discovery and execution. In one embodiment, the modification to the Bluetooth inquiry result command is to the Class of Device (CoD) parameters. For example, if the terminal sending the Bluetooth inquiry result command includes the middleware layer, the terminal will set at least the "ad hoc networking aware" bit (bit 16) to on (1). Alternatively, if the terminal sending the Bluetooth inquiry result command includes the middleware layer, the terminal will set the "ad hoc networking aware" bit (bit 16) to on (1), and the "location info" bit (bit 17) to off (0). Alternatively, if the terminal sending the Bluetooth inquiry result command includes the middleware layer, the terminal will set the "ad hoc networking aware" bit (bit 16) to on (1), and the "telephony capable" bit (bit 22) to on (1). Alternatively, if the terminal sending the Bluetooth inquiry result command includes the middleware layer, the terminal will set the "ad hoc networking aware" bit (bit 16) to on (1), the "location info" bit (bit 17) to off (0), and the "telephony capable" bit (bit 22) to on (1). In yet another embodiment, the modification to the Bluetooth inquiry result command is not necessary, if a dedicated indication parameter to indicate the presence of the middleware software is introduced to the Bluetooth inquiry result command specifications.

**[0044]** Following the inquiry, as shown in Figure **5**, terminal X may create a connection to each nearby device indicating possible possession of the middleware layer by the inquiry response message, such as terminal Y, by sending a paging request message. If terminal Y does not indicate possible possession of the middleware layer (e.g., by setting the "ad-hoc networking aware" bit (bit 16) to off (0)), no paging request message is transmitted and the communication session is disconnected. After conducting

an inquiry including an indication that terminal Y possibly includes a middleware layer, terminal X sends the paging message request, as discussed above. Terminal Y receives the paging request message and optionally sends a paging accept message to accept the connection request. In one embodiment, the paging request message is a Bluetooth create connection command and the paging accept message is a Bluetooth accept connection request command.

**[0045]** Following the connection to each nearby device, as shown in Figure **5**, terminal X sends a recognition request message to confirm whether a nearby device such as terminal Y definitely includes the middleware layer. Terminal Y receives the recognition request message and sends a recognition response message to terminal X. In one embodiment, the receipt of the recognition response message is confirmation that terminal Y includes the middleware layer. In another embodiment, the content of the recognition response message will indicate whether terminal Y includes the middleware layer. In one embodiment, the recognition request message and the recognition response message utilize the Bluetooth Service Discovery Protocol (SDP). If terminal Y does not include the middleware layer, the communication session may be disconnected.

**[0046]** Following the confirmation that a nearby device such as terminal Y includes the middleware layer, as shown in Figure **5**, terminal X and terminal Y use the middleware layer to discover and launch applications and services. In one embodiment, terminal X and terminal Y use the methods disclosed in the flow diagrams shown in Figure **3A** and Figure **3B** to discover and launch applications and services.

**[0047]** Although the disclosed embodiments describe a fully functioning device detection and service discovery system and method for a mobile ad hoc communications network, the reader should understand that other equivalent embodiments exist. Since numerous modifications and variations will occur to those who review this disclosure, the device detection and service discovery system and method for a mobile ad hoc communications network is not limited to the exact construction and operation illustrated and disclosed. Accordingly, this disclosure intends all suitable modifications and equivalents to fall within the scope of the claims.

**What is claimed is:**

1. A method of performing device detection and service discovery in a mobile ad hoc communications network including at least one network node, each network node storing a local application directory, comprising:

selecting a directory server node from said at least one network node, the directory server node having a coverage area and storing a combined application directory;

sending an inquiry message to a listening node that enters the coverage area;

receiving a notification message from the listening node, the notification message comprising the local application directory stored in the listening node;

storing, in the directory server node, an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory; and

sending an update message to each network node communicating with the mobile ad hoc communications network, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network.

2. The method of claim 1, wherein the combined application directory stores a reference to each application program stored in said at least one network node, and wherein the local application directory stores a reference to each application program stored in the network node associated with the local application directory.

3. The method of claim 2, wherein the reference stored in the combined application directory, and the reference stored in the local application directory, include an application identifier, a network node identifier, and an application role.

4. A method of performing device detection and service discovery in a mobile ad hoc communications network including at least one network node, each network node storing a local application directory, comprising:

selecting a directory server node from said at least one network node, the directory server node having a coverage area and storing a combined application directory;

receiving an inquiry message at a listening node that enters the coverage area;

sending a notification message to the directory server node, the notification message comprising the local application directory stored in the listening node; and

receiving an update message from the directory server node, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network;

wherein the directory server node stores an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory.

5. The method of claim 4, wherein the combined application directory stores a reference to each application program stored in said at least one network node, and wherein the local application directory stores a reference to each application program stored in the network node associated with the local application directory.

6. The method of claim 5, wherein the reference stored in the combined application directory, and the reference stored in the local application directory, include an application identifier, a network node identifier, and an application role.

7. An apparatus, comprising:

at least one processor; and

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

select a directory server node from at least one network node, the directory server node having a coverage area and storing a combined application directory;

cause sending an inquiry message to a listening node that enters the coverage area;

receive a notification message from the listening node, the notification message comprising the local application directory stored in the listening node;

cause storing, in the directory server node, an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory; and

cause sending an update message to each network node communicating with the mobile ad hoc communications network, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network.

8. The apparatus of claim 7, wherein the combined application directory stores a reference to each application program stored in said at least one network node, and wherein the local application directory stores a reference to each application program stored in the network node associated with the local application directory.

9. The apparatus of claim 8, wherein the reference stored in the combined application directory, and the reference stored in the local application directory, include an application identifier, a network node identifier, and an application role.

10. An apparatus, comprising:

at least one processor; and

at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

select a directory server node from at least one network node, the directory server node having a coverage area and storing a combined application directory;

receive an inquiry message at a listening node that enters the coverage area;

cause sending a notification message to the directory server node, the notification message comprising the local application directory stored in the listening node; and

receive an update message from the directory server node, the update message comprising an update portion of the combined application directory for updating the local application directories of each of the nodes within the mobile ad hoc communications network,

wherein the directory server node stores an update to the combined application directory based on a comparison of the local application directory included with the notification message and the combined application directory.

11. The apparatus of claim 10, wherein the combined application directory stores a reference to each application program stored in said at least one network node, and wherein the local application directory stores a reference to each application program stored in the network node associated with the local application directory.

12. The apparatus of claim 11, wherein the reference stored in the combined application directory, and the reference stored in the local application directory, include an application identifier, a network node identifier, and an application role.

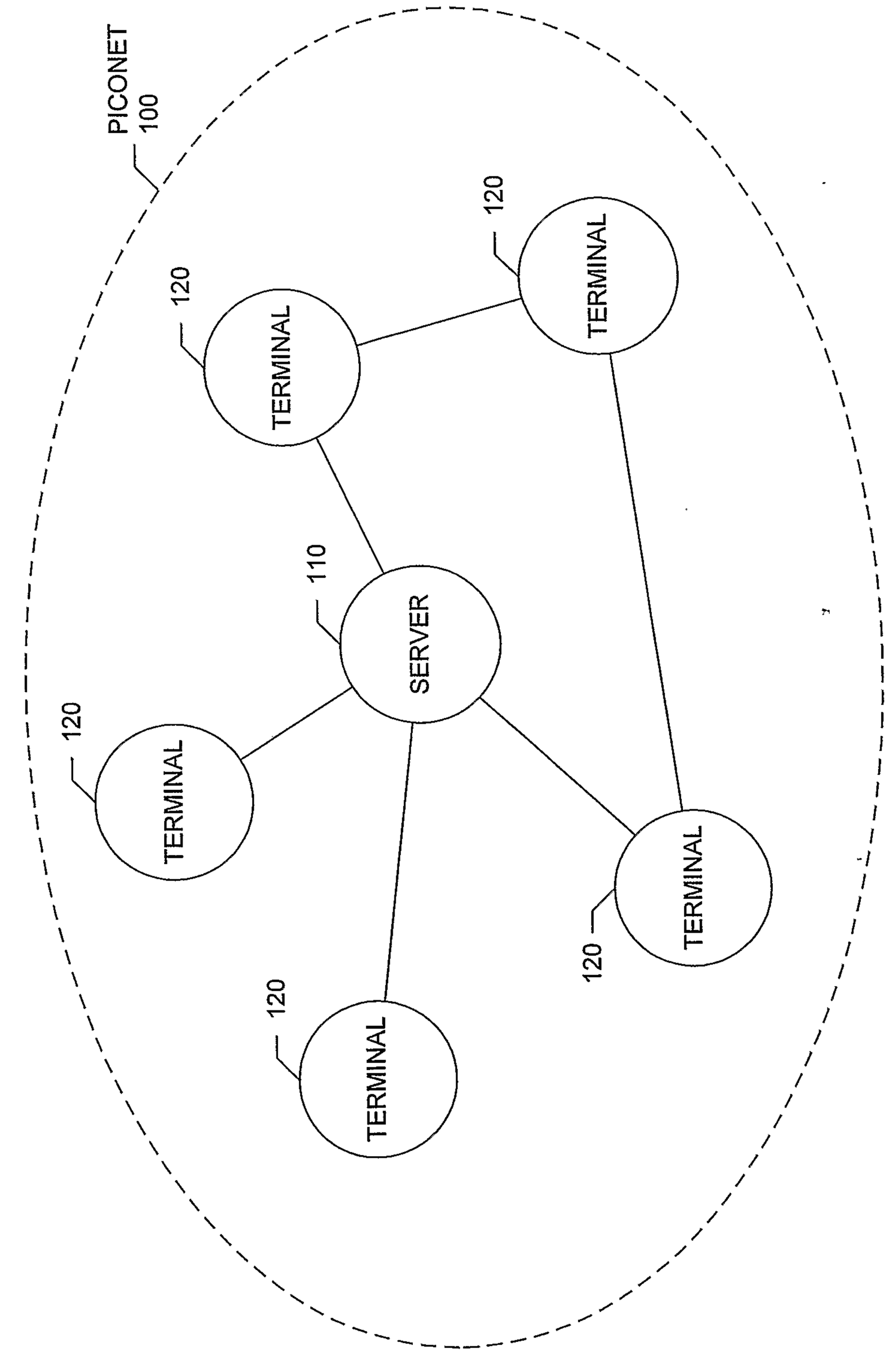

PICONET
100
SERVER
110
TERMINAL
120
TERMINAL
120
TERMINAL
120
TERMINAL
120
TERMINAL
120
TERMINAL
120

Figure 1

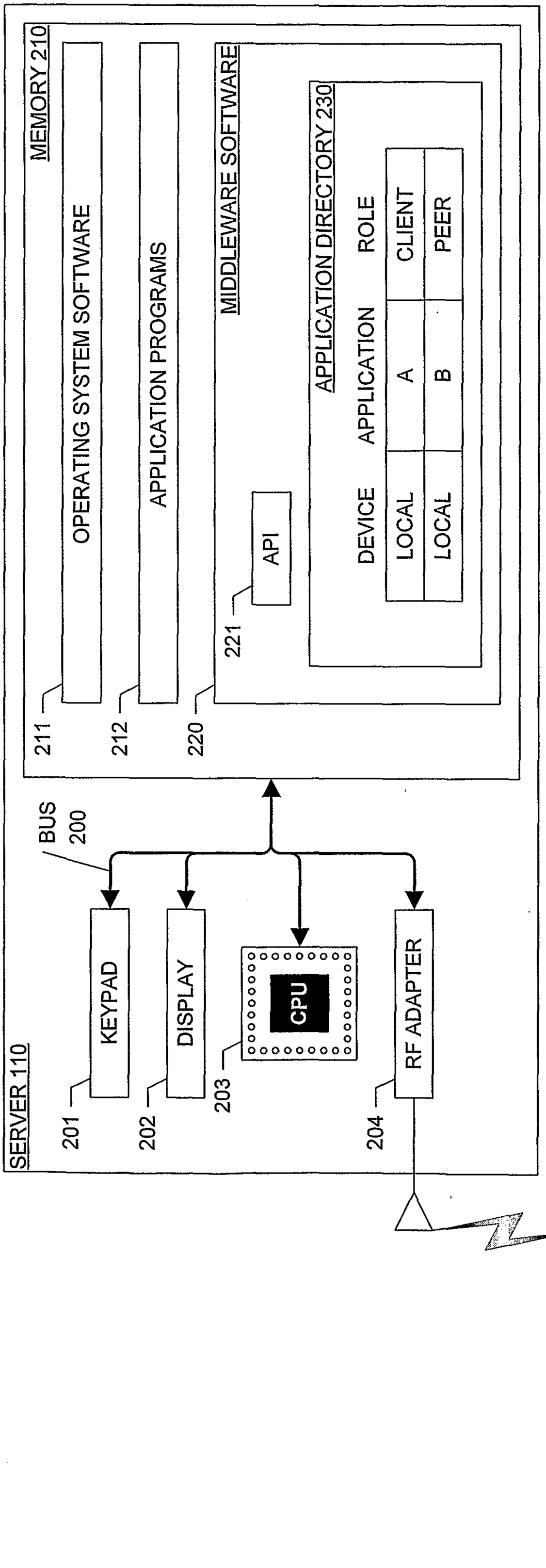

SERVER 110
BUS
200
KEYPAD
201
DISPLAY
202
CPU
203
RF ADAPTER
204
MEMORY 210
OPERATING SYSTEM SOFTWARE
211
APPLICATION PROGRAMS
212
MIDDLEWARE SOFTWARE
220
API
221
APPLICATION DIRECTORY 230
DEVICE
APPLICATION
ROLE
LOCAL
A
CLIENT
LOCAL
B
PEER
TERMINAL
120

Figure 2A

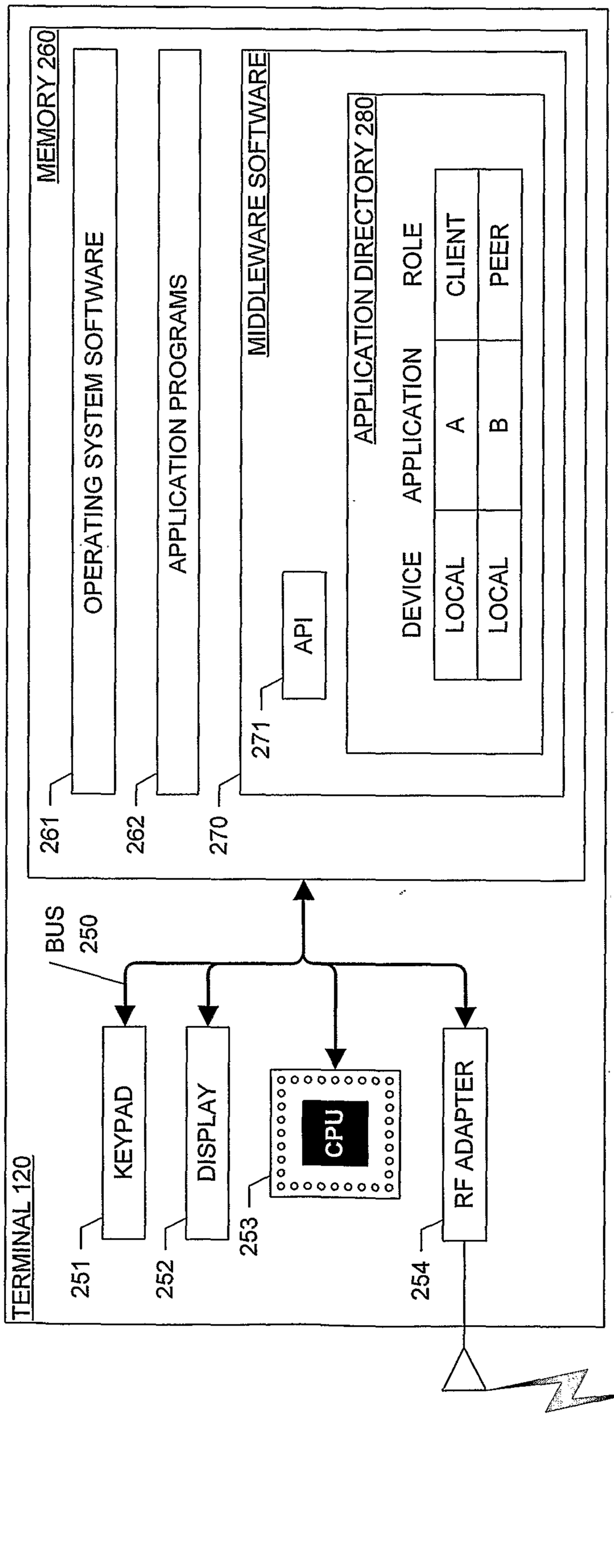
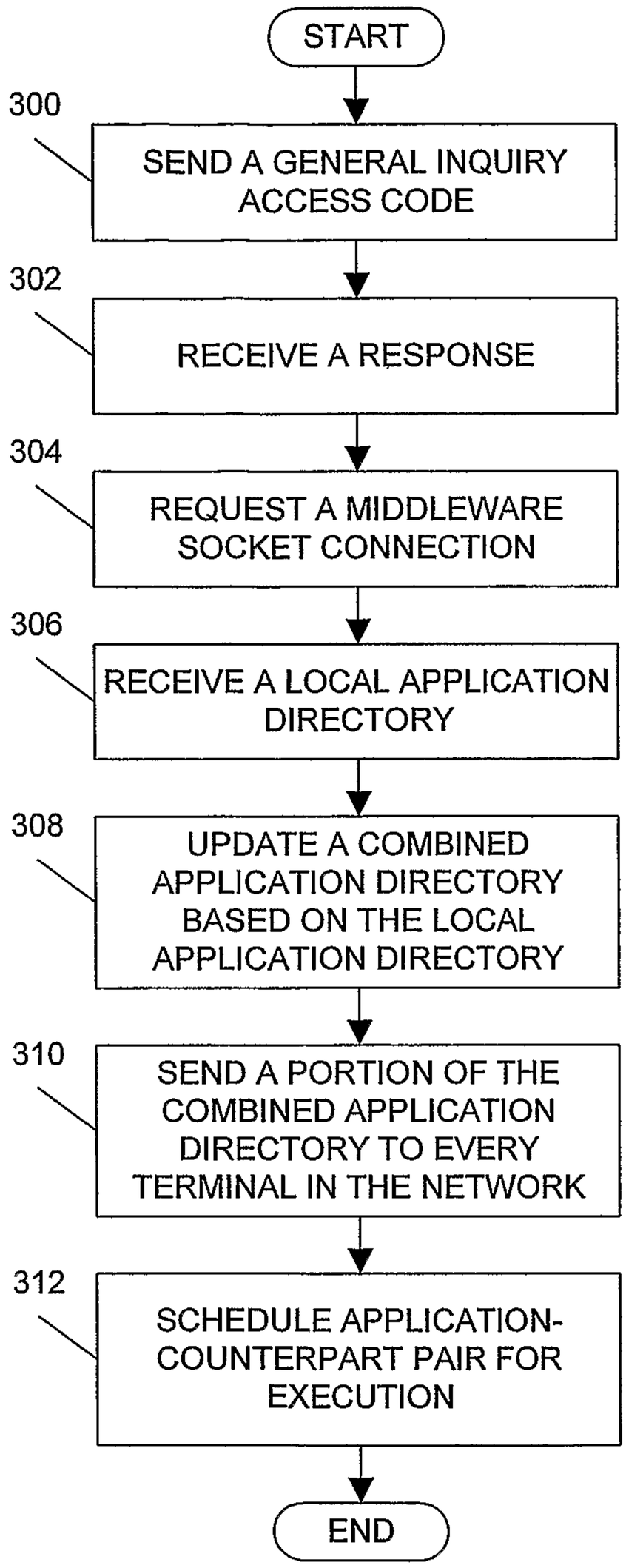
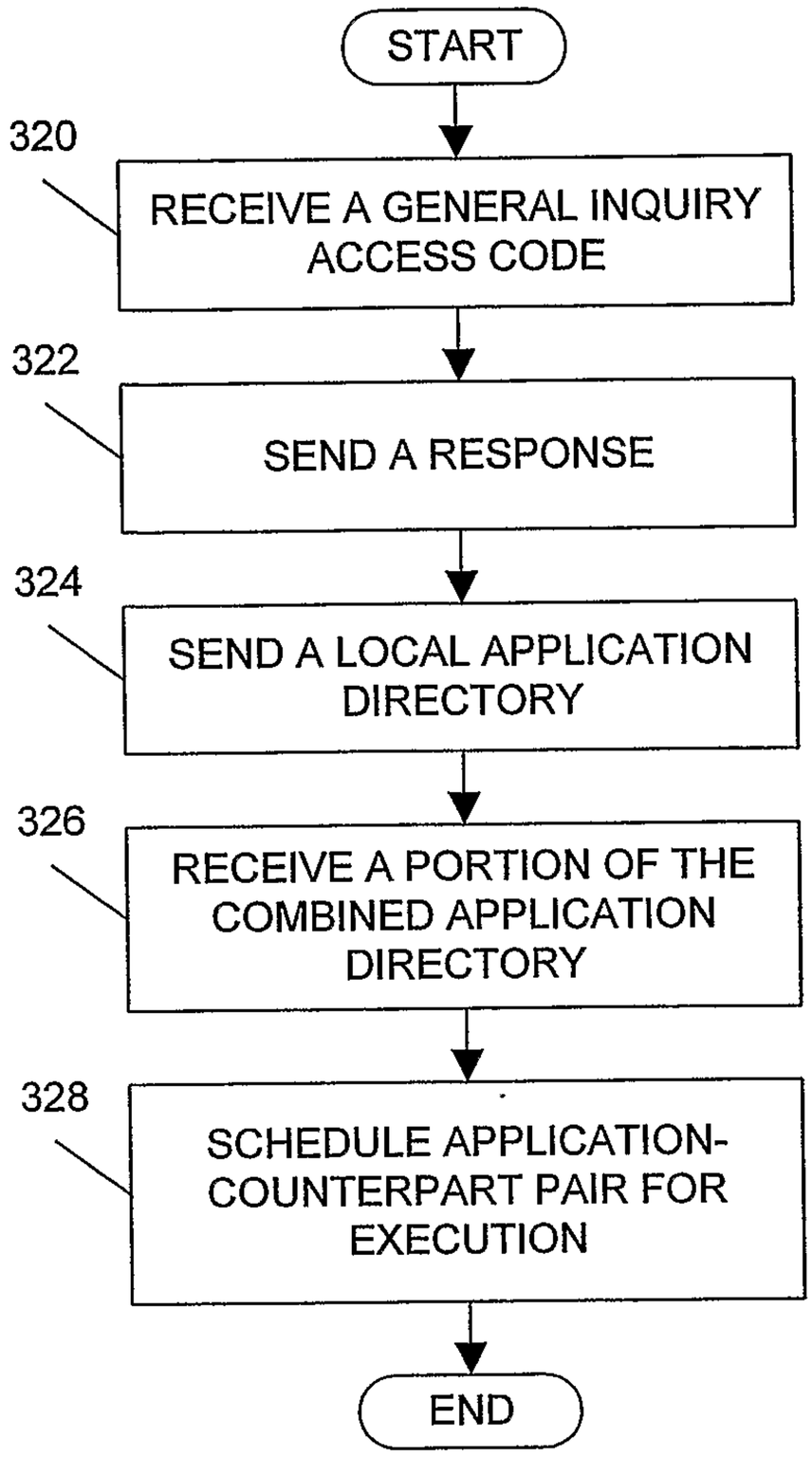

MEMORY 260
OPERATING SYSTEM SOFTWARE
APPLICATION PROGRAMS
MIDDLEWARE SOFTWARE
APPLICATION DIRECTORY 280
ROLE
CLIENT
PEER
APPLICATION
A
B
DEVICE
LOCAL
LOCAL
271
API
261
262
270
BUS
250
TERMINAL 120
KEYPAD
DISPLAY
CPU
RF ADAPTER
251
252
253
254
SERVER
110

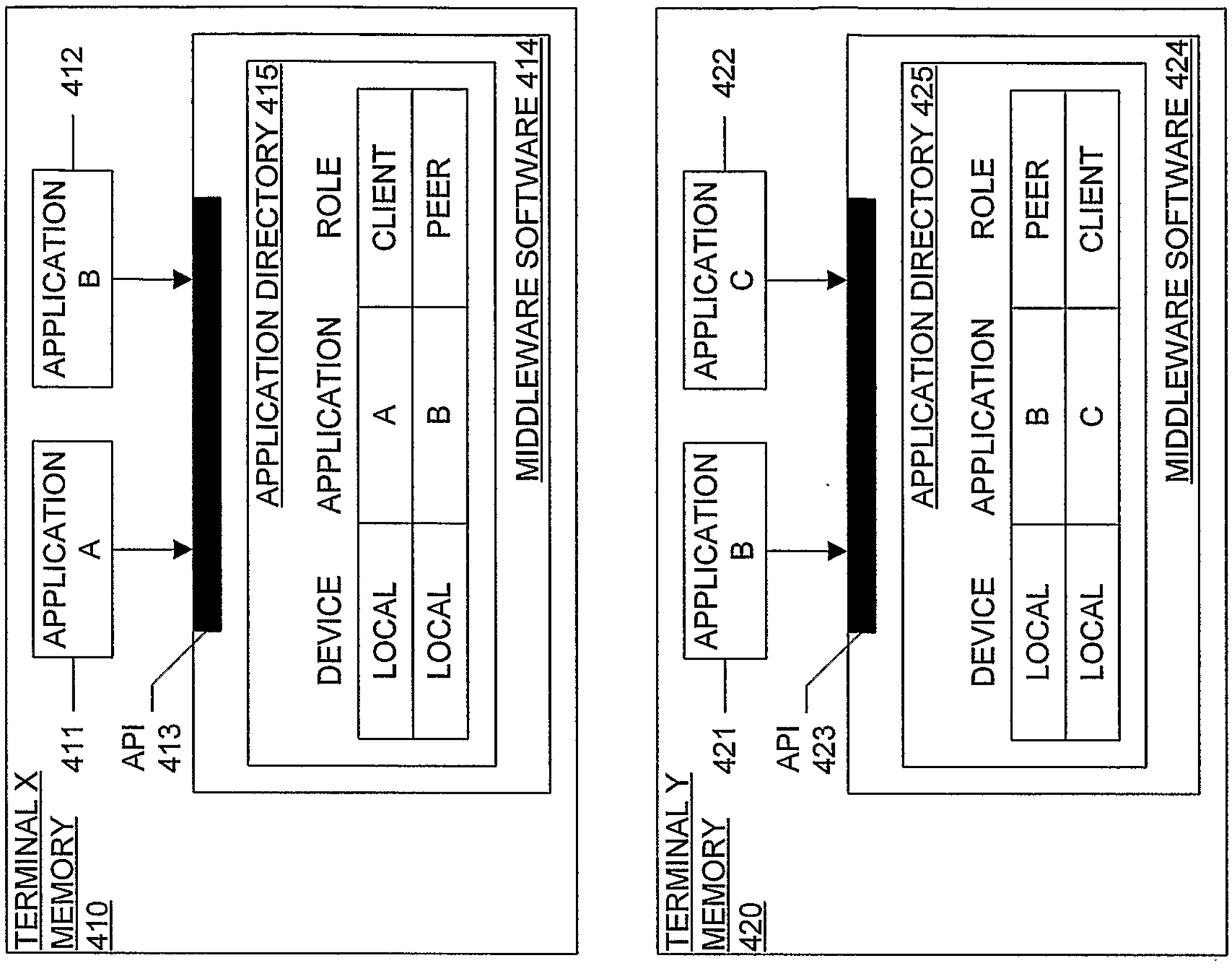
Figure 2B

# Figure 3A


START
300
SEND A GENERAL INQUIRY ACCESS CODE
302
RECEIVE A RESPONSE
304
REQUEST A MIDDLEWARE SOCKET CONNECTION
306
RECEIVE A LOCAL APPLICATION DIRECTORY
308
UPDATE A COMBINED APPLICATION DIRECTORY BASED ON THE LOCAL APPLICATION DIRECTORY
310
SEND A PORTION OF THE COMBINED APPLICATION DIRECTORY TO EVERY TERMINAL IN THE NETWORK
312
SCHEDULE APPLICATION-COUNTERPART PAIR FOR EXECUTION
END

# Figure 3B


START
320
RECEIVE A GENERAL INQUIRY ACCESS CODE
322
SEND A RESPONSE
324
SEND A LOCAL APPLICATION DIRECTORY
326
RECEIVE A PORTION OF THE COMBINED APPLICATION DIRECTORY
328
SCHEDULE APPLICATION-COUNTERPART PAIR FOR EXECUTION
END


TERMINAL X
MEMORY
410
APPLICATION A
411
API
413
APPLICATION B
412
MIDDLEWARE SOFTWARE 414
APPLICATION DIRECTORY 415
DEVICE
APPLICATION
ROLE
LOCAL
A
CLIENT
LOCAL
B
PEER
TERMINAL Y
MEMORY
420
APPLICATION B
421
API
423
APPLICATION C
422
MIDDLEWARE SOFTWARE 424
APPLICATION DIRECTORY 425
DEVICE
APPLICATION
ROLE
LOCAL
B
PEER
LOCAL
C
CLIENT

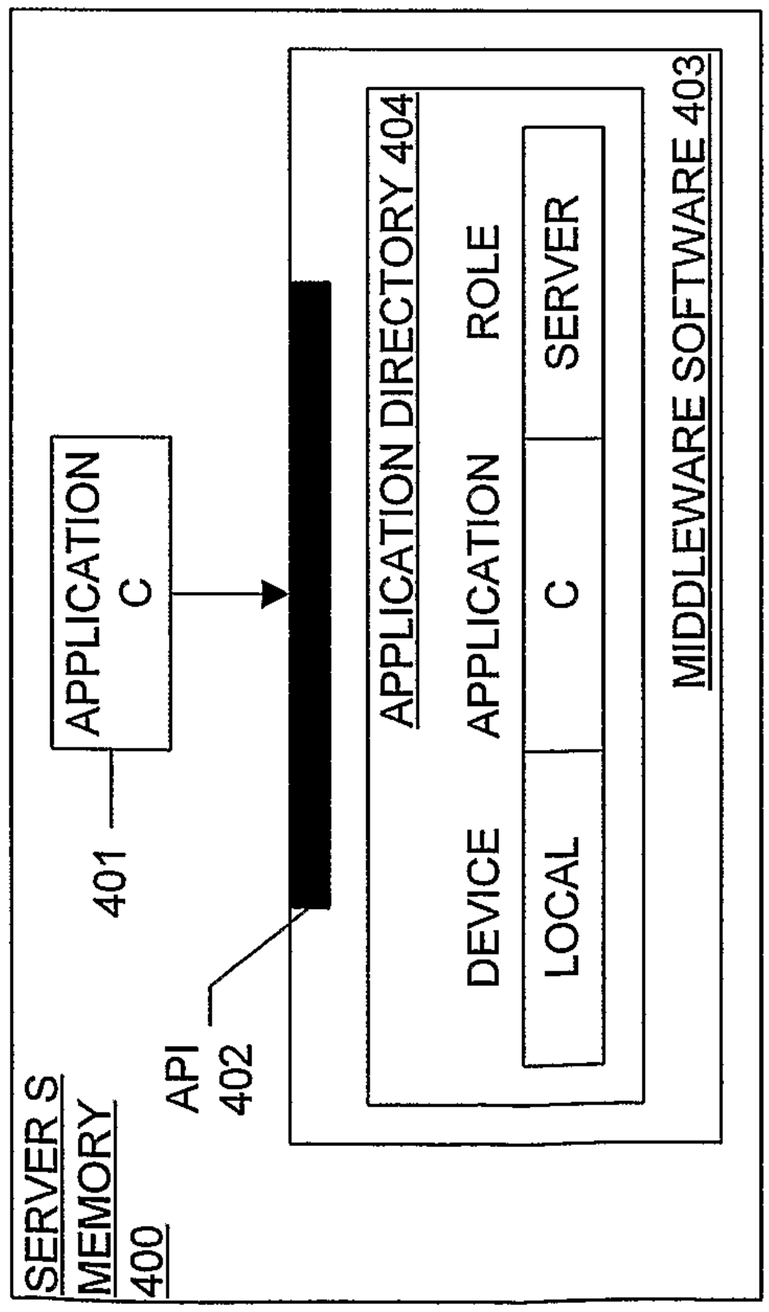
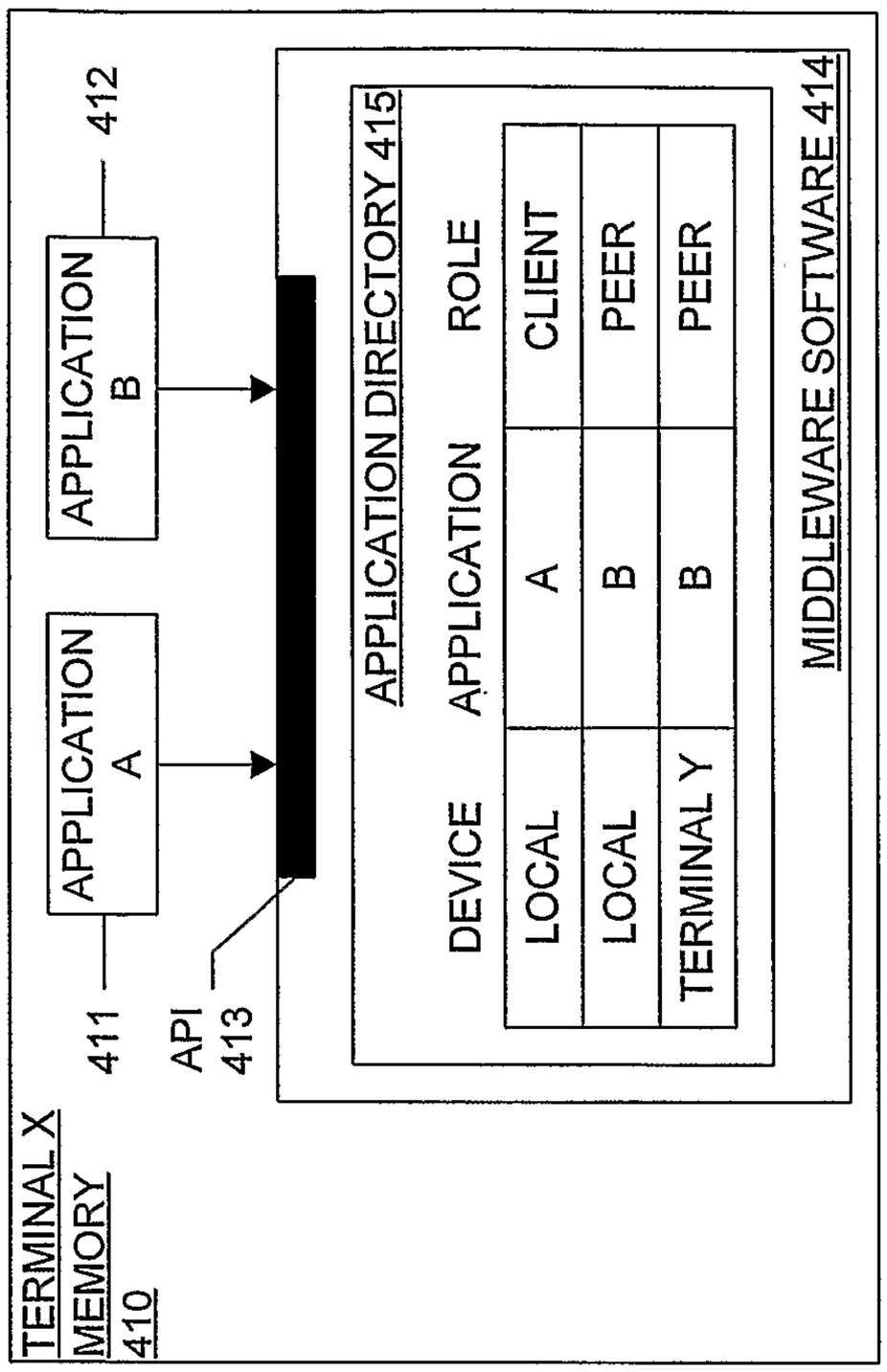

SERVER S
MEMORY
400
API
402
APPLICATION C
401
MIDDLEWARE SOFTWARE 403
APPLICATION DIRECTORY 404
DEVICE
APPLICATION
ROLE
LOCAL
C
SERVER

Figure 4A


TERMINAL X
MEMORY
410
APPLICATION A
411
API
413
APPLICATION B
412
MIDDLEWARE SOFTWARE 414
APPLICATION DIRECTORY 415
DEVICE
APPLICATION
ROLE
LOCAL
A
CLIENT
LOCAL
B
PEER
TERMINAL Y
B
PEER
TERMINAL Y
MEMORY
420
APPLICATION B
421
API
423
APPLICATION C
422
MIDDLEWARE SOFTWARE 424
APPLICATION DIRECTORY 425
DEVICE
APPLICATION
ROLE
LOCAL
B
PEER
LOCAL
C
CLIENT
TERMINAL X
B
PEER
SERVER S
C
SERVER

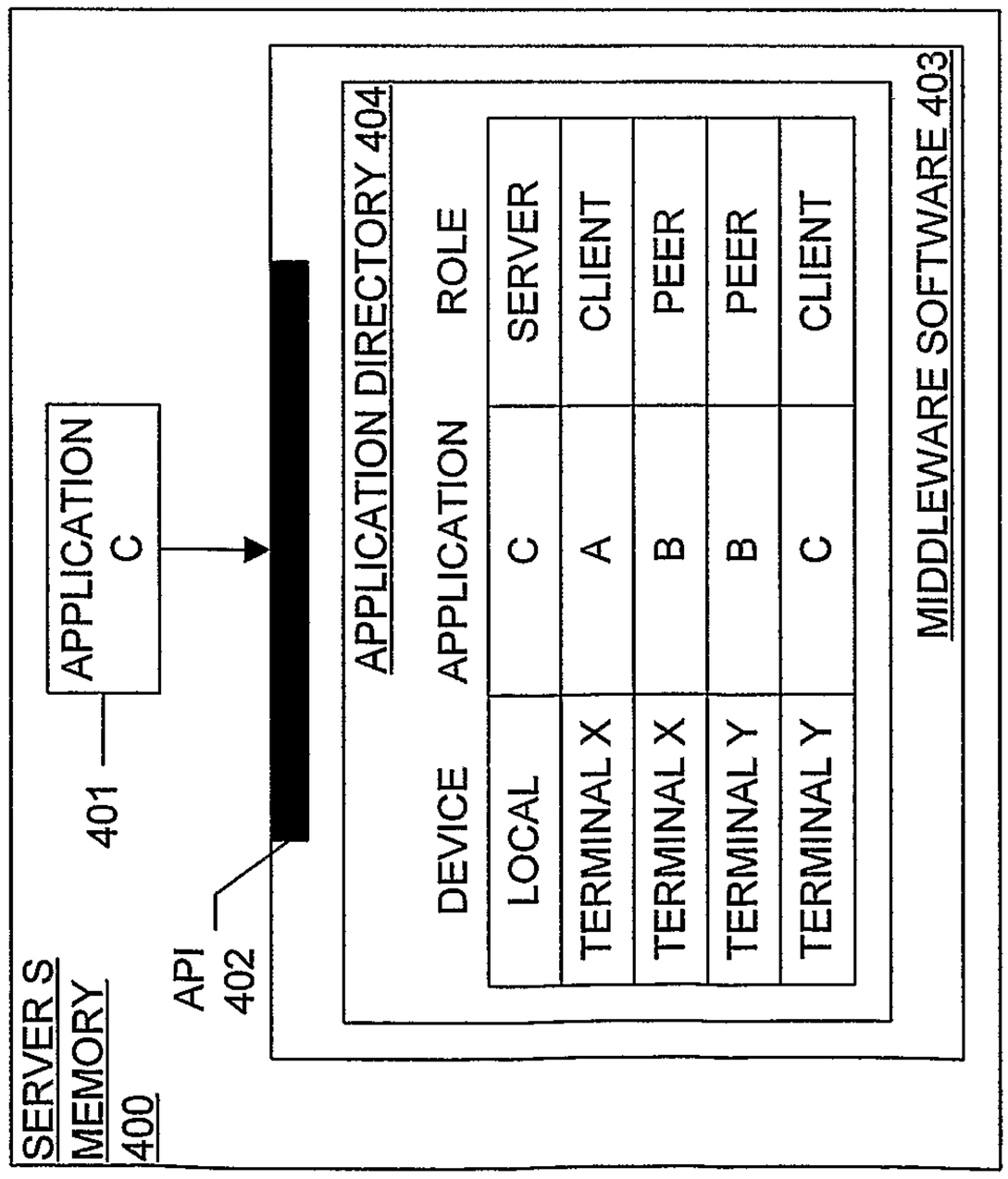
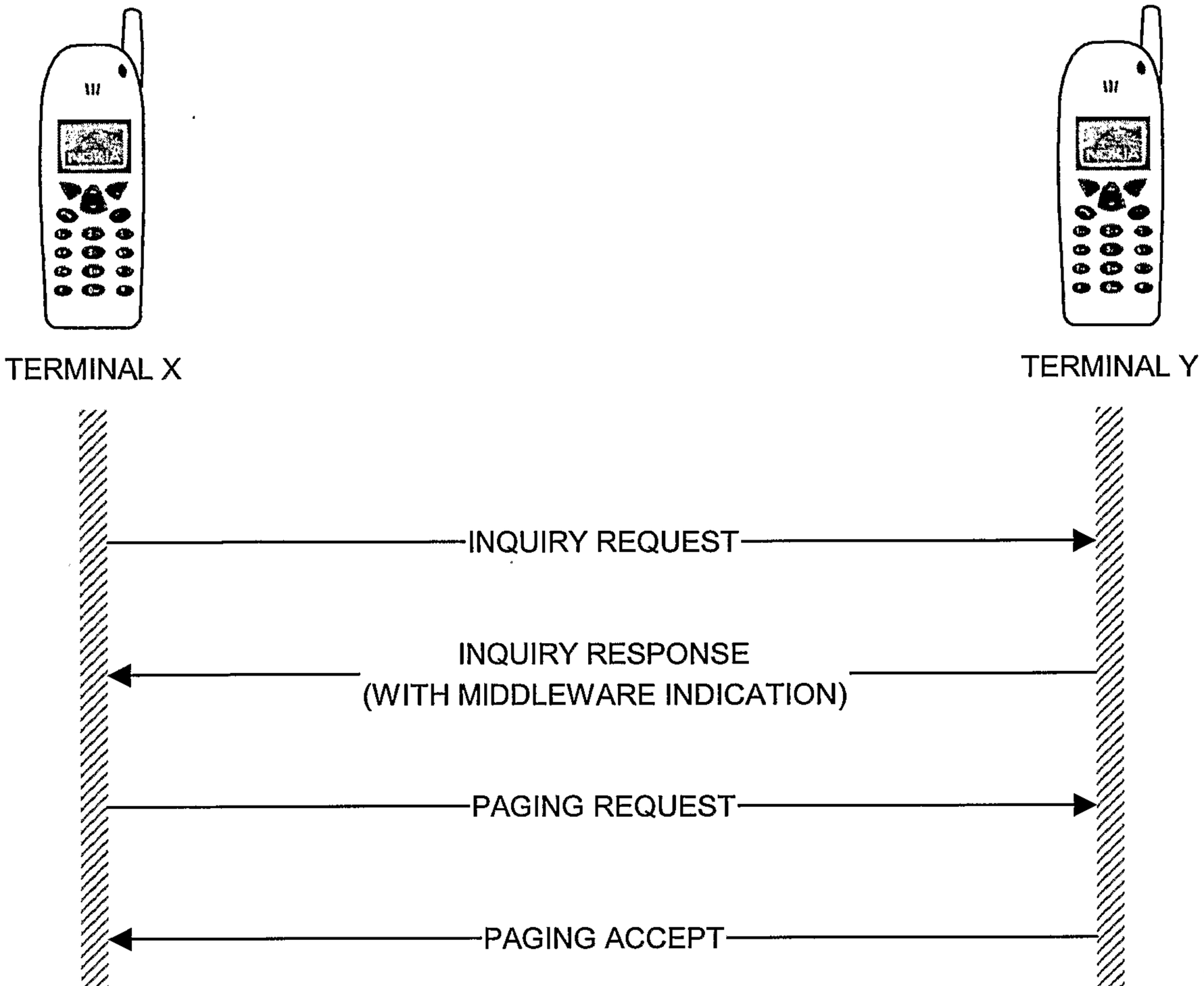
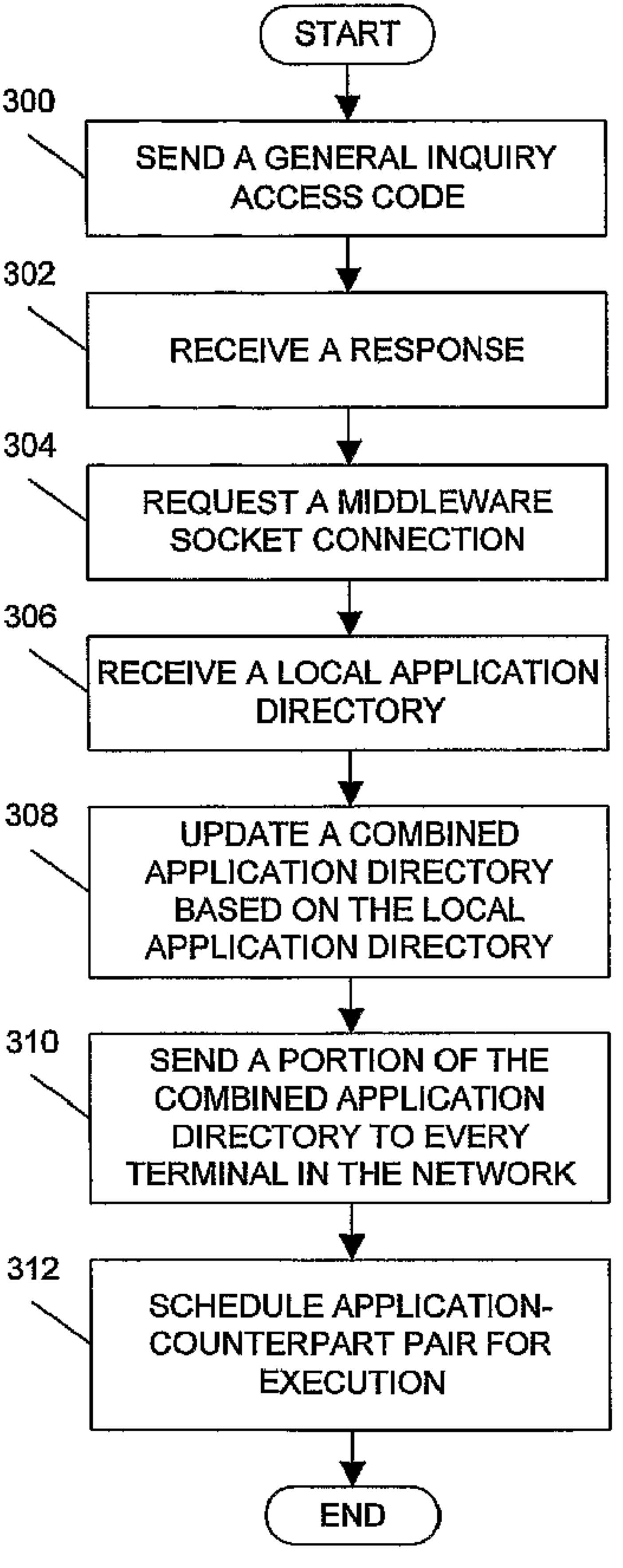

SERVER S
MEMORY
400
APPLICATION C
401
API
402
MIDDLEWARE SOFTWARE 403
APPLICATION DIRECTORY 404
DEVICE
APPLICATION
ROLE
LOCAL
C
SERVER
TERMINAL X
A
CLIENT
TERMINAL X
B
PEER
TERMINAL Y
B
PEER
TERMINAL Y
C
CLIENT

**Figure 4B**

# Figure 5


TERMINAL X
TERMINAL Y
INQUIRY REQUEST
INQUIRY RESPONSE
(WITH MIDDLEWARE INDICATION)
PAGING REQUEST
PAGING ACCEPT
RECOGNITION REQUEST
RECOGNITION RESPONSE
(WITH MIDDLEWARE CONFIRMATION)
APPLICATION AND SERVICE DISCOVERY
USING MIDDLEWARE
APPLICATION AND SERVICE EXECUTION
USING MIDDLEWARE


START
300
SEND A GENERAL INQUIRY ACCESS CODE
302
RECEIVE A RESPONSE
304
REQUEST A MIDDLEWARE SOCKET CONNECTION
306
RECEIVE A LOCAL APPLICATION DIRECTORY
308
UPDATE A COMBINED APPLICATION DIRECTORY BASED ON THE LOCAL APPLICATION DIRECTORY
310
SEND A PORTION OF THE COMBINED APPLICATION DIRECTORY TO EVERY TERMINAL IN THE NETWORK
312
SCHEDULE APPLICATION-COUNTERPART PAIR FOR EXECUTION
END