

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 November 2002 (21.11.2002)

PCT

(10) International Publication Number
WO 02/093289 A2

(51) International Patent Classification⁷: **G06F**

(21) International Application Number: PCT/IB02/01585

(22) International Filing Date: 8 May 2002 (08.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/854,619 15 May 2001 (15.05.2001) US
10/078,353 21 February 2002 (21.02.2002) US

(71) Applicants: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI). **NOKIA INC.** [US/US]; 6000 Connection Drive, Irving, TX 75039 (US).

(72) Inventor: **NYKÄNEN, Petri**; Lehdokkikatu 3, FIN-37120 Nokia (FI).

(74) Agent: **ZURA, Peter**; c/o 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, 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, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

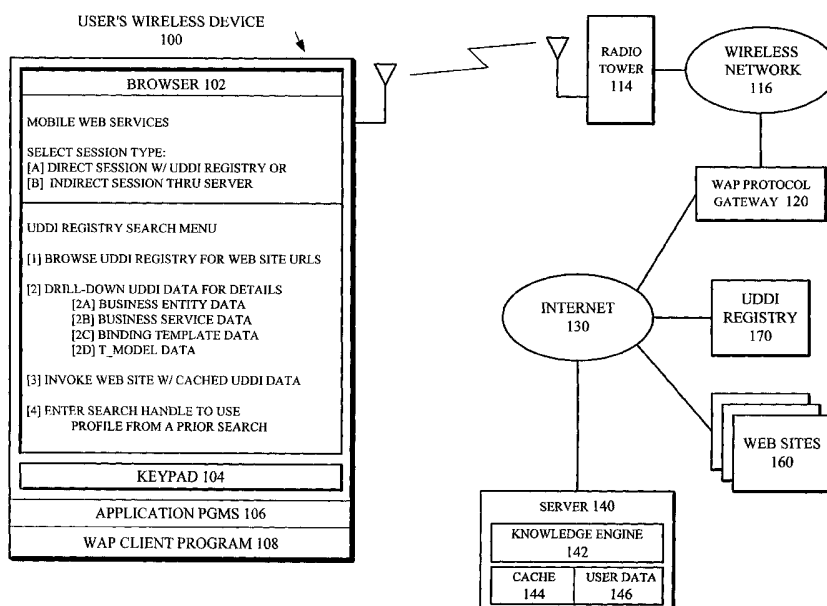
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, 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.

(54) Title: MOBILE WEB UTILIZING SERVICES



(57) Abstract: A system and method is disclosed to enable wireless device (100) to discover mobile web services by accessing UDDI registry (170) via wireless network (116) using manual and programmed instructions. User profiles stored in wireless device (100) provide shortcuts for online or offline queries. UDDI registry (170) may also be equipped with executable content to provide additional services for device (100). Alternately, server (140) is provided to execute additional programmed instructions to query UDDI registry (170) in response to commands from device (100). Server (140) can be used to cache files accessed from web sites (160), for selective forwarding to the wireless device (100).



WO 02/093289 A2

MOBILE WEB UTILIZING SERVICES

BACKGROUND OF THE INVENTION

5 This application claims priority to U.S. Application Serial No. 10/078,353, filed February 21, 2002, entitled, "Mobile Web Utilizing Services, " which is a continuation-in-part of U.S. Application 09/854,619, entitled "Mobile Web Services", filed May 15, 2001, both of which are incorporated herein by reference.

Field of the Invention:

The invention disclosed broadly relates to methods for providing Internet services and more particularly relates to improvements in mobile device accessing of Internet and other network services.

Background Art:

15 Universal Description, Discovery and Integration (UDDI) is a de facto standard for an Internet-based registry. The UDDI registry enables users to discover services and businesses on the Internet. The UDDI standard takes advantage of World Wide Web Consortium (W3C) standards and Internet Engineering Task Force (IETF) standards such as Extensible Markup Language (XML), Hypertext Transfer Protocol (HTTP), Domain Name System (DNS)
20 protocol, and Simple Object Access Protocol (SOAP) messaging protocol. The UDDI registry enables users to quickly, easily and dynamically find businesses and services on the Internet. The UDDI registry enables businesses to reach their customers and partners with information about their products and Web services. The UDDI registry also enables businesses to integrate into each other's systems and processes. Registering enables a
25 business to publicly list basic information about its company and offerings. There is also the option to list a catalog of products, services and guidelines for engagement. Registered businesses and their catalogs of services and products are then accessible in searches by potential buyers. Details of the UDDI registry and its searching protocol can be found in the following online papers:

UDDI Executive White Paper, September 6, 2000,
http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf

UDDI Technical White Paper, September 6, 2000,
http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

5 UDDI Programmer's API 1.0, UDDI Open Draft Specification 30 September 2000,
by Toufic Boubez, et al., <http://www.uddi.org/pubs/ProgrammersAPI-V1-1.pdf>

UDDI Data Structure Reference, UDDI Open Draft Specification 30 September 2000,
by Toufic Boubez, et al., <http://www.uddi.org/pubs/DataStructure-V1.pdf>

Mobile phones and wireless personal digital assistants (PDAs) are able to access the
10 Internet using the Wireless Application Protocol (WAP). As an example, the Nokia WAP
Client Version 2.0 is a software product containing the components necessary to implement a
WAP client on a wireless device. These components include a Wireless Markup Language
(WML) Browser, WMLScript engine, Push Subsystem, and Wireless Protocol Stack. The
Nokia WAP Client is a source-code product that can port and integrate into wireless devices
15 such as mobile phones and wireless PDAs. Application programs stored in the wireless
device interact with the WAP Client to implement a variety of communications applications.
Details of the Nokia WAP Client Version 2.0 can be found in the online paper: Nokia WAP
Client Version 2.0, Product Overview, Nokia Internet Communications, 2000,
www.nokia.com/corporate/wap.

20 Digital communications networks can be categorized in terms of their geographic
coverage, their transmission media, their protocols, their transmission speeds, the types of
equipment that they interconnect, and other criteria. An example of geographic coverage
categories includes wide area networks (WANs), metropolitan area networks (MANs), local
area networks (LANs), and personal area networks (PANs). An example of transmission
25 media categories includes fixed station wireline networks, mobile wireless networks, and
hybrid combinations of fixed station wireline networks communicating through wireless
access points with wireless networks. There are many digital wireless, wide area network
architectures. Most of them are connected to the public switched telephone network (PSTN)
to provide access to wireline telephones and digital computers. A short list includes Global
30 System for Mobile Communication (GSM), IS-136 TDMA-based Digital Advanced Mobile

Phone Service (DAMPS), Personal Digital Cellular (PDC), IS-95 CDMA-based cdmaOne System, General Packet Radio Service (GPRS) and broadband wireless systems such as W-CDMA, and Broadband GPRS. For more information on these digital wireless, wide area network architectures, see the book by Yi-Bing Lin, et al. entitled Wireless and Mobile
5 Network Architectures, John Wiley & Sons, 2001.

Wide area networks can include communications satellite links that interconnect nation-wide digital networks located on different continents. Nation-wide digital networks typically include backbone networks, regional distribution hubs, and routers, which interconnect access subnetworks serving local routers, servers, and service providers. The
10 Internet is a familiar example of a wide area network. For more information on the Internet as a wide area network, see the book by Daniel Minoli, et al. entitled Internet Architectures, John Wiley & Sons, 1999.

At the other end of the range for geographic coverage are short-range wireless systems. Short-range wireless systems have a typical range of one hundred meters or less.
15 They often combine with systems wired to the Internet to provide communication over long distances. The category of short-range wireless systems include both a wireless personal area network ("PAN") and a wireless local area network ("LAN"). Both of these networks have the common feature of operating in unlicensed portions of the radio spectrum, usually either in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band or the 5 GHz Unlicensed-
20 National Information Infrastructure ("U-NII") band. Wireless personal area networks use low cost, low power wireless devices that have a typical range of ten meters. The best-known example of wireless personal area network technology is the Bluetooth Standard, which operates in the 2.4 GHz ISM band. It provides a peak air link speed of one Mbps and a power consumption low enough for use in personal, portable electronics such as PDAs and
25 mobile phones. Wireless local area networks generally operate at higher peak speeds of from 10 to 100 Mbps and have a longer range, which requires greater power consumption. Wireless local area networks are typically used as wireless links from portable laptop computers to a wired LAN, via an access point (AP). Examples of wireless local area network technology include the IEEE 802.11 Wireless LAN Standard and the HIPERLAN

Standard, which operates in the 5 GHz U-NII band. For more information on wireless LANs, see the book by Jim Geier entitled Wireless LANs, Macmillan Technical Publishing, 1999.

An ad hoc network is a short range wireless system composed primarily of mobile wireless devices, which associate together for a relatively short time to carry out a common purpose. A temporary network such as this is called a "piconet" in the Bluetooth Standard, an "independent basic service set" ("IBSS") in the IEEE 802.11 Wireless LAN Standard, a "subnet" in the HIPERLAN Standard, and generally a radio cell or a "micro-cell" in other wireless LAN technologies. Ad hoc networks have the common property of being an arbitrary collection of wireless devices, which are physically close enough to be able to communicate and which are exchanging information on a regular basis. The networks can be constructed quickly and without much planning. Members of the ad hoc network join and leave as they move into and out of the range of each other. Most ad hoc networks operate over unlicensed radio frequencies at speeds of from one to fifty-four Mbps using carrier sense protocols to share the radio spectrum. The distance over which they can communicate ranges from ten meters for Bluetooth piconets to over one hundred meters for wireless LAN micro-cells in an open environment. Ad hoc networks consist primarily of mobile wireless devices, but can also include one or more access points, which are stationary wireless devices operating as a stand-alone server or connected as gateways to other networks.

What is needed is the ability of a mobile phone or wireless PDA to discover Internet businesses and services by accessing the UDDI registry. It would be even more useful to facilitate the formation of a query to the UDDI registry for the wireless device user. It would be beneficial to maintain a personal profile of the user's Internet accessing preferences and to use that profile in formulating a detailed UDDI query of the UDDI registry from the user's abbreviated inputs to the wireless device. What is also needed is a generic facility to invoke required components in a user terminal and in a server to execute services for the end user. Incorporation of pre- and post-processing applications, along with search agents, would greatly increase the flexibility of mobile web services. Terminals with limited memory capacity would further require interaction protocols that would accommodate such devices.

SUMMARY OF THE INVENTION:

In accordance with the invention, a system and method is disclosed to enable a mobile phone or wireless PDA to discover Internet businesses and services by accessing the Universal Description, Discovery and Integration (UDDI) registry. The method facilitates the formation of a query to the UDDI registry for the wireless device user. The method
5 constructs a personal user profile of the user's UDDI searching strategies and Internet accessing preferences. The user profile can be used as a shortcut for online or offline queries to the UDDI registry or for accessing pages from web sites, in response to the user's entry of abbreviated inputs to the wireless device. The method is embodied as programmed instructions which may be executed within the user's wireless device to query the UDDI
10 registry. Alternately, method is embodied as programmed instructions which may be executed within a separate knowledge engine server to query the UDDI registry in response to commands from the user's wireless device. The server can be used to cache files accessed from web sites, for selective forwarding to the user's wireless device. Additional programmed instructions are acquired by the device and executed within the device to
15 communicate with mobile web services that are made available through business-end servers.

In one aspect of the invention, a terminal configured with advanced search functionality contacts a service provider that has stored history information of a users. A knowledge engine of the service provider performs pre-processing of queries sent by a terminal, or may provide back the pre-processed query to the terminal for enabling a direct
20 request from the terminal to a preferred search engine or system. Post-processing capabilities for searches received from the knowledge engine may also be included in the terminal. The knowledge engine may further act as a search agent for the user and make queries on behalf of the terminal. When search results are received back to the knowledge engine or terminal, the search results may be filtered or prioritized and sorted based on user profile information
25 available to the knowledge engine. Additional components may be added to the search to enable branding of the search functionality.

In another aspect of the invention, an invocation server may be implemented in a web services site to facilitate mobile web applications. Service logic can be set up in the server to serve as an intermediary between mobile web services and a server. Mobile web services are
30 set up to be customized for each web services site, and can be securely downloaded by

terminals by invoking the service *get_access* to the service logic in the site. The mobile web services may be partitioned and invoked separately as needed.

The sequence of operational steps for the wireless device's UDDI registry browsing method begins by entering a search handle that will be associated with the user's search
5 strategy. Then the query terms are entered by the user, which may be a business name, part of a business name, a service description, or other characterization. If the characterization is part of a business name, the wireless device then sends a *find_business* XML inquiry using Simple Object Access Protocol (SOAP) to the UDDI registry. The wireless device then receives back from the UDDI registry, a *businessList* message that contains a list of business
10 names satisfying the *find_business* query. The user can then select an item from the returned *businessList* message and drill-down in the selected business' entity data.

The wireless device then sends a *find_service* XML inquiry using SOAP to the UDDI registry. The wireless device then receives back from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business. The user
15 can then select an item from the returned *serviceList* message and drill-down in the selected service data.

The wireless device then sends a *_get_serviceDetail_* XML inquiry using SOAP to the UDDI registry. The wireless device then receives back from the UDDI registry, a *serviceDetail* message that includes *bindingTemplate* data that contains the details of the
20 selected service. Included in the *bindingTemplate* data is the *accessPoint URL*, which is the URL of the selected service on the web site of the selected business. The service details may be comprehensive or abbreviated, and may also be prioritized according to the needs of the user (or not prioritized at all).

The wireless device then displays the *accessPoint URL* to the user. The wireless
25 device also stores the search handle in user profile with the selected *accessPoint URL*, to enable the user to access web pages from the web site of the selected business. This provides the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

The wireless device also stores the search handle in user profile with the UDDI
30 registry search strategy. The stored search strategy includes the business name query , the

selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint* URL. This provides the user with a shortcut for online or offline queries to the UDDI registry, in response to the user's entry of abbreviated search handle to the wireless device. The search handle may be associated with an icon on the user's screen. Thus, activation of the icon by the user would initiate the shortcut.

To replay a UDDI registry search strategy, the user merely enters a search handle into the wireless device and selects the replay option. The wireless device then accesses the UDDI registry search strategy from user profile corresponding to the search handle and loads the business name query, the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the UDDI registry has been updated since the user's last query, the *bindingTemplate* data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business.

The sequence of operational steps described above for the wireless device's UDDI registry browsing method can also be carried out in a separate knowledge engine server. The server performs the above described method to query the UDDI registry in response to commands from the user's wireless device. The knowledge engine in the server begins by receiving user's query or search handle. If a search handle has been received, then the server replays a corresponding search strategy for the UDDI registry accessed from the user's profile and uses the query steps specified in the strategy instead of requesting further inputs from the user.

If, however, the knowledge engine server has received a new user query, the server then accesses the UDDI registry using the method described above to identify web sites. The server returns a list of web sites to the user and stores binding templates in the user's profile. The server then receives the user's selection of web sites to search and the server updates user profile with these preferences.

Whether the server begins by receiving the user's new query or the user's search handle, the server proceeds to search the identified web sites using the URLs contained in the stored binding templates. The server retrieves any documents resulting from the search of the specified web sites. The server applies a filter that is prescribed by the user and stored in

the user's profile, to limit the returned documents to only those of particular interest to the user. The server sorts the documents in a list having an order established in accordance with user's profile. The server further stores the filtered documents and the sorted list in a cache for later use. The documents may subsequently be accessed selectively by the user. The server also returns the list of documents to user, and if the user is not logged on, the list will be returned to the user when he/she next logs on. It should be noted that the filter may be automatically created, based on the profile information or history information available on the interests of the user. Thus it would be possible, for example, for a server to determine interests of a user through earlier selections that were made.

The server receives the user's selections from the list and it updates the user's profile with the user's preferences. The server then returns the selected documents to user. As was described above, the knowledge engine server associates the search handle with user's selections and with the user's search strategy, storing that association in user's profile. It is understood that the term "document" under the present invention relates not just to web page documents, but also to services such as streaming video, audio or other application-specific communication. Thus, the present disclosure is not limited to just browsing after user discovery has taken place.

DESCRIPTION OF THE FIGURES:

Figure 1 is a network diagram of the invention, showing an exemplary relationship between the user's Wireless Application Protocol (WAP)-enabled portable wireless device, the WAP protocol gateway to the Internet, the knowledge engine server, the Universal Description, Discovery and Integration (UDDI) registry, and a plurality of web sites;

Figure 1A through **1H** show the user's wireless device in a progression of stages as it carries out the UDDI registry browsing method;

Figure 1I and **1J** show the user's wireless device in a progression of stages as it carries out the method of replaying a UDDI registry search strategy as a shortcut for online or offline queries to the UDDI registry;

Figure 2 is a functional block diagram of the wireless device **100**, showing its various components and programs;

Figure 2A is a flow diagram of the sequence of operational steps for the wireless device's UDDI registry browsing program;

Figure 2B is a flow diagram of the sequence of operational steps for the wireless device's program to replay the UDDI registry search strategy;

5 **Figure 3A** is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**;

10 **Figure 3B** is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the program to replay the UDDI registry search strategy;

Figure 4 is a functional block diagram of the knowledge engine server, showing the memory storing the application services software programs needed to perform the operations of the invention;

15 **Figure 4A** is a more detailed functional block diagram of the server, showing the knowledge engine;

Figure 4B is a network process flow diagram of the interaction of the wireless device, the knowledge engine server, and the UDDI registry when carrying out the UDDI registry browsing program in the server;

20 **Figure 5** illustrates an embodiment of web services wherein a user's wireless device communicates to a mobile network which in turn is communicatively coupled to an Internet/Internet network;

Figure 6A illustrates an example of advanced search and result processing;

Figure 6B illustrates another example of advanced search and result processing;

25 **Figure 6C** illustrates advanced search and result processing, wherein a terminal is not utilizing a UDDI system;

Figure 6D illustrates a process of invocation mobile web services;

Figure 7 illustrates yet another embodiment, wherein a terminal communicates through a mobile network to web service sites to access mobile web services;

30 **Figure 8A** illustrates layered interaction of web services utilizing a description/discovery services layer and a web services layer;

Figure 8B illustrates another layered interaction of web services utilizing a description/discovery services layer, a web services layer, and a mobile web services layer.

DISCUSSION OF THE PREFERRED EMBODIMENT:

5 The invention is applied to wireless telephones and wireless personal digital assistants (PDAs) implementing the Wireless Application Protocol (WAP) standard. **Figure 1** is a network diagram of an embodiment of the invention, illustrating a relationship between the user's Wireless Application Protocol (WAP)-enabled portable wireless device **100**, a WAP protocol gateway **120**, and the server **140**. The user's WAP-enabled portable wireless device
10 **100** can be a wireless mobile phone, pager, two-way radio, smartphone, personal communicator, or the like. The user's WAP-enabled portable wireless device **100** accesses a small file called a deck which is composed of several smaller pages called cards which are small enough to fit into the display area of the device's microbrowser **102**. The small size of the microbrowser **102** and the small file sizes accommodate the low memory constraints of
15 the portable wireless device **100** and the low-bandwidth constraints of a wireless network **116**. The cards are written in the Wireless Markup Language (WML) which is specifically devised for small screens and one-hand navigation without a keyboard. The WML language is scaleable from two-line text displays on the microbrowser **102** of a cellular telephone, up through large LCD screens found on smart phones and personal communicators. The cards
20 written in the WML language can include programs written in WMLScript, which is similar to JavaScript, but makes minimal demands on memory and CPU power of the device **100** because it does not contain many of the unnecessary functions found in other scripting languages.

 The Nokia WAP Client Version 2.0 is a software product containing the components
25 necessary to implement a WAP client on a wireless device. These components include a Wireless Markup Language (WML) Browser, WMLScript engine, Push Subsystem, and Wireless Protocol Stack. The Nokia WAP Client is a source-code product that can port and integrate into wireless devices such as mobile phones and wireless PDAs. Application programs stored in the wireless device interact with the WAP Client to implement a variety
30 of communications applications. Details of the Nokia WAP Client Version 2.0 can be found

in the online paper: Nokia WAP Client Version 2.0, Product Overview, Nokia Internet Communications, 2000, www.nokia.com/corporate/wap. It is also understood that the system and methods disclosed herein are applicable to other platforms as well, such as XHTML, and is not limited strictly to the WAP protocol.

5 The user's WAP-enabled portable wireless device **100** communicates with the radio tower **114** and can exchange messages for distances up to several kilometers. The types of wireless networks **116** supported by the WAP standard include Cellular Digital Packet Data (CDPD), Code-Division Multiple Access (CDMA), Global System for Mobile Communications (GSM), Time Division Multiple Access (TDMA), GPRS, 3G-Broadband,
10 and the like.

 The overall process of communication between the user's WAP-enabled wireless device (the client) **100**, through the WAP protocol gateway **120**, to the server **140** resembles the way Web pages are served on the Internet using the HyperText Transfer Protocol (HTTP) or World Wide Web protocol:

15 [1] The user presses a key on the user's device **100** related to the Uniform Resource Locator (URL) of the server **140**.

 [2] The user's device **100** sends the URL, via the radio tower **114** and the wireless network **116**, to the gateway **120** using WAP protocols.

 [3] The gateway **120** translates the WAP request into an HTTP request and sends it
20 over the Internet **130** to the server **140**, via Transmission Control Protocol/ Internet Protocol (TCP/IP) interfaces.

 [4] The server **140** handles the request just like any other HTTP request received over the Internet. The server **140** either returns a WML deck or a HyperText Markup Language (HTML) page back to the gateway **120** using standard server programs written, for example
25 in Common Gateway Interface (CGI) programs, Java servlets, or the like.

 [5] The gateway **120** receives the response from the server **140** on behalf of the user's device **100**. If the response is an HTML page, it gets transcoded into WML if necessary. Then the WML and WMLScript coding is encoded into a byte code that is then sent to the user's device **100**.

[6] The user's device **100** receives the response in the WML byte code and displays the first card in the deck on the microbrowser **102** to the user.

In **Figure 1**, the protocol gateway **120** includes a WAP protocol stack organized into five different layers (not shown). An application layer is the wireless application environment, which executes portable applications and services. A session layer is the wireless session protocol, which supplies methods for the organized exchange of content between client/server applications. A transaction layer is the wireless transaction protocol, which provides methods for performing reliable transactions. A security layer is the wireless transport layer security, which provides authentication, privacy, and secure connections between applications. The transport layer is the wireless datagram protocol, which shelters the upper layers from the unique requirements of the diverse wireless network protocols, such as CDPD, CDMA, GSM, etc. Additional information about the WAP standard and the WAP protocol stack can be found in the book by Charles Arehart, et al. entitled, "Professional WAP", published by Wrox Press Ltd., 2000 (ISBN 1-861004-04-1).

In **Figure 1**, the user's portable wireless device **100** includes the microbrowser **102** displaying the Mobile Web Services menu, that enables the user to navigate through the cards being displayed and to select options that are programmed by the application programs **106**. The user's device **100** also includes the WAP client program **108** which has been previously discussed.

The Mobile Web Services menu displayed by the microbrowser **102** in **Figure 1** is rendered by the WAP client program **108** under the control of the application programs **106**, which are further illustrated in **Figures 2, 2A, and 2B**. The user can select the session type utilizing the Mobile Web Services menu, by either [A] a direct session with the UDDI registry or [B] an indirect session with the UDDI registry through a knowledge server **140**. The direct session with the UDDI registry is further illustrated in the network process flow diagrams of **Figures 3A and 3B**. The knowledge service adds value to UDDI searches with pre and post-processing, regardless of the device being used to access UDDI. The indirect session with the UDDI registry through the knowledge server **140** is further illustrated in the network process flow diagram of **Figure 4B**. Whichever session type is selected by the user,

the Mobile Web Services menu of **Figure 1** presents to the user the UDDI Registry Search Menu from which the user can select the following options:

- 5 [1] BROWSE UDDI REGISTRY FOR WEB SITE URLS
- [2] DRILL-DOWN UDDI DATA FOR DETAILS
 - [2A] BUSINESS ENTITY DATA
 - [2B] BUSINESS SERVICE DATA
 - 10 [2C] BINDING TEMPLATE DATA
 - [2D] T_MODEL DATA
- [3] INVOKE WEB SITE W/ CACHED UDDI DATA
- 15 [4] ENTER SEARCH HANDLE TO USE
PROFILE FROM A PRIOR SEARCH

Option [1] of browsing the UDDI registry for web site URLs allows the user to explore and examine data organized by the UDDI registry in a hierarchy. The core information model used by the UDDI registries is defined in an XML schema. XML allows hierarchical relationships to be described for four types data: business information; service information, binding information; and information about specifications for services.

A first type of data is Business information, which is specified in the UDDI registry with the *businessEntity* XML element. The *businessEntity* XML element typically includes the name, general description, contacts, and categories of the business whose web site is on the Web. The *businessEntity* XML element serves as the top of the information hierarchy for all of the information about a business under the present embodiment.

A second type of data is Service information, which is specified in the UDDI registry with the *businessService* XML element. One or more *businessService* XML elements is contained in each *businessEntity* XML element. The *businessService* XML element includes one or more *bindingTemplate* XML elements, which are a third type of data. The *businessService* XML element is a descriptive container that is used to group a series of related Web services related to either a business process or category of services, such as purchasing services, shipping services, news services, and other high-level business

processes. The *businessService* XML element information sets can each be further categorized in combinations of industry, product and service or geographic subjects.

The *bindingTemplate* XML element contains the detailed technical descriptions of Web services. As such, these structures provide the technical entry point URL for specific services and products offered by a business. Each *bindingTemplate* XML element structure has a single logical *businessService* XML element parent, which in turn has a single logical *businessEntity* XML element parent. An important piece of information in the *bindingTemplate* XML element is the *accessPoint* element. The *accessPoint* element is the URL of a specific service provided by the business. A single attribute is typically provided, and is identified in the present embodiment as *URLType*. The purpose of the *URLType* attribute is to facilitate searching for entry points associated with a particular type of service. An example might be a purchase order service that provides three entry points, one for HTTP, one for SMTP, and one for FAX ordering. In this example, a *businessService* XML element contains three *bindingTemplate* XML element entries, each with identical data with the exception of the *accessPoint* value and *URLType* value.

A fourth type of data in the UDDI registry is the *tModel* XML element, which is pointed to by a pointer in the *bindingTemplate* XML element. The *tModel* XML element specifies the protocols, interchange formats and interchange sequencing rules for accessing web pages from the business' server having the service information specified in the *businessService* XML element.

Option [1] of the Mobile Web Services menu of **Figure 1**, is:

[1] BROWSE UDDI REGISTRY FOR WEB SITE URLS

Option [1] of browsing the UDDI registry for web site URLs involves starting with some broad information, performing a search, finding general result sets and then selecting more specific information for drill-down. The UDDI registry accommodates the browse pattern with the *find_xx* API call. These calls form the search capabilities provided by the UDDI registry and are matched with return messages that return overview information to

match the supplied search criteria. A typical browse sequence may involve finding whether a particular business has any information registered in the UDDI registry. This sequence would start with a call *find_business*, and may subsequently pass the first few characters of the businesses name. The UDDI registry would then return a *businessList* result. The

5 *businessList* result is a list of overview information (keys, names and description) of the *businessEntity* information that matched the search results returned by the *find_business* query. **Figure 1A** through **1H** illustrate the user's wireless device in a progression of stages as it carries out the UDDI registry browsing method. **Figure 2A** illustrates a flow diagram of the sequence of operational steps for the wireless device's UDDI registry browsing program.

10 **Figure 3A** illustrates a network process flow diagram, showing the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**.

Option [2] of the Mobile Web Services menu of **Figure 1**, is:

- 15 [2] DRILL-DOWN UDDI DATA FOR DETAILS
 [2A] BUSINESS ENTITY DATA
 [2B] BUSINESS SERVICE DATA
 [2C] BINDING TEMPLATE DATA
 [2D] T_MODEL DATA
- 20

After the user has identified a business he/she has been browsing for in Option [1], the user can drill-down into their *businessService* information. Here, the user may search for particular service types (e.g. purchasing, shipping, news and the like) using the *find_service* API call. Similarly, if the user knows the technical fingerprint (*tModel* signature) of a

25 particular product and wants to see if the business he/she has chosen supports a compatible service interface, the user can use the *find-binding* API call.

Once the user has a key for one of the four main data types managed by the UDDI registry, he/she can use that key to access the full registered details for a specific data type (*businessEntity*, *businessService*, *bindingTemplate*, or *tModel*). The full registered

30 information for any of these structures can be accessed by passing a relevant key type to one of the *get_xx* API calls to the UDDI registry.

Continuing with the example on browsing, one of the data items returned by all of the *find-xx* return sets is key information. In the case of a business that the user is interested in, the *businessKey* value returned within the contents of a *businessList* structure can be passed as an argument to the UDDI registry to *get_businessDetail*. The successful return of this
5 message is a *businessDetail* message containing the full registered information for the entity whose key value was passed. This typically will be a full *businessEntity* structure. Since full *businessEntity* structures can contain a large quantity of information, it can be optionally cached in the cache 144 of the knowledge engine server 140 of **Figure 1**.

10 Option [3] of the Mobile Web Services menu of **Figure 1**, is:
 [3] INVOKE WEB SITE W/ CACHED UDDI DATA

After the user has retrieved the *businessEntity* structure from the UDDI registry 170
of **Figure 1**, the user can access the *accessPoint* element of the *bindingTemplate* XML
15 element in the *businessEntity* structure to obtain the URL of a specific service provided by the business. Option [3] invokes the business' web site 160 with the cached URL to access the desired web pages. Since the accessed web pages from the web site 160 can contain a large quantity of information, it can be optionally cached in the cache 144 of the knowledge engine server 140 of **Figure 1**. Typically, if *bindingTemplate* is used to attempt contact
20 services directly at the *accessPoint* and fails, the terminal may use the *bindingTemplate*'s unique ID to fetch a new *bindingTemplate* instance from the UDDI registry, assuming that the new instance contains up-to-date information on the service.

Option [4] of the Mobile Web Services menu of **Figure 1**, is:
25 [4] ENTER SEARCH HANDLE TO USE PROFILE FROM A PRIOR SEARCH

Option [4] enables the user to replay a prior UDDI registry search strategy. The user typically enters a previously used search handle into the wireless device and selects the

replay Option [4]. The wireless device then accesses the UDDI registry search strategy from the user's stored profile corresponding to the search handle and loads the business name query, the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the UDDI registry has been updated since the user's last query, the bindingTemplate data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business. **Figure 1I** and **1J** illustrate the user's wireless device in a progression of stages as it carries out the method of replaying a UDDI registry search strategy as a shortcut for online or offline queries to the UDDI registry.

Figure 2B discloses a flow diagram of the sequence of operational steps for the wireless device's program to replay the UDDI registry search strategy. **Figure 3B** is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the program to replay the UDDI registry search strategy.

The sequence of operational steps for the wireless device's UDDI registry browsing method are shown in **Figure 2A**. The Mobile Web Services menu of **Figure 1A** begins by entering a search handle that will be associated with the user's search strategy. The query terms are subsequently entered by the user, which may be a business name, part of a business name, a service description, or other characterization. If the characterization is part of a business name, the wireless device then sends a *find_business* XML inquiry using Simple Object Access Protocol (SOAP) to the UDDI registry. The wireless device then receives back from the UDDI registry, a *businessList* message shown in the Mobile Web Services menu of **Figure 1B**, that contains a list of business names satisfying the *find_business* query. The user can then select an item from the returned *businessList* message and drill-down in the selected business' entity data.

The Mobile Web Services menu of **Figure 1C** shows the wireless device then sending a *find_service* XML inquiry using SOAP to the UDDI registry. The Mobile Web Services menu of **Figure 1D** shows the wireless device then receiving back from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business. The user can then select an item from the returned *serviceList* message and drill-down in the selected service data, as shown by the Mobile Web Services menu of **Figure 1E**.

The wireless device then sends a *_get_serviceDetail_XML* inquiry using SOAP to the UDDI registry, as shown by the Mobile Web Services menu of **Figure 1E**. The Mobile Web Services menu of **Figure 1F** then shows the wireless device receiving back from the UDDI registry a *serviceDetail* message that includes bindingTemplate data that contains the
5 details of the selected service. Included in the *bindingTemplate* data is the *accessPoint URL*, which is the URL of the selected service on the web site of the selected business.

The Mobile Web Services menu of **Figure 1G** shows the wireless device displaying the accessPoint URL to the user. The Mobile Web Services menu of **Figure 1H** shows that the *serviceDetail* message can contain one or a plurality of URLs, depending on the number
10 of matches made against the user's query in the search by the UDDI registry. The wireless device also stores the search handle in user profile with the selected accessPoint URL, to enable the user to access web pages from the web site of the selected business. This provides the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

15 The wireless device also stores the search handle in user profile with the UDDI registry search strategy. The stored search strategy includes the business name query, the selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint URL*. This provides the user with a shortcut for online or offline queries to the UDDI registry, in response to the user's entry of abbreviated search
20 handle to the wireless device.

To replay a UDDI registry search strategy, the user enters a search handle into the wireless device and selects the replay option, as shown in the Mobile Web Services menu of **Figure 1I**. The wireless device then accesses the UDDI registry search strategy from user profile corresponding to the search handle and loads the business name query, the selected
25 *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the UDDI registry has been updated since the user's last query, the bindingTemplate data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business, as shown by the Mobile Web Services menu of **Figure 1J**.

Figure 2 is a functional block diagram of the wireless device **100**, and illustrates its various components and programs. **Figure 2** discloses the memory **202** connected by means of the bus **204** to the radio **206**, the keypad **104**, the central processor **210** and the display **212**. The memory **202** contains program modules each consisting of a sequence of

5 programmable instructions. The wireless devices UDDI registry browsing program **240** (see **Figure 2A**) is stored in the memory **202**. The wireless devices replay UDDI registry search strategy program **270** (see **Figure 2B**) is also stored in the memory **202**. The indirect session thru server program **220** is also stored in the memory **202**.

User data **222** is stored in the memory **202**, which includes the user ID **230** and the

10 user profile **232**. The user profile **232** includes the user's name and email address, the user's search handles, the UDDI search strategies, the sorting and filtering specifications, selected URLs, selected document titles and binding templates which contain URLs. Also contained in the memory **202** of **Figure 2** is the cache **224** wherein documents and lists returned from a website, can be stored. In addition, the WAP client program **108** is stored in the memory

15 **202**.

Figure 2a is a flow diagram disclosing the sequence of operational steps for the wireless device's UDDI registry browsing program **240**. The steps depicted in **Figure 2A** are as follows:

- 20 **Step 250:** WIRELESS DEVICE BROWSING UDDI REGISTRY ENTER SEARCH HANDLE "_STOCKS_"
- Step 252:** ENTER QUERY TERMS "_WALL STREET_" + "_FINANC*_" IS THIS A BUSINESS NAME? Y/N "_Y_" IS THIS A SERVICE NAME? Y/N "_N_"
- 25 **Step 254:** SEND "_find_business_" XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY
- 30 **Step 256:** SELECT ITEM FROM RETURNED businessList MESSAGE DRILL-DOWN BUSINESS ENTITY DATA "_WALL STREET JOURNAL ONLINE_"
- 35 **Step 258:** SEND "_find_service_" XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY

- Step 260:** SELECT ITEM FROM RETURNED serviceList MESSAGE DRILL-DOWN BUSINESS SERVICE DATA "_TECH STOCK NEWS_"
- 5 **Step 262:** SEND *_get_serviceDetail_* XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY
- 10 **Step 264:** SELECT ITEM OF RETURNED serviceDetail MESSAGE DISPLAY accessPoint URL FROM bindingTemplate DATA "COMPUTERS" IN RETURNED serviceDetail MESSAGE URL =
 "_www.wsj.com/news/computers_"
- 15 **Step 266:** STORE SEARCH HANDLE "_STOCKS_" IN USER PROFILE WITH SELECTED URL = "_www.wsj.com/news/computers_"
- 20 **Step 268:** STORE SEARCH HANDLE "_STOCKS_" IN USER PROFILE WITH UDDI REGISTRY SEARCH STRATEGY: BUSINESS NAME QUERY
 "_WALL STREET_" + "_FINANC*" businessEntity DATA
 SELECTED "_WALL STREET JOURNAL ONLINE_" businessService
 DATA SELECTED "_TECH STOCK NEWS_" bindingTemplate DATA
 SELECTED "_COMPUTERS_" accessPoint URL SELECTED
 "_www.wsj.com/news/computers_"

Figure 2B shows a flow diagram of the sequence of operational steps for the wireless
 25 device's program to replay the UDDI registry search strategy. **Figure 2B** includes the following steps:

- Step 271:** REPLAY UDDI REGISTRY SEARCH STRATEGY ENTER SEARCH HANDLE "_STOCKS_"
- 30 **Step 272:** ACCESS UDDI REGISTRY SEARCH STRATEGY IN USER PROFILE FOR SEARCH HANDLE "_STOCKS_", BUSINESS NAME QUERY
 "_WALL STREET_" + "_FINANC_", businessEntity DATA SELECTED
 "_WALL STREET JOURNAL ONLINE_", businessService DATA
 SELECTED "_TECH STOCK NEWS_", bindingTemplate DATA
 SELECTED "_COMPUTERS_", accessPoint URL SELECTED
 "_www.wsj.com/news/computers_"
- 35 **Step 274:** SEND *_find_business_* SML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.
- 40 **Step 276:** SELECT ITEM FROM RETURNED businessList MESSAGE DRILL-DOWN BUSINESS ENTITY DATA "_WALL STREET JOURNAL
 ONLINE_"

Step 278: SEND *_find_service_* XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.

5 **Step 280:** SELECT ITEM FROM RETURNED *serviceList* MESSAGE DRILL-DOWN BUSINESS SERVICE DATA “_TECH STOCK NEWS_”.

Step 282: SEND *_get_serviceDetail_* XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.

10

Step 284: SELECT ITEM OF RETURNED *serviceDetail* MESSAGE DISPLAY accesPoint URLs FROM bindingTemplate DATA “COMPUTERS” IN RETURNED *serviceDetail* MESSAGE, URL =

15

“_www.wsj.com/news/computers_”, URL =
 “_www.wsj.com/quotes/computers_”, URL =
 “_www.wsj.com/graphs/computers_”.

Figure 3A discloses a network process flow diagram showing the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**. The network process flow diagram in **Figure 3A** consists of three columns labeled respectively, wireless device **100**, server **140** and UDDI registry **170**. The process flow diagram **Figure 3A** illustrates the interaction between steps carried in the wireless device **100** and steps carried out in the UDDI registry **170**. The diagram of **Figure 3A** begins with the wireless device **100** step **250** where a search handle is entered. Then, at step **252**, query terms are entered. At step **254**, the *_find_business_* query is sent to the UDDI registry **170**. In the UDDI registry column **170** in **Figure 3A**, the UDDI registry conducts searches based on the *_find_business_* query and returns the *businessList* in step **255**. The flow then returns to the wireless device **100** and passes to step **256** wherein an item of the *businessList* is selected which typically is a *businessEntity*. The *businessEntity* is then drilled down. The flow then passes to step **258** in which the *_find_service_* query is sent to the UDDI registry. The flow then passes to the UDDI registry **170** where the *_find_service_* query is searched and the service list is returned in step **259**. Then the flow passes to the wireless device **100** where an item of the *serviceList* is selected which is a *businessService*. The *businessService* is subsequently drilled down in step **260**. The flow proceeds to step **262** in which the *_get_serviceDetail_* query is sent to the UDDI registry. Then the flow passes to

the UDDI registry 170 where the *_get_serviceDetail_* query is responded to and the binding template is returned in step 263. Then the flow passes back to the wireless device 100 where in step 264, the *serviceDetail* is selected from the *bindingTemplate* and the accessPoint URL is stored in step 264.

5 **Figure 3B** illustrates a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out a program to replay the UDDI registry search strategy. **Figure 3B** is divided into three columns, the wireless device column 100, the server 140 column, and the UDDI registry 170 column. **Figure 3B** starts with the wireless device entering the replay UDDI registry search strategy wherein the search handle
10 is entered in step 271. Then the process flows to step 272 where the search strategy is accessed in the user profile which corresponds to the search handle which was input in step 271. **Figure 3B** then proceeds through the remainder of the processes in a similar manner as that disclosed in **Figure 3A**. **Figure 3B** discloses how the user is enabled to replay a prior UDDI registry search strategy. The user merely enters a previously used search handle into
15 the wireless device and selects the replay option. The wireless device then accesses the UDDI registry search strategy from the user's stored profile corresponding to that search handle. This may be performed either at the wireless device 100 or, in the alternate embodiment in the server 140. The search strategy includes information such as the *businessEntity* data and *businessService* data and *bindingTemplate* data that was found
20 during the course of an earlier search of the UDDI registry 170. This data contained in the UDDI registry search strategy (accessed from the stored profile) is then loaded as the data for each respective operand that would have been otherwise entered by the user. In this way, the flow diagram of **Figure 3B** enables the user to automatically invoke a search strategy of the UDDI registry 170, that was used in the past.

25 **Figure 4** is a functional block diagram of the knowledge engine server, showing the memory storing the application services software programs needed to perform the operations of an embodiment of the invention. **Figure 4** discloses the functional components of an exemplary knowledge engine server 140 arranged as an object model. The object model groups the object oriented software programs into components that perform the major
30 functions and applications in knowledge engine server 140. The object model for memory

402 of knowledge engine server 140 employs a three-tier architecture that includes presentation tier 415, infrastructure objects partition 422, and business logic tier 414. The object model further divides business logic tier 414 into two partitions, application objects partition 422 and data objects partition 426.

5 Presentation tier 415 retains the programs that manage the device interfaces to knowledge engine server 140. In **Figure 4**, presentation tier 415 includes network interface 420. A suitable implementation of presentation tier 415 may use Java servlets to interact with WAP protocol gateway 120 via the hypertext transfer protocol ("HTTP"). The Java servlets run within a request/response server that manages the exchange of messages between
10 WAP protocol gateway 120 and knowledge engine server 140. A Java servlet is a Java program that runs within a Web server environment. A Java servlet takes a request as input, parses the data, performs logic operations, and issues a response back to WAP protocol gateway 120. The Java runtime platform pools the Java servlets to simultaneously service many requests. Network interface 420 accepts request messages from WAP protocol
15 gateway 120 and passes the information in the request to visit object 428 for further processing. Visit object 428 passes the result of that processing to network interface 420 for transmission back to the WAP protocol gateway 120. Network interface 420 may also use network adapter 406 to exchange data with another user device.

 Infrastructure objects partition 422 retains the programs that perform administrative
20 and system functions on behalf of business logic tier 414. Infrastructure objects partition 422 includes operating system 425, and an object oriented software program component for database server interface 430, and system administrator interface 432.

 Business logic tier 414 in **Figure 4** includes multiple instances of visit object 428, 428', 428". A separate instance of visit object 428 exists for each network interface 420
25 session. Each visit object 428 is a stateful session bean that includes a persistent storage area from initiation through termination of the session, not just during a single interaction or method call. The persistent storage area retains information associated with the session.

 When WAP protocol gateway 120 sends a query message to knowledge engine server 140, the message is sent to network interface 420 to invoke a method that creates visit object
30 428 and stores connection information as a state in visit object 428. Visit object 428 may, in

turn, invoke a method in UDDI registry browsing application 440 to browse the UDDI registry 170. Application 440 sends queries to the UDDI registry, as discussed above.

When WAP protocol gateway 120 sends a search handle message to knowledge engine server 140, a message is sent to network interface 420 to invoke a method that creates
 5 visit object 428 and stores connection information as a state in visit object 428. Visit object 428 may, in turn, invoke a method in replay UDDI registry search strategy application 442 to replay a prior search strategy. The application 442 performs the replay method discussed above. Similar operations occur for applications 444, 446 and 448 in Figure 4.

Figure 4A is a more detailed functional block diagram of the server, showing the
 10 knowledge engine 142. The knowledge engine 142 includes a program which is shown in Figure 4A as a sequence of steps 1 through 13.

KNOWLEDGE ENGINE 142

- 15 [1] RECEIVE USER'S QUERY OR SEARCH HANDLE
- [2] IF SEARCH HANDLE RECEIVED, THEN REPLAY UDDI SEARCH
PROFILE & GOTO 6
- [3] ACCESS UDDI REGISTRY WITH USER'S QUERY TO IDENTIFY WEB
SITES
- 20 [4] RETURN LIST OF WEB SITES TO USER AND STORE BINDING
TEMPLATES IN PROFILE
- [5] RECEIVE USER'S SELECTION OF WEB SITES TO SEARCH & UPDATE
USER PROFILE
- [6] SEARCH THE IDENTIFIED WEB SITES USING URLS FROM BINDING
25 TEMPLATES
- [7] RETRIEVE DOCUMENTS RESULTING FROM SEARCH
- [8] SORT LIST OF DOCUMENTS IN ACCORDANCE WITH USER'S PROFILE
- [9] STORE DOCUMENTS AND LIST IN CACHE
- [10] RETURN LIST OF DOCUMENTS TO USER WHENEVER USER IS
30 LOGGED ON
- [11] RECEIVE USER'S SELECTIONS FROM LIST AND UPDATE USER'S
PROFILE
- [12] RETURN SELECTED DOCUMENTS TO USER
- 35 [13] ASSOCIATE SEARCH HANDLE WITH USER'S SELECTIONS IN USER'S
PROFILE

Also provided in server **140** is the user data **146** which includes the user ID profile **230** which is discussed above. Since a plurality of users may make use of the server **140**, there are a plurality of user profiles shown in **Figure 4A**, one for the user ID **230'** having user profile **232'** and another for the user ID **230''** having user profile **232''**. The server **140** of **Figure 4A** also has a cache **144** which stores documents and lists which are obtained from the various websites **160** that have been interrogated by the user with the aid of the server **140**.

Figure 4B is a flow diagram of the sequence of operational steps for the server **140** UDDI registry browsing program **170**. **Figure 4B** has three columns, the first column labeled user's wireless device **100**, the second column labeled server **140**, and a third column labeled UDDI registry **170** and web sites **160**. **Figure 4B** illustrates the interaction of the wireless device **100**, the server **140**, the UDDI registry **170** and the web sites **160**, in accordance with an embodiment of the invention. Starting with the user's wireless device **100**, **Figure 4B** shows sending a query to the server, in step **401**. Prior to sending the request from wireless device **100**, the user location included in wireless device **100**, is sent in step **400**. The query request is then transmitted in step **401** to server **140**. At the server **140**, the query is received from the user in step **402**, and the process flows to step **403** where web sites are identified from the UDDI registry and the user's profile is updated. The process in step **403** for identification of the web sites from the UDDI registry is the process which has been discussed above in connection with **Figures 2A** and **3A**. The process then flows to step **410** in the UDDI registry **170**, wherein the UDDI registry accesses the requested information in response to the queries sent from the server **140** to identify web sites. Step **410** then transfers the results of that search from the UDDI registry **170** back to the server **140**. At the server **140**, the process flows to step **404** wherein the server has taken the information identifying the web sites received from the UDDI registry **170**, and formulates a request to retrieve documents which is sent to the web sites **160**. The process then flows to step **411** where the web sites **160** receiving the request from the server **140**, access their respective servers for the requested documents and then return the documents to the server **140**. The server **140** then sorts the documents into a list in accord with the user's profile, sorting the list into the order requested by the user, and filtering out any documents which the user is not interested, in

accordance with the user's profile. The process then flows to step **405** in which the documents are stored in the cache at the server, cache **144**, and the list which has been sorted by the server **140**, is returned to the user. The process then flows to step **406** at the user's wireless device **100** where the sorted list received from the server **140** is presented to the user
5 and the user can select from that list those documents desired to be reviewed. In step **406**, the user's request for documents is then sent back to the server **140**. The process then flows to step **407** where the server **140** accesses its cache **144** to retrieve those documents selected by the user in step **406**, and then the server **140** returns the selected documents to the user's wireless device **100**. Step **407** then compiles the user's preferences in the user profile **230**.
10 The server **140** can also update the user's preferences in the user's profile in step **409**. The process flows from step **407** to step **408** at the user's wireless device **100**, where the user receives the selected documents.

Using another embodiment of the present invention, a generic facility may be used to invoke required components in the terminal and in the server to implement end services for
15 the user. Under this embodiment, terminal-end software or components are not required to be resident in the user's terminal, and can automatically be retrieved from the server prior to service activation. By locating executable components on the server side, devices with limited memory capabilities or display sizes may still utilize web services and access run-time service installation and invocation. Numerous security protocols may also be
20 incorporated to ensure that user information is protected throughout the access/execution user iterations.

Figure 5 discloses an embodiment of the web services, wherein a user's wireless device **500** communicates to mobile network **501**, which in turn is communicatively coupled to an Internet/Intranet network **502**. It should be noted that the system of **Figure 5** parallels
25 the systems disclosed in **Figures 1, 2** and **4**. Mobile network **501** is further coupled to service description & discovery service **506** and knowledge engine **505**. Knowledge engine **505** can access information from customer profile **503** in order to tailor searches and requests, described in greater detail above. Knowledge Engine **505** can additionally serve to pre-process search queries, and post-process received data. Service description & discovery
30 service **506** is further connected to web service server **507**, which has web services **504**

resident therein. Each web service has a defined protocol interface for utilizing that service. When the user device **500** uses the specified protocol, the device will initiate a web service session initiated from the Web Service Server **507** site. Description and discovery services (such as UDDI) may thus further be enhanced.

5 Additionally, a search agent (see **Figure 6B**) may be made resident within the knowledge engine **505**. The agent may be configured to conduct searches on behalf of the user, based on the information in the customer profile **503** (see also **Figure 1** and accompanying text). The knowledge engine **505** can thus support agent functions and prepare and/or execute searches while the device **500** is not in active use.

10 **Figure 6A** illustrates an example of advanced search and result processing under an embodiment of the invention. A user's device, or terminal **600**, communicates through network **601** to knowledge engine server **602** and search engine server **603**. The terminal **600**, knowledge engine server **602** and search engine server **603** are provided with platforms **615**, **621**, and **622**. Terminal **600** is provided with a plurality of software systems **604**, **605**,
15 **606**, that communicate through a search application programming interface (API) **610**, as well as other API's **611** that may appear on the terminal **600**. It is understood that the actual number of software systems in terminal **600** may vary in accordance with the capabilities of the terminal. Software system **606** is shown as having a search algorithm **607**, as well as multiple application algorithms **608**, **609**. The algorithms are configurable to communicate
20 through a variety of API's specified by the device or by the user. The algorithms in software system **606** may be duplicated, modified, increased or decreased in number in each software system (**604**, **605**) residing in the terminal **600** and may be further altered to suit the user's needs. When a search request is initiated to the search API **601**, search function **612**
25 activates request pre-processing **614**, which in turn establishes communication with search processor **616** in the knowledge engine server **602**. Once a link is established with search processor **616**, knowledge engine **617** processes the request to match a user/customer identity and profile, stored in customer profile database **618**. Knowledge engine **617** utilizes user profiles and preferences, as well as user terminal capabilities, to modify the search request and match the search with the customer profile information. The modified request is then
30 sent back to request pre-processing **614** and to search function **612**. The modified search

request is then transmitted to search engine (or UDDI operator site) **619**, located in search engine server **603**.

At this point, search engine **619** may further initiate modifications to the request in accordance with customer profile **620**, which is also located in search engine server **603**.

5 Customer profile **620** contains additional information about a user from the search engine server side, and serves to complement pre-processing functions with post-processing capabilities. As search engine **619** provides back search results, the results may be tailored to match user profiles through transmission of the results between result post-processing **613**, search processor **616**, and knowledge engine **617**. It is understood that the search API may
10 alternately be set to function without pre- or post-processing functions in mobile terminals in which the service provider does not have knowledge engine functionality.

Another embodiment of advanced search and result processing is illustrated in **Figure 6B**. A user's device, or terminal **600**, communicates through network **601** to knowledge engine server **602** and search engine server **603**. The terminal **600**, knowledge engine server
15 **602** and search engine server **603** are provided with platforms **615**, **621**, and **622**. Terminal **600** is provided with a plurality of software systems **604**, **605**, **606**, that communicate through a search API **610**, as well as other API's **611** that may appear on the terminal **600**. Once again, it is understood that the actual number of software systems in terminal **600** may vary in accordance with the capabilities of the terminal. Software system **606** is shown as
20 having search algorithm **607**, as well as multiple application algorithms **608**, **609**. The algorithms are configurable to communicate through a variety of API's specified by the device or by the user. The algorithms in software system **606** may be duplicated, modified, increased or decreased in number in each of the other software systems (**604**, **605**) residing in the terminal **600** and may be further altered to suit the user's needs.

25 As the search is initiated through the search API **610**, search function **612** is executed. Search function **612** transmits a search request to search proxy **623**, located in knowledge engine server **602**. Search proxy **623** is configured to handle pre- and post-processing features. Search proxy **623** communicates with knowledge engine **624** to initiate request pre-processing functions. Knowledge engine **624** recalls information from customer profile **618**
30 from which a search is processed in accordance with the profile (see **Figure 6A**). Once the

search has been pre-processed, search proxy 623 forwards the search request to search engine 619, located in search engine server 603. Search engine 619 is communicatively coupled to customer profile 620, wherein additional customer information is collected with the request for post-processing. When search engine 619 returns the profiled search results to search proxy 623, a post-processing function is performed on the search results to establish a user-tailored search before transmitting it back to search function 612.

Search agent 625 may also be implemented in the knowledge engine server 602. Search agent 625 is communicatively coupled with the search proxy 623 and the customer profile 618. Search requests from search function 612 that are sent to search proxy 623, wherein the search function is transmitted to the search agent 625 and stored. Additional data, such as time of search execution, type of search, user profile data, or additional search preferences may be additionally transmitted to the search agent 625. Once the search agent 625 obtains the search request and related information, search agent 625 may provide supplementary search functions, as well as conducting off-line searches.

Figure 6C illustrates another embodiment of the invention, where the user's terminal 600 is not utilizing a UDDI system. Instead, Browser 630 directly accesses WWW Server or WAP gateway 626, wherein WWW Server or WAP gateway 626 is communicatively coupled to knowledge engine 624 and/or search agent 625. WWW Server or WAP gateway 626 transmits requests to search engine 619, wherein search results are generated. The remaining function of the system in **Figure 6C** parallel those already described in **Figures 6A-B**. It is understood that the system configuration of the present invention may be spread among numerous servers, or may be condensed into a single server.

Figure 6D illustrates another embodiment of the invention, wherein the user terminal performs requests or invocations for mobile web services. The web services are temporarily downloaded to terminal 600, wherein server logic 642 in web service server 638 communicates with back office systems and business logic 648, located in remote server(s) 647. **Figures 7, 8A-B** illustrate further details of the mobile web service system. Requests are transmitted through invocation API 633, and may be subject to access restrictions in privacy control 636. Data stored in privacy profile database 634 would determine the access capabilities of a user. Once access is granted, invocation client 637 is activated to contact

invocation server **644**, shown in web service server **638**. invocation Client **637** communicates to invocation server **644** typically through a web service interface (not shown). The web service interface can be a published API or protocol that has been requested by the mobile terminal. The API/protocol is configured to support web services
5 that match the query. Web service server **638** is loaded with numerous software systems (**639-641**), wherein service logic **642** resides. Service logic **642** accesses API's **643** and transmits them to invocation server **644**.

Invocation client **637** is further enabled to provide device profile information to assist invocation server **644** in selecting suitable web services **645** for the mobile terminal **600**.
10 Invocation server **644** will typically access various web service protocols from logic storage **646**, or may dynamically create new web services for a particular instance. Invocation server **644** may invoke service logic **642** locally in relation to the selected mobile web service, or may delay the invocation until the web service is accessed from terminal **600**.

Figure 7 illustrates an embodiment of the invention, wherein terminal **700**, with
15 platform **708**, communicates through mobile network **711** to web service sites **712**. The mobile network **711** may also include a public cloud UDDI. Knowledge engine server **709**, and customer profile **710** provides additional support to UDDI functionality as described above. In order to access mobile web services, terminal **700** searches first for a web service server (from web service sites **712**) that supports the web service capabilities of terminal **700**.
20 Once an appropriate web service server has been located, the service logic in a server from the web service sites functions to serve as the intermediary between terminal **700** and back office systems and systems logic (see **Figure 6D**).

Once a mobile web service **702** is downloaded or stored in terminal **700**, the mobile web service **702** has a limited number of API's to utilize (**704-706**). This means that the web
25 services in the mobile terminal are executed in a limited "sandbox". The term "sandbox" refers to applications that are downloaded from a client or an enabled object access to operating system calls or to other resources. To compensate for this, mobile web services **702** access API's locally from the execution environment of terminal **700** (i.e., the sandbox). Enabled API's can automatically provide user data and profile information.

The manner of providing API's, as well as the actual number of API's, may further be manipulated to serve the user's needs and minimize the complexities of data entry and transmission. The API's can thus create web services for mobile domains that are user-friendly. Once a user has entered information into a device, the information may be
5 configured to an API to provide local or remote execution of applications that will utilize that information. The information may be locally stored in the device, or alternately stored remotely at a network storage site. Similarly, new information (e.g., "cookies") may be added to the information. Users may have the ability to review the information and choose to transmit or block information that is shared with other applications.

10 For example, a personal API configuration can enable a mobile device to query for items like name, postal address, e-mail address or query for permission to utilize e-mail address in posting lists for new products. A mobile web service can query the API when the user is registering for the first time to a service. Privacy control features may then be used to prompt an end-user for permission to release such information. This enables an end user to
15 be anonymous until registration, or until an m-commerce transaction is initiated. If registration to a specific web service is done, additional API may be provided to mobile web services to store the user registration ID to the mobile terminal, and enable a secure environment for the user. Furthermore, privacy controls may be used to enable automatic transmission to/from web services. Thus, web services in which the user is registered to may
20 make an unsolicited "push" of mobile web services to the terminal when updated or new information becomes available.

As an example, terminal 700 could be initially engaged in a search to find organizations or companies that support the mobile web services of the terminal 700. The terminal may additionally include other search criteria, such as those illustrated in **Figures 1-**
25 **1H**. The search may be done directly with UDDI or may utilize WAP or some other browser intermediate between terminal 700 and a portal capable of accessing a registry. When a web server site 712 has been found that supports the desired service, terminal 700 can initiate communication through an invocation client or browser (see **Figure 6C-D**) to the invocation web service in a server web service site 712. Once communication is established, terminal
30 700 can request mobile web services (e.g., "B2C e-Commerce Mobile Web Service") from

the server invocation web service. Once the request has been received, the server invocation web service will transmit to terminal 700 the appropriate web server instance (i.e., mobile web service), and will simultaneously activate the server side logic 642. Once transmitted, the mobile web service may be automatically or manually activated. Once activated, the web service will communicate with the appropriate API's in the terminal, and will subsequently connect with the service logic in the web service server 638 in the web service site 712.

Communication from terminal 700 to the network may additionally include Secure Sockets Layer (SSL). SSL is known in the art and may be defined as a transport level technology for authentication and data encryption between a server and a browser. SSL typically sends data over a "socket", which is a secure channel located at the connection layer existing in most TCP/IP applications. Sites can use the protocol to obtain and/or manage confidential user information, such as credit card numbers. The information is usually encrypted, and only the user's Web browser and the computer server at the other end running the web site have the key, and thus can understand what is being transmitted. Thus any unwanted outside parties would be prevented from understanding what the user was transmitting. The mobile web service 702 may be configured to access API's in the terminal 700 execution environment, (i.e., the "sandbox") to determine which user data is transmitted.

Thus for a UDDI invocation method, each individual advertised web service can be modeled in a *binding_Template* structure. Invocation of a web service can be performed based on cached *binding_Template* data (discussed above). Using a *find_XX* inquiry API, a UDDI compatible browser may display more or less detail as the user searches through information. The following example steps through one iteration of discovering a remote web service:

(1) Access UDDI business registry via web interface that uses the inquiry API to locate *businessEntity* information by or for the appropriate business that is advertising the web service;

(2) Drill-down for more detail about a *businessService* or request a full *businessEntity* structure (*businessEntity* structure typically contains all information regarding advertised web services);

(3) Select a desired *bindingTemplate* – once the desired information is located, the *get_XX* call returns full information about the key UDDI structures and saves particular *bindingTemplates* for later use;

(4) Run program based on the knowledge of the specifications for the web service
5 – this information may be obtained by using a *tModel* key information contained in the *bindingTemplate* for a service;

(5) The program invokes the web service using the cached *bindingTemplate* information (as appropriate) and implements the required interface conventions as defined in the specification referenced in the *tModel* information.

10

Figure 8A discloses exemplary layers involved during the use of mobile web services. Terminal **817** initiates a network connection or search **800**. Knowledge engine **801** supplements search **800** with information retrieved from customer profile **802**, and a transmission is made to UDDI cloud **803** located in description/discovery services layer **808**.
15 Web services **804** communicate with service logic **805**, which in turn communicates with back office systems **806**. The resulting invoked web service **807** is transmitted back to terminal **817**. Service delivery options include executable content (e.g., JAVA, WML Scripts, etc.) that are downloadable to terminal **817**. This executable content contains the “front-end” logic that would communicate with the “back end” logic located in the web
20 server or business logic. Mobile web services that have been discovered may be contacted by web service instance in the mobile terminal. This service instance may be either stored in the terminal semi-statically, or it may be dynamically loaded, as described below.

Figure 8B illustrates mobile web services with invocation capabilities. **Figure 8B** has the same basic structure as that shown in **Figure 8A**, except that additional functions and connections are established with mobile web services layer **816**. Web services layer **809** is
25 used to find web sites that support invocation web services **811** for terminal **817**. The downloaded invocation service **810** enables stored mobile web services **813** that are transmitted from web services **804**. Mobile web services **813** are associated with service logic **805** from web services **804** or can be associated with service logic from other remote

sites (see **Figure 6D**). Service logic **815** in mobile web services layer **815** handles terminal-side processing, and may be enabled to link directly to back office systems **816**.

During the discovery phase, sites can be matched that have the capability to support both invocation web services **811** and certain mobile web services **813**. This means that a
5 search **800** for such a request for include a logical AND combination of terms querying “invocation web service support” and “mobile web service [X]”, wherein “X” would represent the supported invocation mobile web service for the invocation web service.

Mobile web services layer **814** becomes active when mobile web service instance is received
10 **814** and activated in terminal **817**. The mobile web services layer **816** becomes active when mobile web service instance **814** is received and activated in mobile terminal **817**. Upon activation, mobile web service instance **814** begins to communicate with server logic **805** in web services **804** to enable the services provided and supported by web service **804**. With added privacy/security protocols, invocations of web services (and connections through API's) may be limited to web service instances authorized for a particular user or terminal.

15 Terminal **817** may make a request for a supported mobile web service during or after making an initial search. A matching service instance located in web services **804** will be identified to terminal **817** software. Once the web service instance is activated, the appropriate server acknowledges the presence of terminal **817** for further communications under a mobile web service environment. Thus, as an example, terminal **817** may initiate a
20 mobile terminal to search (e.g., UDDI registry) for all services that provide *mCommerce_services* and *CD_retail* and *mobile_web_services*. All of the addresses returned to terminal **817** will be identified as being online CD retailers, wherein each service provider infrastructure is enabled to provide mobile web service software to mobile terminal **817**. Through a software module resident in terminal **817**, a request can be made to web
25 services **804** to retrieve and download mobile web service *mCommerce*. Mobile web service *mCommerce* is typically an executable application that will control and optimize user interfaces and terminal-side functions. The mobile web service will then establish terminal-server functionality for invoking further services. The executable content can now contact the server logic **815** to utilize *mCommerce* retail services. The executable application may be
30 temporarily downloaded (e.g., JAVA applet), or may reside permanently in a terminal.

Additionally, the application may be sent statically, or may be dynamically generated to match requests made by various terminals. Once a server has prior knowledge of a terminal possessing mobile web services, the server may automatically push subsequent mobile web service instances to the terminal as needed.

5 The system and methods described above allows businesses, for example, with “eCommerce” sites to easily incorporating “mCommerce” capabilities by implementing the invocation server in the web services server.

 The mobile web service (or web service instance) that is temporarily downloaded to the terminal coordinates with the service logic to communicate with the “eCommerce” site’s
10 back office systems and business logic. The service provide may also tailor searches to direct selected portions of the search results to “preferred” service sites.

 The resulting invention is applicable to virtually all digital communications networks, including wide area networks (WANs), metropolitan area networks (MANs), local area networks (LANs), and personal area networks (PANs). The resulting invention is applicable
15 to fixed station wireline networks, mobile wireless networks, and hybrid combinations of fixed station wireline networks communicating through wireless access points with mobile wireless networks. In particular, the resulting invention is applicable to any mobile computing environment, including any wireless wide area network such as a cellular telephone network or any short range wireless system such as a wireless local area network
20 or a wireless personal area network. Examples of wireless, wide area network architectures to which the invention applies include Global System for Mobile Communication (GSM), IS-136 TDMA-based Digital Advanced Mobile Phone Service (DAMPS), Personal Digital Cellular (PDC), IS-95 CDMA-based cdmaOne System, General Packet Radio Service (GPRS) and broadband wireless systems such as W-CDMA, and Broadband GPRS.
25 Examples of short-range wireless systems to which the invention applies include the Bluetooth Standard, the IEEE 802.11 Wireless LAN Standard the HIPERLAN Standard, the IEEE 802.15 Wireless Personal Area Network (WPAN) standard, the Infrared Data Association (IrDA) standard, the Digital Enhanced Cordless Telecommunications (DECT) standard, the Shared Wireless Access Protocol (SWAP) standard, the Japanese 3rd

Generation (3G) wireless standard, and the Multimedia Mobile Access Communication (MMAC) Systems standard of the Japanese Association of Radio Industries and Businesses.

Although illustrative embodiments have been described herein in detail, it should be noted and understood that the descriptions and drawings have been provided for purposes of
5 illustration only and that other variations both in form and detail can be made thereupon without departing from the spirit and scope of the invention. The terms and expressions have been used as terms of description and not terms of limitation. There is no limitation to use the terms or expressions to exclude any equivalents of features shown and described or portions thereof.

CLAIMS

What is claimed is:

1. A method to enable a wireless device to discover mobile web services comprising:
accessing a Universal Description, Discovery and Integration (UDDI) registry via
5 web interface, wherein UDDI registry contains downloadable executable content associated
with a request transmitted from a wireless terminal;
receiving said executable content to said wireless terminal;
activating executable content in said wireless terminal, wherein said executable
content implements a connection to a remote server to receive web services.
10
2. The method of claim 1, wherein executable content is in the form of a JAVA applet.
3. The method of claim 1, wherein executable content is in the form of a WML script.
- 15 4. The method of claim 1, wherein said executable content implements a connection
through a terminal application interface (API).
5. The method of claim 4, wherein said connection is dynamically established with the
API.
20
6. A method to enable a search agent to discover Internet businesses or services by
accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:
receiving a search agent handle that will be associated with a user's search profile;
receiving query terms as at least part of a business name;
25 sending a *find_business* XML inquiry to a search agent, wherein said search agent
processes and forwards the *find_business* XML inquiry to a UDDI registry; and
receiving back from the search agent a *businessList* message that contains a list of
business names satisfying the *find_business* query in accordance with user's search profile.
- 30 7. The method of claim 6, which further comprises:

processing an item from the returned *businessList* message in accordance with user's search profile;

transmitting the selected business' entity data;

sending a *find_service* XML inquiry to the UDDI registry;

5 receiving back from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business.

8. The method of claim 7, which further comprises:

selecting an item from the returned *serviceList* message;

10 transmitting the selected service data;

sending a *_get_serviceDetail_* XML inquiry to the UDDI registry;

receiving back from the UDDI registry, a *serviceDetail* message that includes *bindingTemplate* data that contains the details of the selected service.

15 9. The method of claim 8, which further comprises:

including in the *bindingTemplate* data an *accessPoint URL*, which is the URL of the selected service on the web site of the selected business.

10. The method of claim 9, which further comprises:

20 transmitting the *accessPoint URL* to the user.

11. The method of claim 9, which further comprises:

storing the user profile with the selected *accessPoint URL*.

25 12. The method of claim 9, which further comprises:

said user profile including the business name query, the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data.

13. A system comprising:

30 (a) a wireless device configured to communicate over a network;

(b) a memory device, communicatively coupled to the wireless device, wherein said memory device stores at least one user profile and at least one executable program; and

(c) a processor, communicatively coupled to the memory device, wherein said processor and memory function to activate said executable program to facilitate access to a
5 network element.

14. The system of claim 13, wherein said at least one executable program is a JAVA applet.

10 15. The system of claim 13, wherein said at least one executable program is a WML script.

16. The system of claim 13, wherein said network element is a Universal Description, Discovery and Integration (UDDI) registry.

15

17. The system of claim 13, wherein said network element is a server including a knowledge engine.

18. The system of claim 13, wherein said network element includes a user profile that
20 further includes a search strategy.

19. The system of claim 18, wherein said user profile includes at least one search strategy that is stored by using a search handle for a business name query.

25 20. The system of claim 19, wherein said search handle for the business name query comprises a business entry data.

21. The system of claim 19, wherein said search handle for the business name query comprises a business service data.

30

22. The system of claim 19, wherein said search handle for the business query comprises a binding template data.

23. The system of claim 22, wherein said binding template data comprises an access point
5 URL of the selected service.

24. The system of claims 13, wherein the wireless device stores a search handle in a user profile with the search strategy of the network element.

10 25. The system of claims 19, wherein the wireless device stores a search handle in a user profile with the search strategy of the network element.

26. A system for enabling a remote wireless device to access internet businesses or services comprising:

15 (a) at least one memory device, communicatively coupled to a communications network; and

(b) at least one processor device, communicatively coupled to the at least one memory device, wherein said at least one processor device and said at least one memory device function to access files in accordance with at least one search handle transmitted from
20 the remote wireless device to a network element.

27. The system according to claim 25, wherein the accessed files are readable or executable computer code stored in a web site.

25 28. The system according to claim 26, wherein the accessed files are cached for selective forwarding to the wireless device.

29. The system of claim 25, wherein said network element is accessed using a direct session.

30

30. The system of claim 25, wherein said network element is accessed using an indirect session through a knowledge server.

31. The system according to claim 28, wherein said network element is a Universal
5 Description, Discovery and Integration (UDDI) registry.

32. The system according to claim 25, wherein the at least one search handle is associated with a user's search strategy.

10 33. The system according to claim 25, wherein said wireless device stores a search handle in a user profile with the search strategy of the network element.

34. A wireless device, configured to discover mobile web services comprising:
at least one Universal Description, Discovery and Integration (UDDI) enabled web
15 interface for transmitting to and from a UDDI registry;
a storage medium, communicatively coupled to the UDDI-enabled web interface,
wherein said storage medium stores at least one user profile and at least one executable
content;
a processor, communicatively coupled to the storage medium, wherein said processor
20 activates executable content to implement a connection to a remote server to receive web
services.

35. The method of claim 34, wherein executable content is in the form of a JAVA applet.

25 36. The method of claim 34, wherein executable content is in the form of a WML script.

37. The method of claim 34, wherein said executable content implements a connection through a terminal application interface (API).

38. The method of claim 37, wherein said connection is dynamically established with the API.

39. A method to enable a wireless device to discover Internet businesses or services by
5 accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:
forming of a query to the UDDI registry for the wireless device user;
constructing a personal user profile of the user's UDDI searching strategies; and
providing a shortcut for queries to the UDDI registry, in response to the user's entry
of abbreviated inputs to the wireless device.

10

40. The method of claim 39, wherein the method is embodied as programmed instructions executed within the user's wireless device to query the UDDI registry.

41. The method of claim 39, wherein the method is embodied as programmed
15 instructions executed within a separate knowledge engine server to query the UDDI registry in response to commands from the user's wireless device.

42. The method of claim 41, wherein the server caches files accessed from web sites, for selective forwarding to the user's wireless device.

20

43. A method to enable a wireless device to discover Internet businesses or services by
accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:
entering a search handle that will be associated with the user's search strategy;
entering query terms as at least part of a business name;
25 sending a *find_business* XML inquiry to the UDDI registry; and
receiving back from the UDDI registry, a *businessList* message that contains a list of
business names satisfying the *find_business* query.

44. The method of claim 43, which further comprises:
30 selecting an item from the returned *businessList* message;

drilling down in the selected business' entity data;
sending a *find_service* XML inquiry to the UDDI registry;
receiving back from the UDDI registry, a *serviceList* message that contains a list of
names of services offered by the selected business.

5

45. The method of claim 44, which further comprises:
selecting an item from the returned *serviceList* message;
drilling down in the selected service data;
sending a *_get_serviceDetail_* XML inquiry to the UDDI registry;
10 receiving back from the UDDI registry, a *serviceDetail* message that includes
bindingTemplate data that contains the details of the selected service.

46. The method of claim 45, which further comprises:
including in the *bindingTemplate* data an *accessPoint URL*, which is the URL of the
15 selected service on the web site of the selected business.

47. The method of claim 46, which further comprises:
displaying the accessPoint URL to the user.

20 48. The method of claim 46, which further comprises:
storing the search handle in a user profile with the selected accessPoint URL;
providing the user with a shortcut for accessing pages from web sites, in response to
the user's entry of abbreviated search handle to the wireless device.

25 49. The method of claim 46, which further comprises:
storing the search handle in a user profile with a UDDI registry search strategy;
providing the user with a shortcut for online or offline queries to the UDDI registry,
in response to the user's entry of abbreviated search handle to the wireless device.

30 50. The method of claim 49, which further comprises:

said search strategy including the business name query , the selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint* URL.

5 51. The method of claim 49, which further comprises:
replaying a UDDI registry search strategy by entering a search handle into the
wireless device;

automatically accessing the UDDI registry search strategy from user profile
corresponding to the search handle;

10 loading query values from said UDDI registry search strategy as each respective
operand that would have been otherwise entered by the user.

52. The method of claim 51, which further comprises:

15 said query values including the business name query , the selected *businessEntity*
data, the selected *businessService* data, and the selected *bindingTemplate* data.

53. A method to enable a wireless device to discover Internet businesses or services by
accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:

20 entering a search handle in a wireless device that will be associated with the user's
search strategy;

entering query terms in the wireless device as at least part of a business name;

transmitting the search handle and query terms to a knowledge engine server;

25 sending with the knowledge engine server a *find_business* XML inquiry to the UDDI
registry; and

receiving back at the knowledge engine server from the UDDI registry, a *businessList*
message that contains a list of business names satisfying the *find_business* query.

54. The method of claim 53, which further comprises:

selecting an item from the returned *businessList* message;

30 drilling down in the selected business' entity data;

sending with the knowledge engine server a *find_service* XML inquiry to the UDDI registry;

receiving back at the knowledge engine server from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business.

5

55. The method of claim 54, which further comprises:

selecting an item from the returned *serviceList* message;

drilling down in the selected service data;

sending with the knowledge engine server a *_get_serviceDetail_* XML inquiry to the

10 UDDI registry;

receiving back at the knowledge engine server from the UDDI registry, a *serviceDetail* message that includes *bindingTemplate* data that contains the details of the selected service.

15 56. The method of claim 55, which further comprises:

including in the *bindingTemplate* data an *accessPoint URL*, which is the URL of the selected service on the web site of the selected business.

57. The method of claim 56, which further comprises:

20 displaying the *accessPoint URL* to the user.

58. The method of claim 56, which further comprises:

storing with the knowledge engine server the search handle in a user profile with the selected *accessPoint URL*;

25 providing the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

59. The method of claim 56, which further comprises:

storing with the knowledge engine server the search handle in a user profile with a

30 UDDI registry search strategy;

providing the user with a shortcut for online or offline queries to the UDDI registry, in response to the user's entry of abbreviated search handle to the wireless device.

60. The method of claim 59, which further comprises:

5 said search strategy including the business name query , the selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint* URL.

61. The method of claim 59, which further comprises:

10 replaying a UDDI registry search strategy by entering a search handle into the wireless device;

transmitting the search handle to the knowledge engine server;

automatically accessing with the knowledge engine server the UDDI registry search strategy from user profile corresponding to the search handle;

15 loading with the knowledge engine server query values from said UDDI registry search strategy as each respective operand that would have been otherwise entered by the user.

62. The method of claim 61, which further comprises:

20 said query values including the business name query , the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data.

63. A method to enable a wireless device to discover Internet businesses or services by accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:

25 entering a search handle in a wireless device that will be associated with the user's search strategy;

entering query terms in the wireless device as at least part of a business name;

transmitting the search handle and query terms to a knowledge engine server;

searching web sites using URLs contained in stored binding templates;

30 retrieving documents resulting from the search of the web sites; and

applying a filter prescribed by the user and stored in the user's profile, to limit the returned documents to only those of particular interest to the user.

64. The method of claim 63, which further comprises:

5 sorting the documents in a list having an order established in accordance with user's profile.

65. The method of claim 64, which further comprises:

10 storing the filtered documents and the sorted list in a cache for later, selective accessing by the user.

66. The method of claim 65, which further comprises:

15 receiving the user's selections from the list and updating the user's profile with the user's preferences.

67. The method of claim 66, which further comprises:

20 associating the search handle with user's selections and with the user's search strategy;
 storing that association in user's profile.

68. The method of claim 67, which further comprises:

 providing the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

25 69. A system to enable a wireless device to discover Internet businesses or services by accessing the Universal Description, Discovery and Integration (UDDI) registry, comprising:

 a processor;

 a memory coupled to the processor, programmed to perform the steps of:

30 entering a search handle in a wireless device that will be associated with the user's search strategy;

entering query terms in the wireless device as at least part of a business name;
transmitting the search handle and query terms to a knowledge engine server;
sending with the knowledge engine server a *find_business* XML inquiry to the UDDI
registry; and

5 receiving back at the knowledge engine server from the UDDI registry a
businessList message that contains a list of business names satisfying the
find_business query.

10 70. A system to enable a wireless device to discover Internet businesses or services
by accessing the Universal Description, Discovery and Integration (UDDI)
registry, comprising:
a processor;
a memory coupled to the processor, programmed to perform the steps of:
forming a query to the UDDI registry for the wireless device user;
15 constructing a personal user profile of the user's UDDI searching strategies; and
providing a shortcut for queries to the UDDI registry in response to the user's
entry of abbreviated inputs to the wireless device.

20 71. A system comprising:
(a) a wireless device configured to communicate over a computer network;
(b) a memory device, communicatively coupled to the wireless device, wherein
said memory device stores at least one executable user profile and at least one
abbreviated input; and
(c) a processor, communicatively coupled to the memory device, wherein said
25 processor and memory function to access a network element in accordance
with the at least one executable user profile or at least one abbreviated input.

72. The system of claim 71, wherein said at least one executable user profile consists of
an abbreviated user input to the wireless device.

30

73. The system of claim 71, wherein said network element is a Universal Description, Discovery and Integration (UDDI) registry.

74. The system of claim 71, wherein said network element is a server including a
5 knowledge engine.

75. The system of claim 71, wherein said network element includes a user profile that comprises a search strategy.

10 76. The system of claim 71, wherein said search strategy is stored by using a search handle for a business name query.

77. The system of claim 76, wherein said search handle for the business name query comprises a business entry data.

15

78. The system of claim 76, wherein said search handle for the business name query comprises a business service data.

79. The system of claim 76, wherein said search handle for the business query comprises
20 a binding template data.

80. The system of claim 79, wherein said binding template data comprises an access point URL of the selected service.

25 81. The system of claims 72, wherein the wireless device stores a search handle in a user profile with the search strategy of the network element.

82. The system of claims 77, wherein the wireless device stores a search handle in a user profile with the search strategy of the network element.

30

83. A system for enabling a remote wireless device to access internet businesses or services comprising:

- (a) at least one memory device, communicatively coupled to a communications network; and
- 5 (b) at least one processor device, communicatively coupled to the at least one memory device, wherein said at least one processor device and said at least one memory device function to access files in accordance with at least one search handle transmitted from the remote wireless device to a network element.

10

84. The system according to claim 83, wherein the accessed files are readable or executable computer code stored in a web site.

85. The system according to claim 84, wherein the accessed files are cached for selective
15 forwarding to the wireless device.

86. The system of claim 83, wherein said network element is accessed using a direct session.

20 87. The system of claim 83, wherein said network element is accessed using an indirect session through a knowledge server.

88. The system according to claim 86, wherein said network element is a Universal Description, Discovery and Integration (UDDI) registry.

25

89. The system according to claim 83, wherein the at least one search handle is associated with a user's search strategy.

90. The system according to claim 83, wherein said wireless device stores a search handle
30 in a user profile with the search strategy of the network element.

91. A system for enabling a wireless device to access Internet businesses or services via a Universal Description, Discovery and Integration (UDDI) registry, comprising:

- 5 (a) means for entering and storing at least one user query and at least one search handle in a wireless device, wherein a particular one of the at least one user query is associated with a respective search handle;
- (b) means for transmitting the at least one query and at least one search handle to a remote server, wherein said server communicates with the UDDI registry to respond to the at least one user query or the at least one search handle; and
- 10 (c) means to receive the server response and interactively display the response on the wireless device.

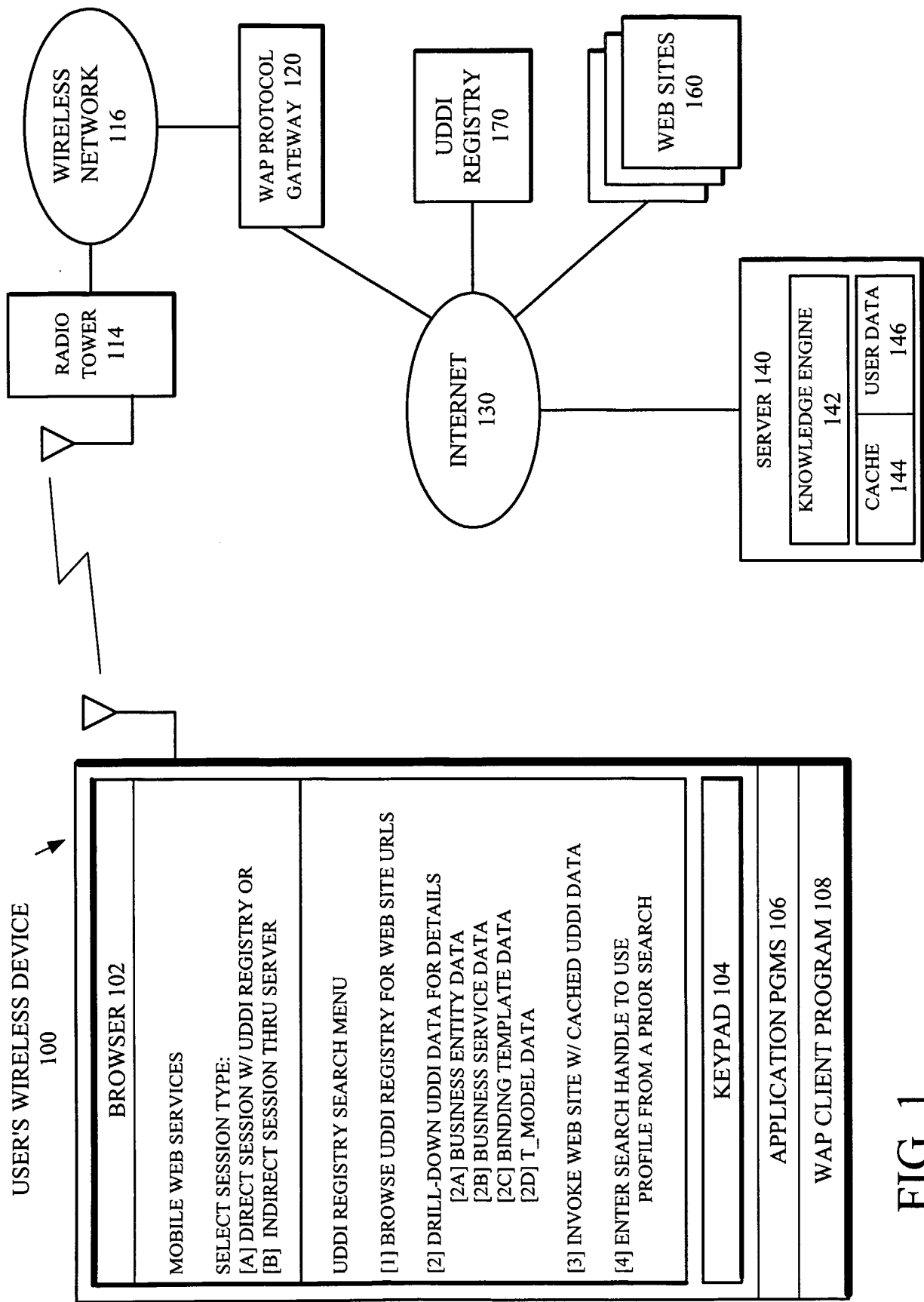


FIG. 1

WIRELESS DEVICE 100

BROWSER 102 MOBILE WEB SERVICES RESULTS FROM BROWING UDDI REGISTRY UNDER SEARCH HANDLE "_STOCKS_" RETURNED businessList MESSAGE: [1] WALL STREET JOURNAL ONLINE [2] CNBC FINANCIAL NEWS [3] MOTLEY FOOL FINANCIAL NEWS [4] YAHOO FINANCIAL NEWS SELECT ITEM FROM RETURNED LIST OR CONTINUE BROWSING	KEYPAD 104
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1B

WIRELESS DEVICE 100

BROWSER 102 MOBILE WEB SERVICES BROWSE UDDI REGISTRY FOR WEB SITE URLS ENTER SEARCH HANDLE "_STOCKS_" ENTER LETTERS OF FIRST TERM "WALL STREET_" IS THIS A BUSINESS NAME? Y/N "Y_" IS THIS A SERVICE NAME? Y/N "N_" ENTER LETTERS OF SECOND TERM "FINANC_" IS THIS A BUSINESS NAME? Y/N "Y_" IS THIS A SERVICE NAME? Y/N "N_" SEND "_find_business_" XML INQUIRY WITH SOAP PROTOCOL	KEYPAD 104
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1A

WIRELESS DEVICE 100

BROWSER 102	
MOBILE WEB SERVICES	
SELECT ITEM FROM RETURNED LIST	
DRILL-DOWN BUSINESS ENTITY DATA	
" _ WALL STREET JOURNAL ONLINE "	
SEND " _find_service_ "	
XML INQUIRY WITH SOAP PROTOCOL	
KEYPAD 104	
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1C

WIRELESS DEVICE 100

BROWSER 102	
MOBILE WEB SERVICES	
RESULTS FROM BROWSING UDDI REGISTRY	
UNDER SEARCH HANDLE " _STOCKS_ "	
RETURNED serviceList MESSAGE FOR	
WALL STREET JOURNAL ONLINE:	
[1] TECH STOCK NEWS	
[2] DRUG STOCK NEWS	
[3] ENERGY STOCK NEWS	
[4] COMMODITIES NEWS	
SELECT ITEM FROM RETURNED LIST	
OR	
CONTINUE BROWSING	
KEYPAD 104	
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1D

WIRELESS DEVICE 100

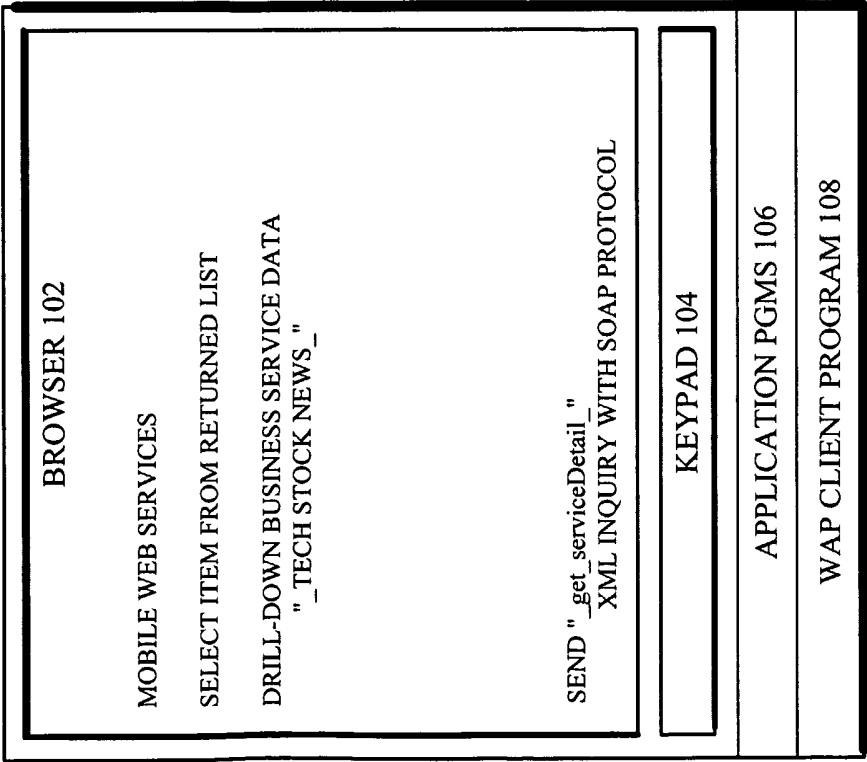


FIG. 1E

WIRELESS DEVICE 100

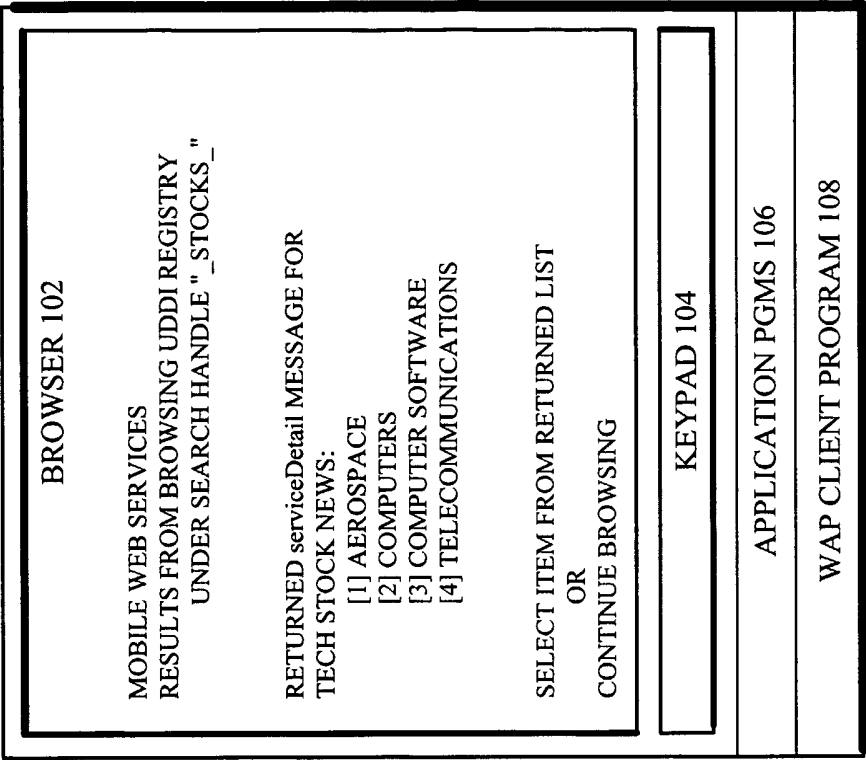


FIG. 1F

WIRELESS DEVICE 100

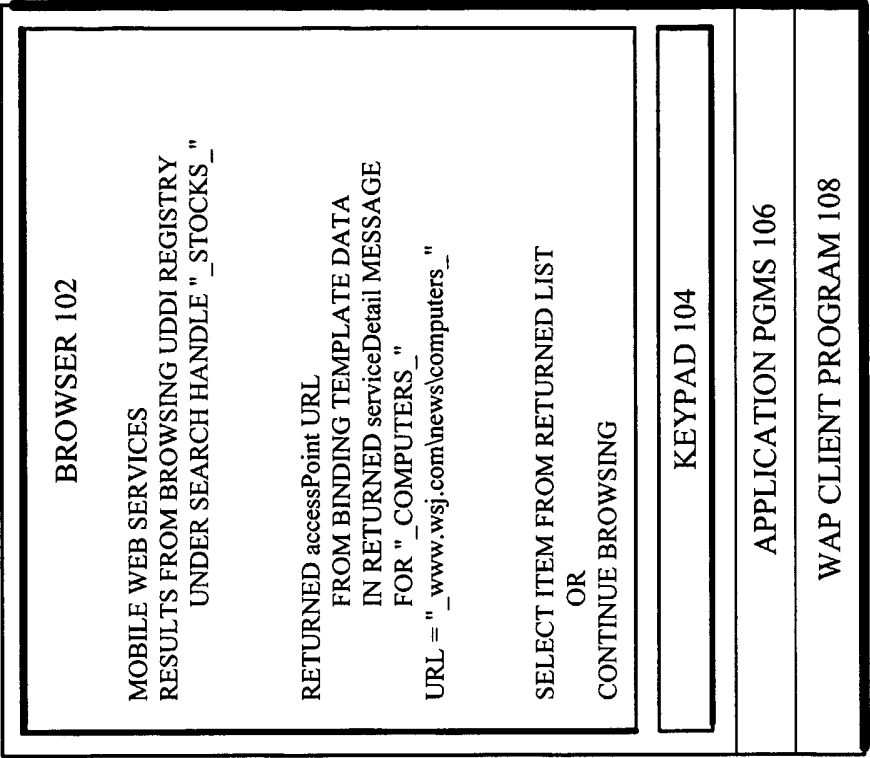


FIG. 1H

WIRELESS DEVICE 100

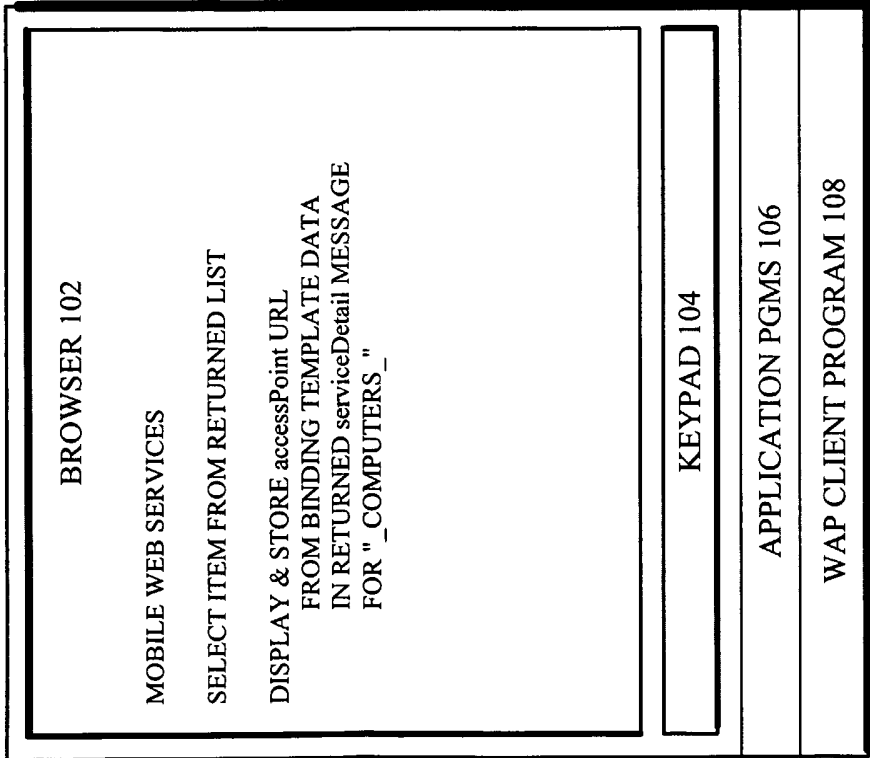


FIG. 1G

WIRELESS DEVICE 100

BROWSER 102	
MOBILE WEB SERVICES	
ENTER SEARCH HANDLE TO USE PROFILE FROM A PRIOR SEARCH	
ENTER SEARCH HANDLE " _STOCKS_ "	
SELECT TYPE OF SEARCH:	
[1] GET WEB PAGES FROM WEB SITE	
[2] REPLAY UDDI REGISTRY SEARCH STRATEGY	
KEYPAD 104	
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1I

WIRELESS DEVICE 100

BROWSER 102	
MOBILE WEB SERVICES	
RESULTS FROM REPLAYING UDDI SEARCH USING SEARCH HANDLE " _STOCKS_ "	
RETURNED accessPoint URLs	
FROM BINDING TEMPLATE DATA	
IN RETURNED serviceDetail MESSAGE	
FOR " _COMPUTERS_ "	
URL = " _www.wsj.com/news/computers_ "	
URL = " _www.wsj.com/quotes/computers_ "	
URL = " _www.wsj.com/graphs/computers_ "	
SELECT ITEM FROM RETURNED LIST	
OR	
CONTINUE BROWSING	
KEYPAD 104	
APPLICATION PGMS 106	
WAP CLIENT PROGRAM 108	

FIG. 1J

7/22

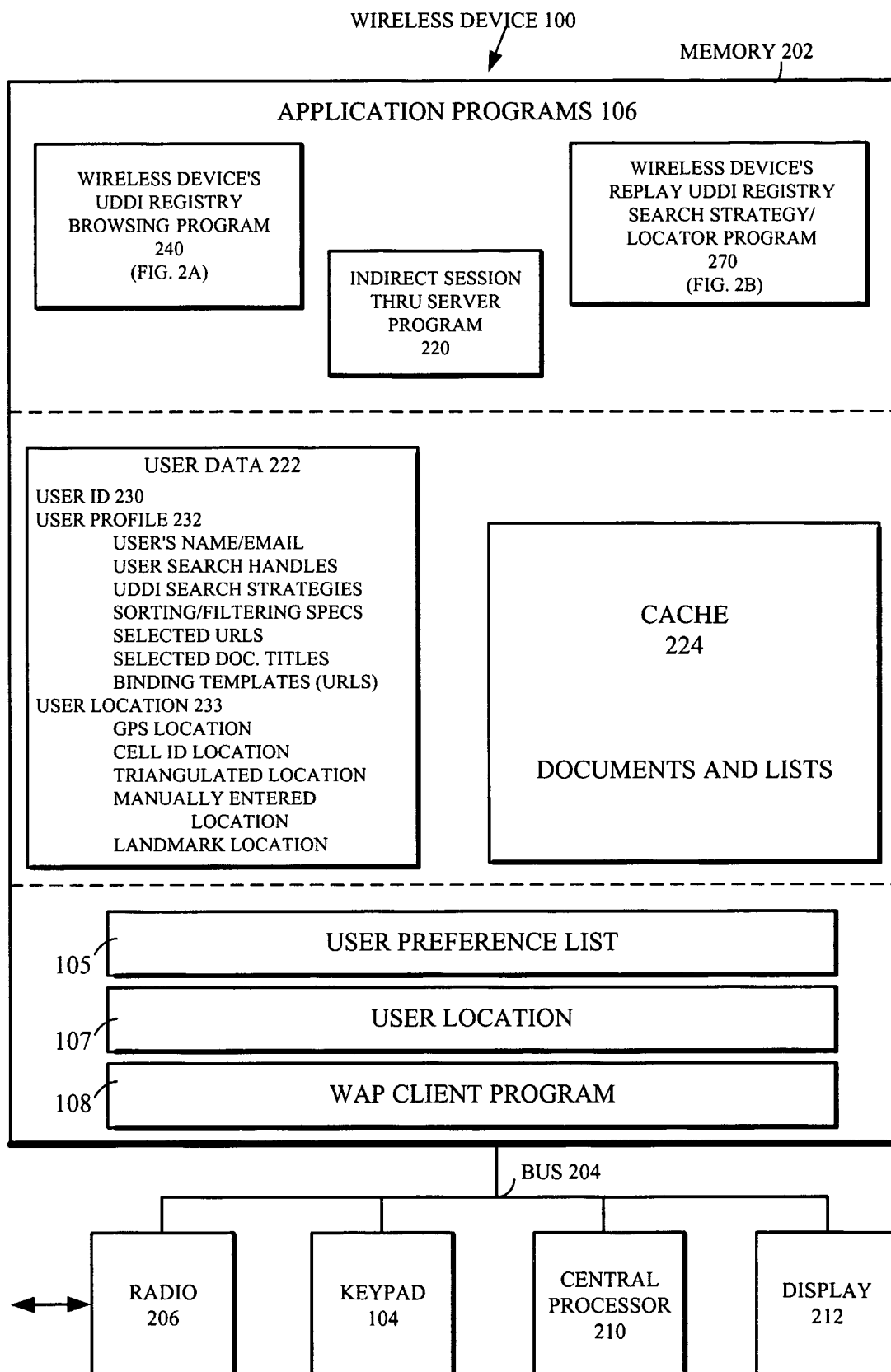


FIG. 2

8/22

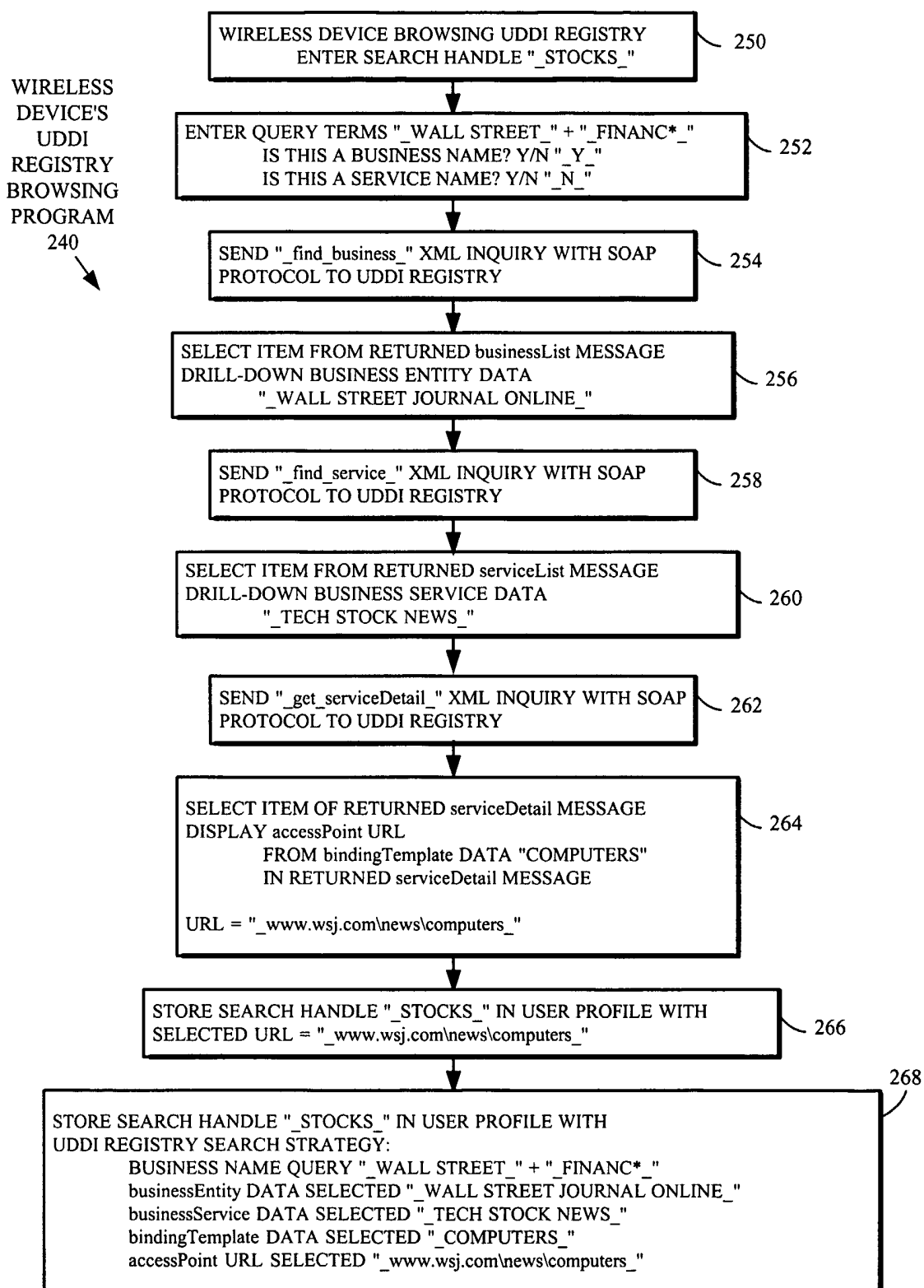


FIG. 2A

9/22

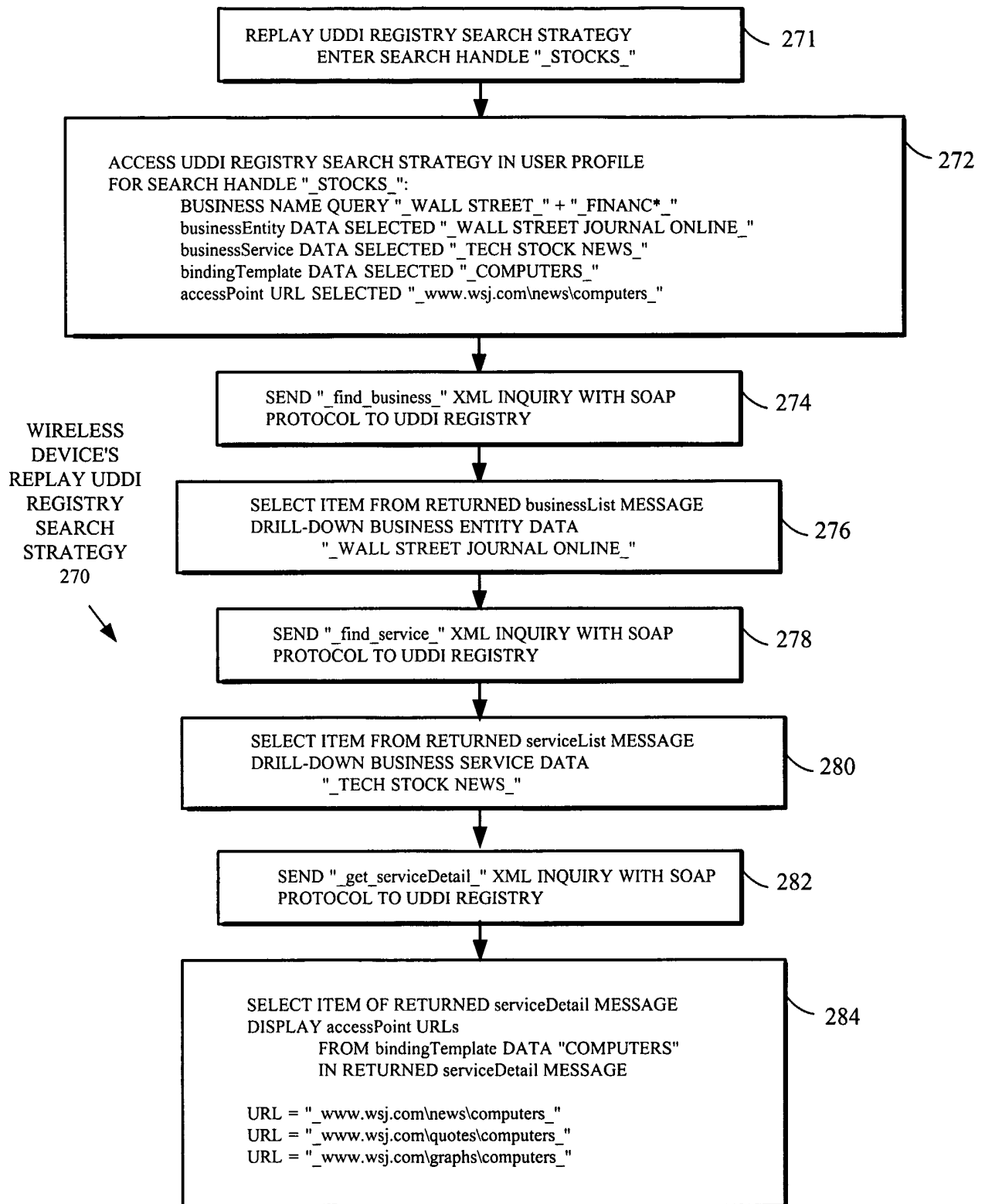


FIG. 2B

10/22

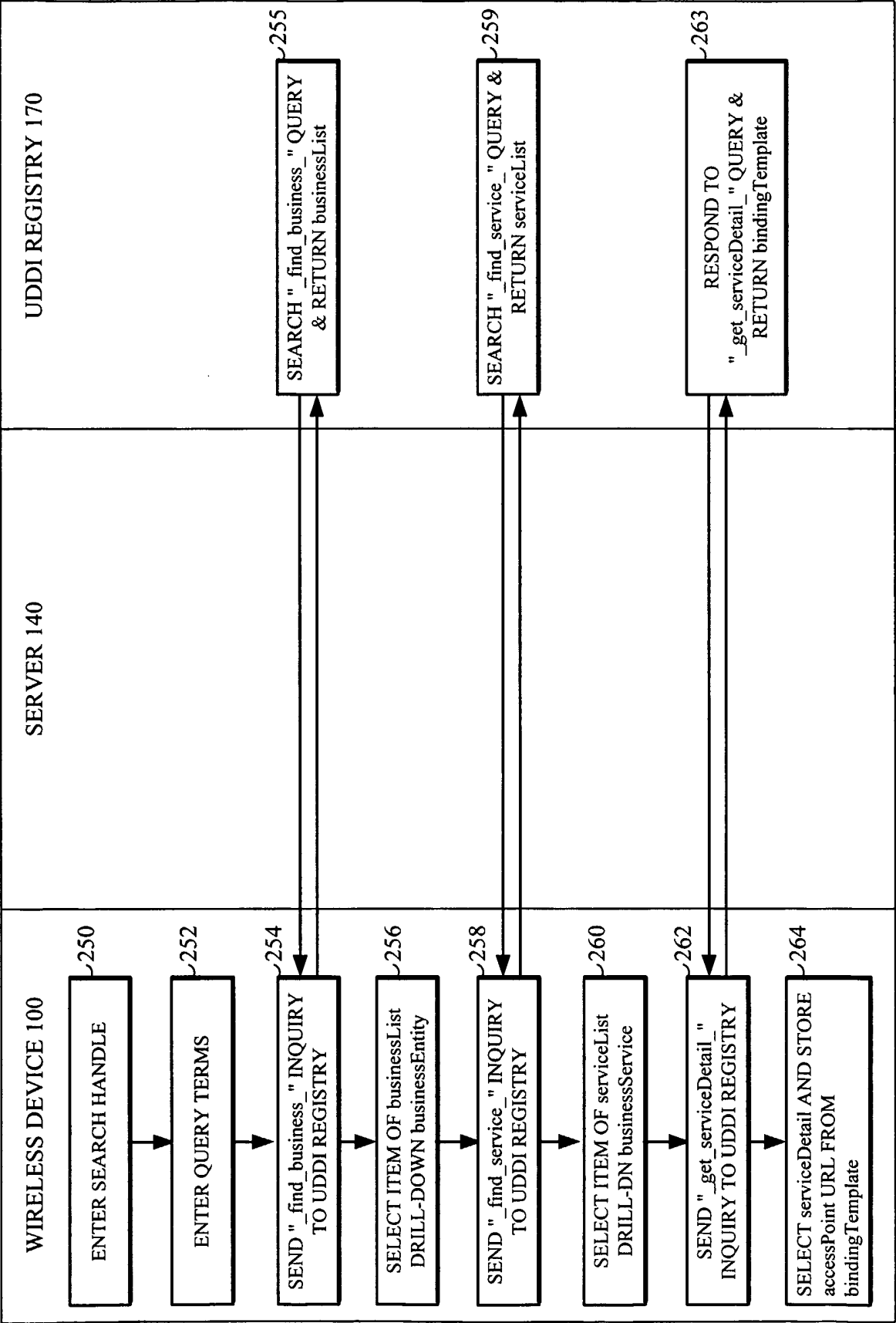


FIG. 3A

11/22

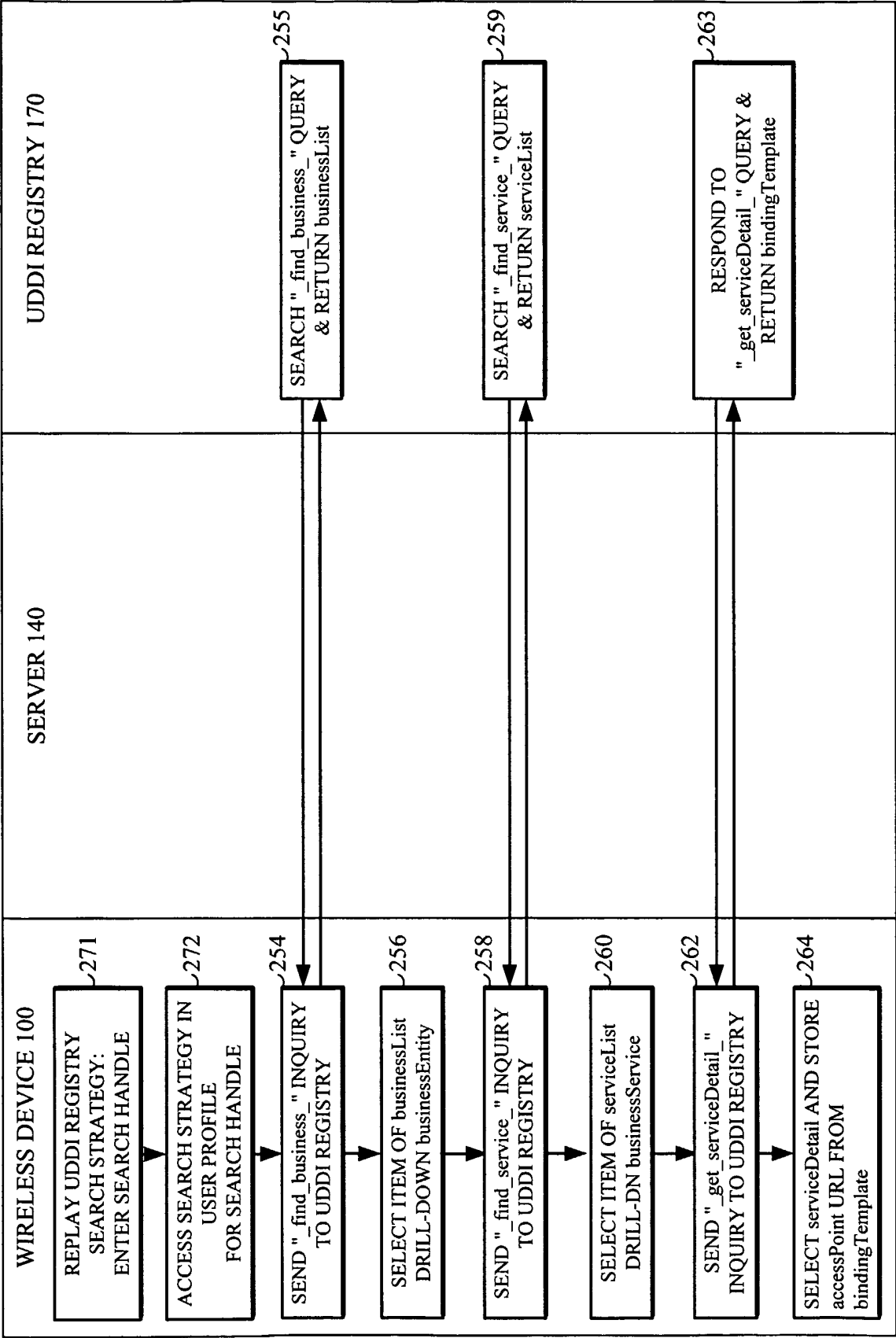


FIG. 3B

12/22

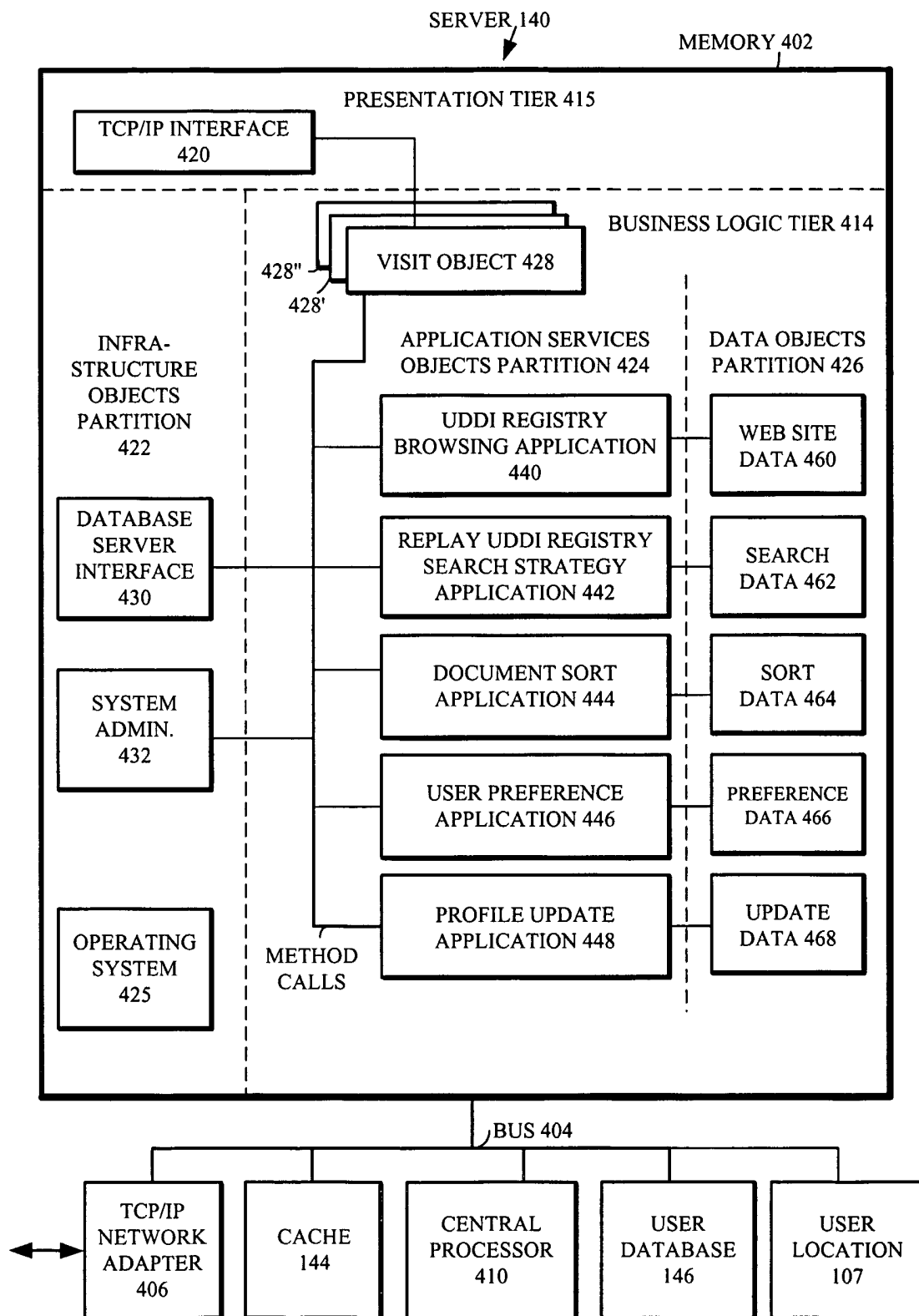


FIG. 4

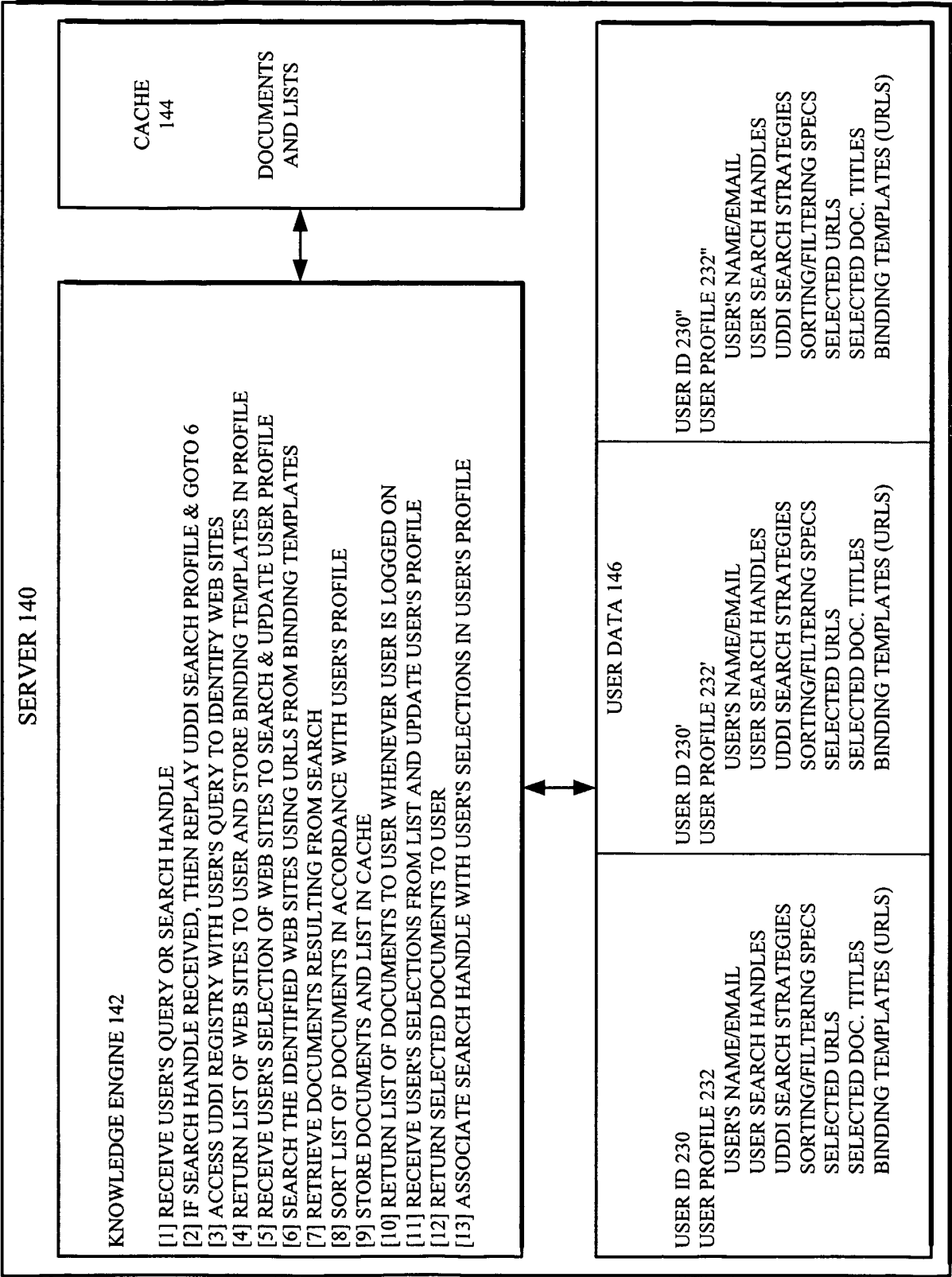


FIG. 4A

14/22

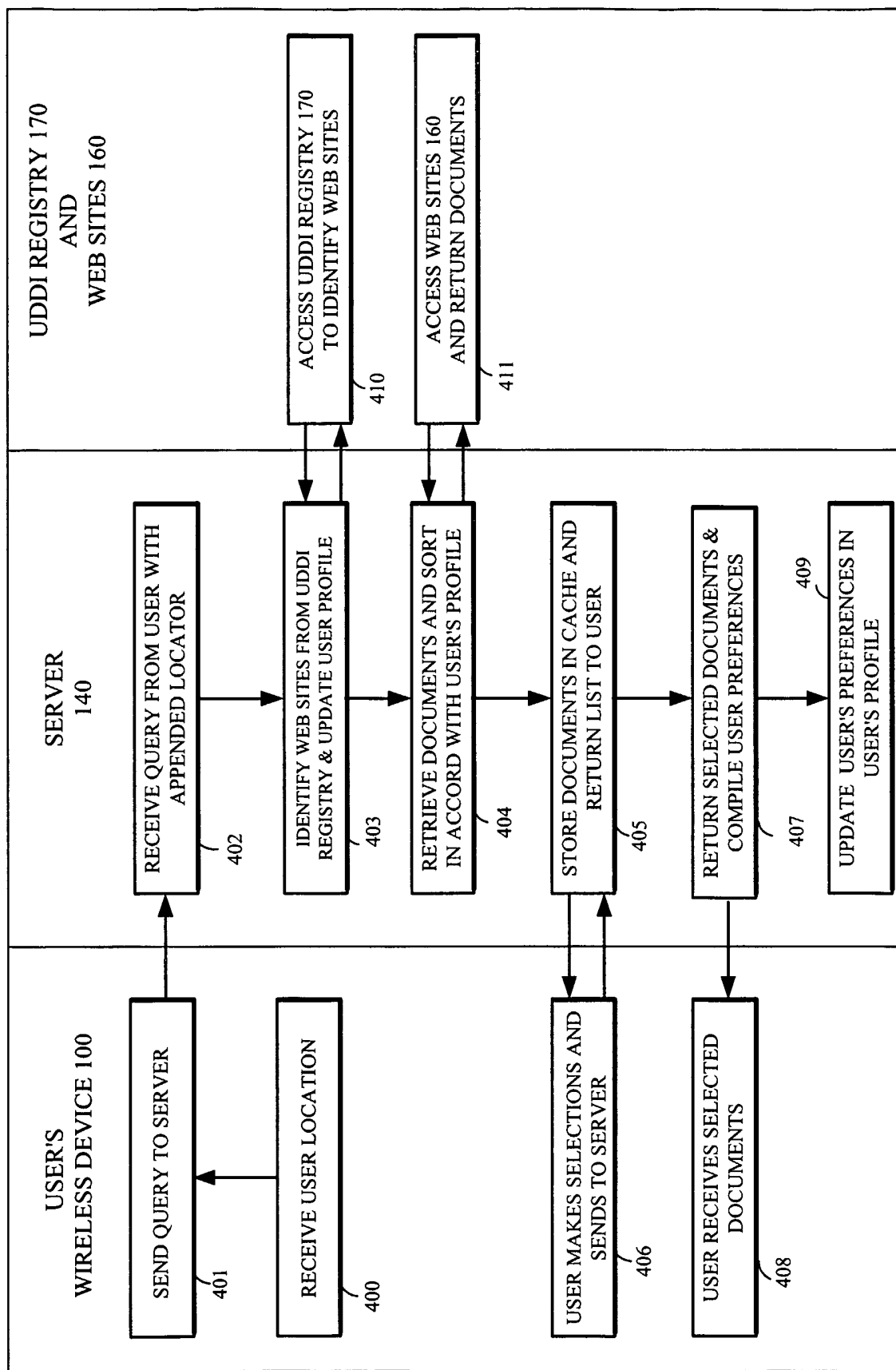


FIG. 4B

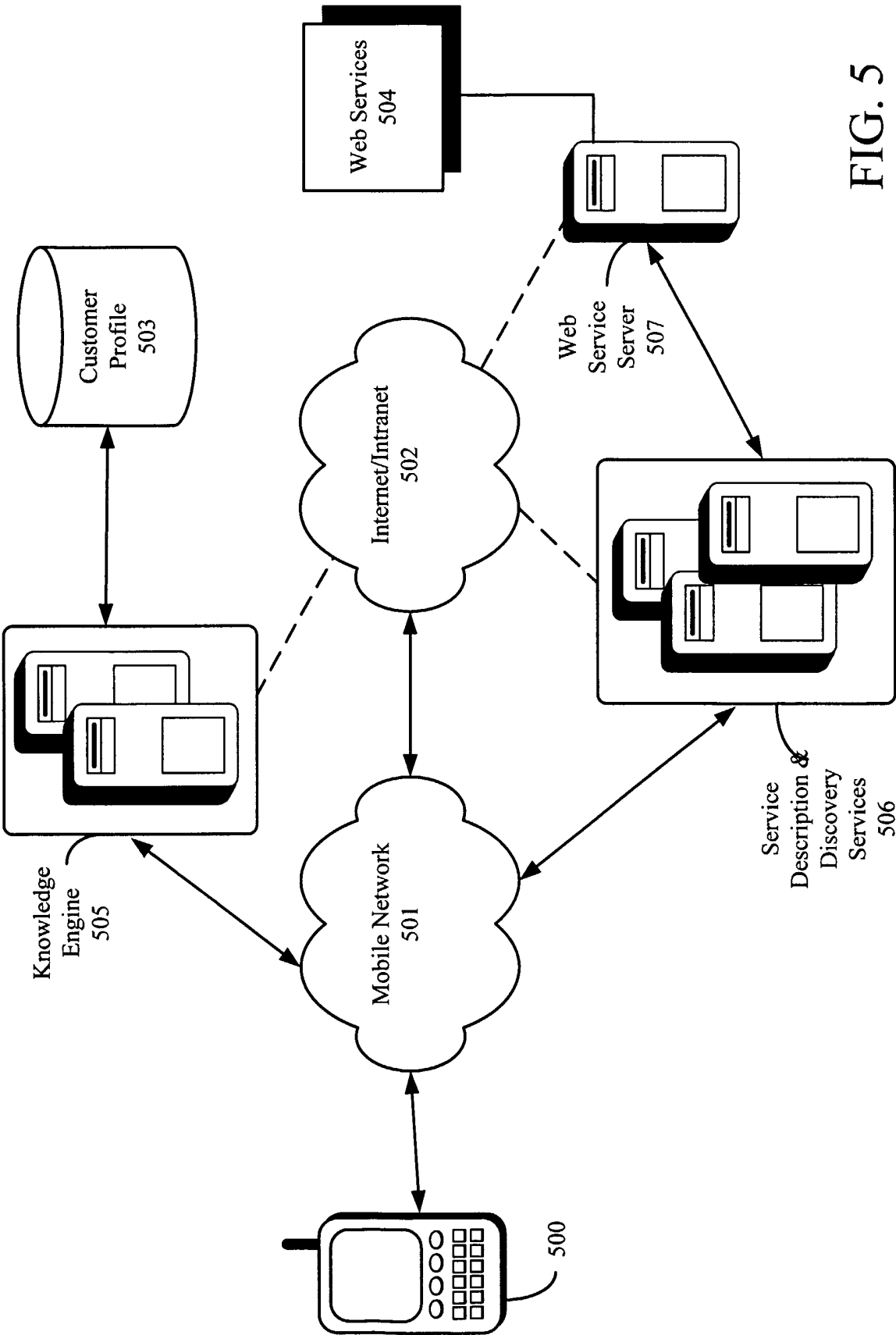


FIG. 5

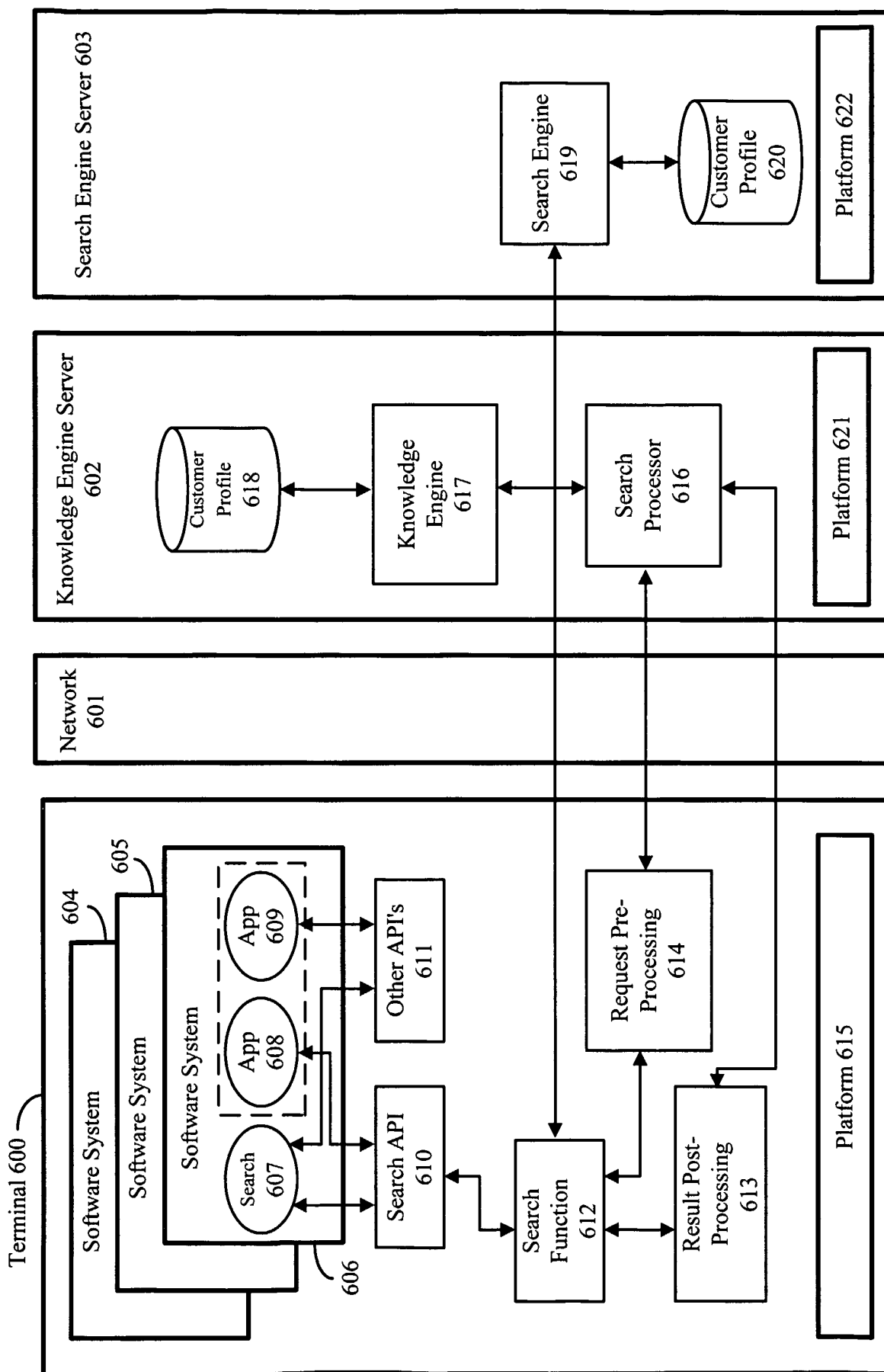


FIG. 6A

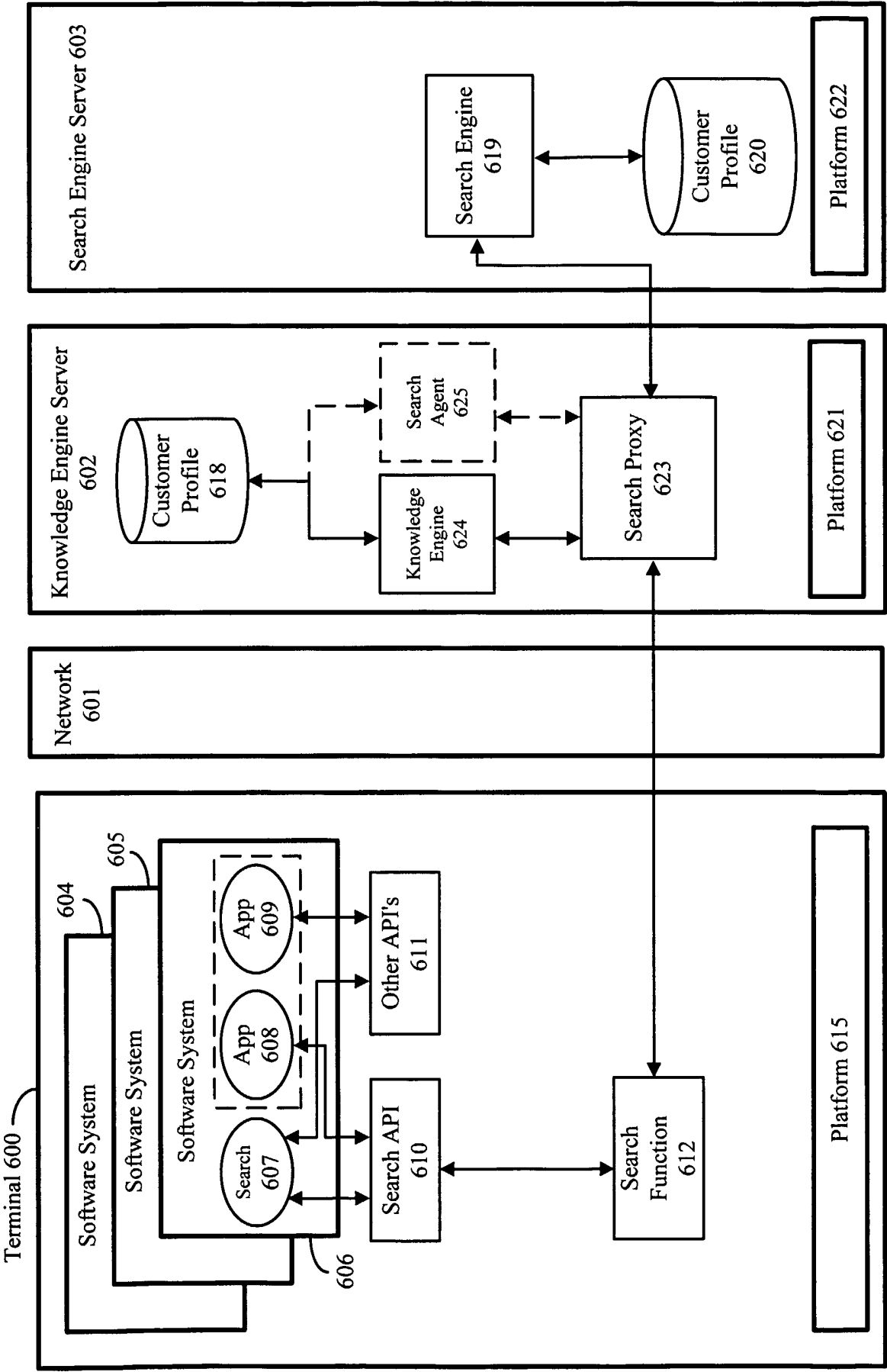


FIG. 6B

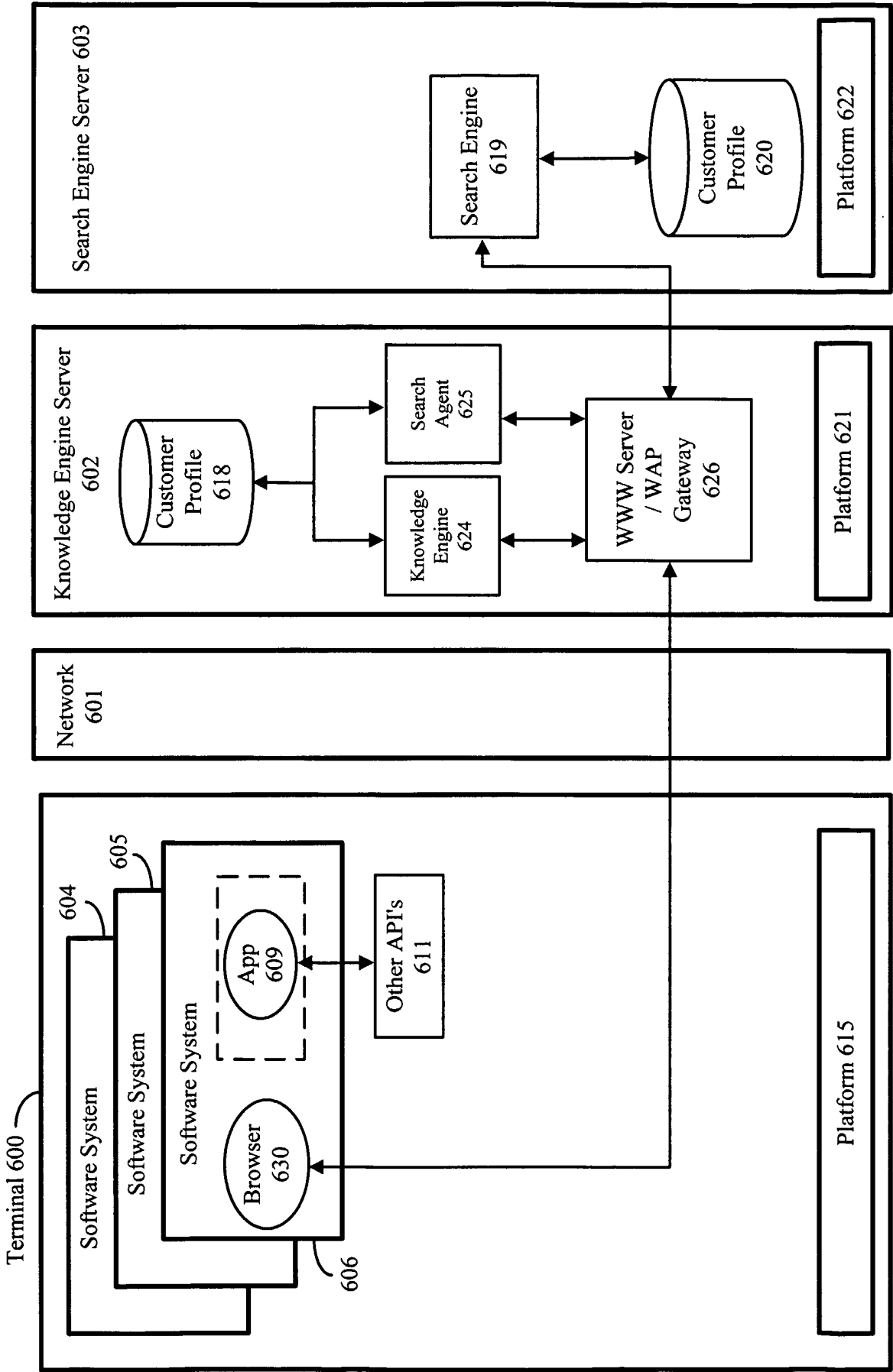


FIG. 6C

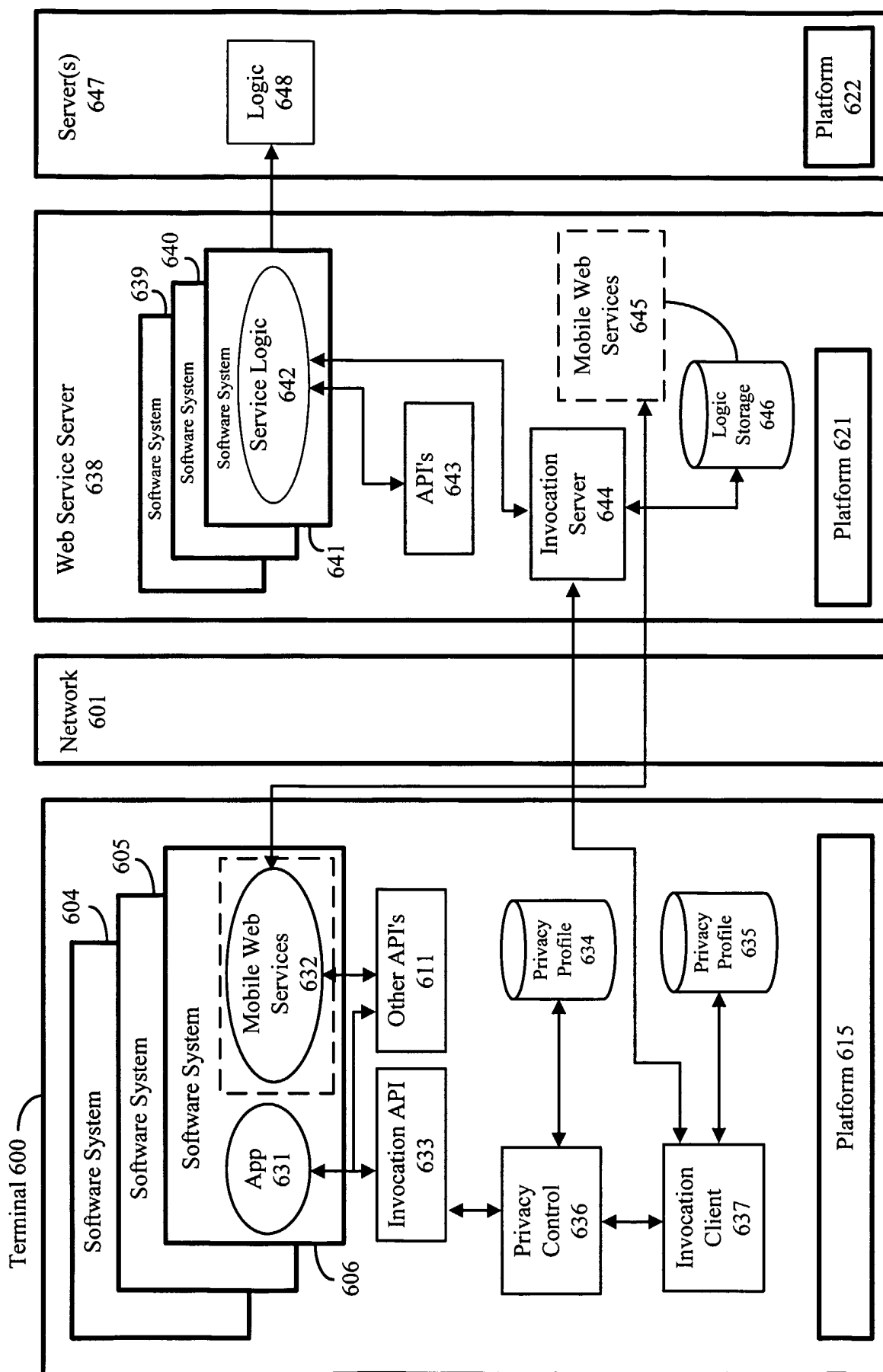


FIG. 6D

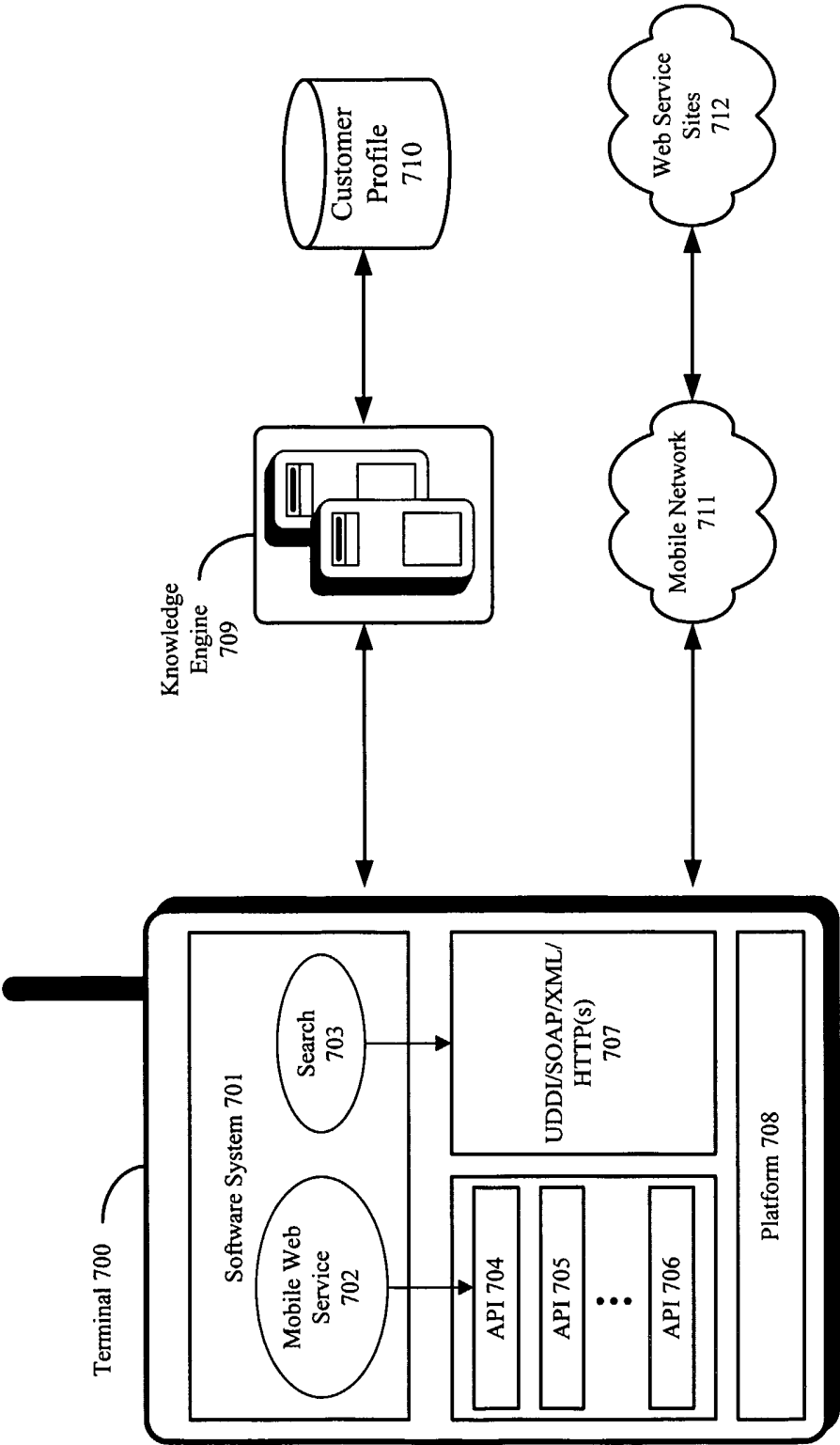


FIG. 7

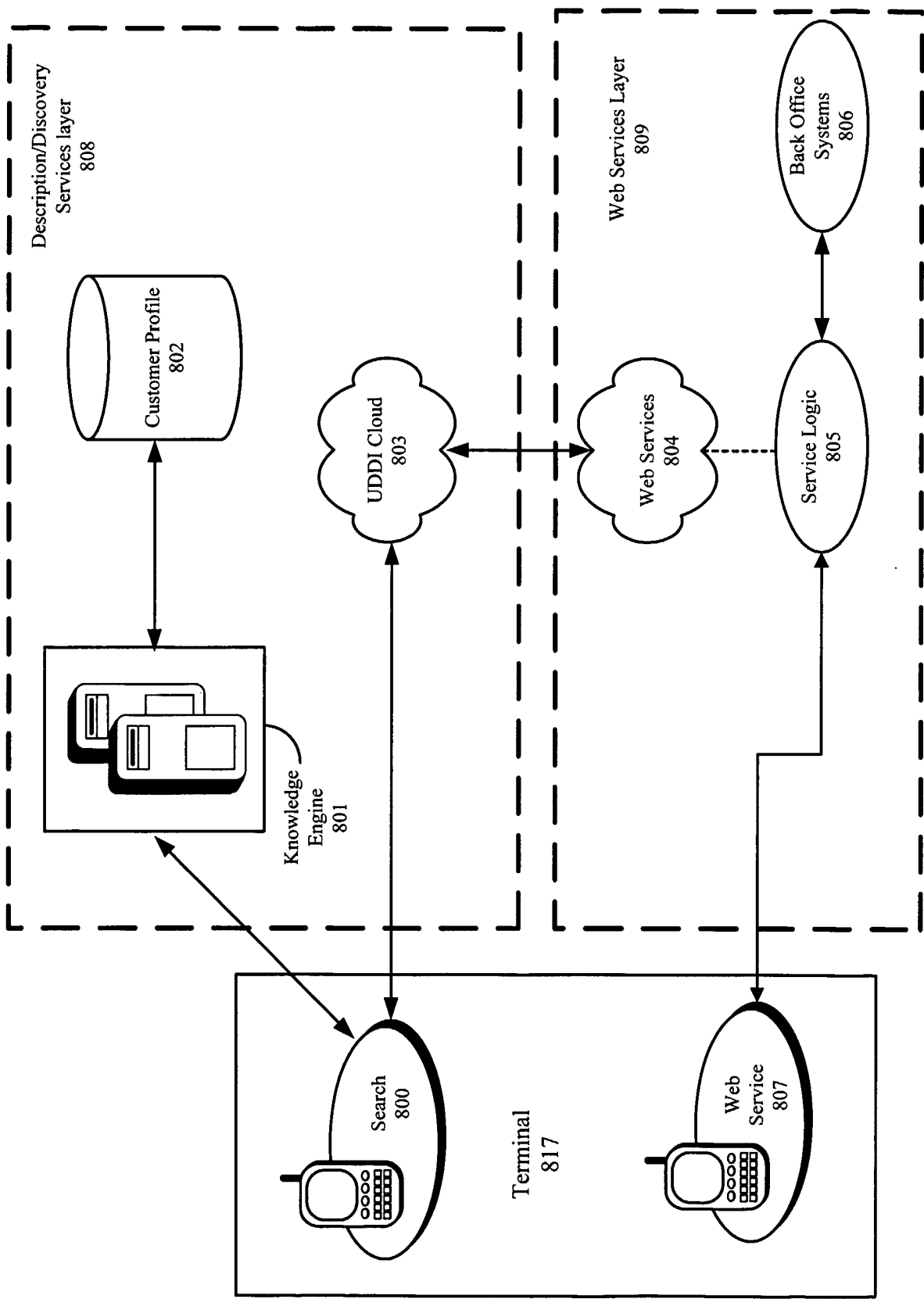


FIG. 8A

22/22

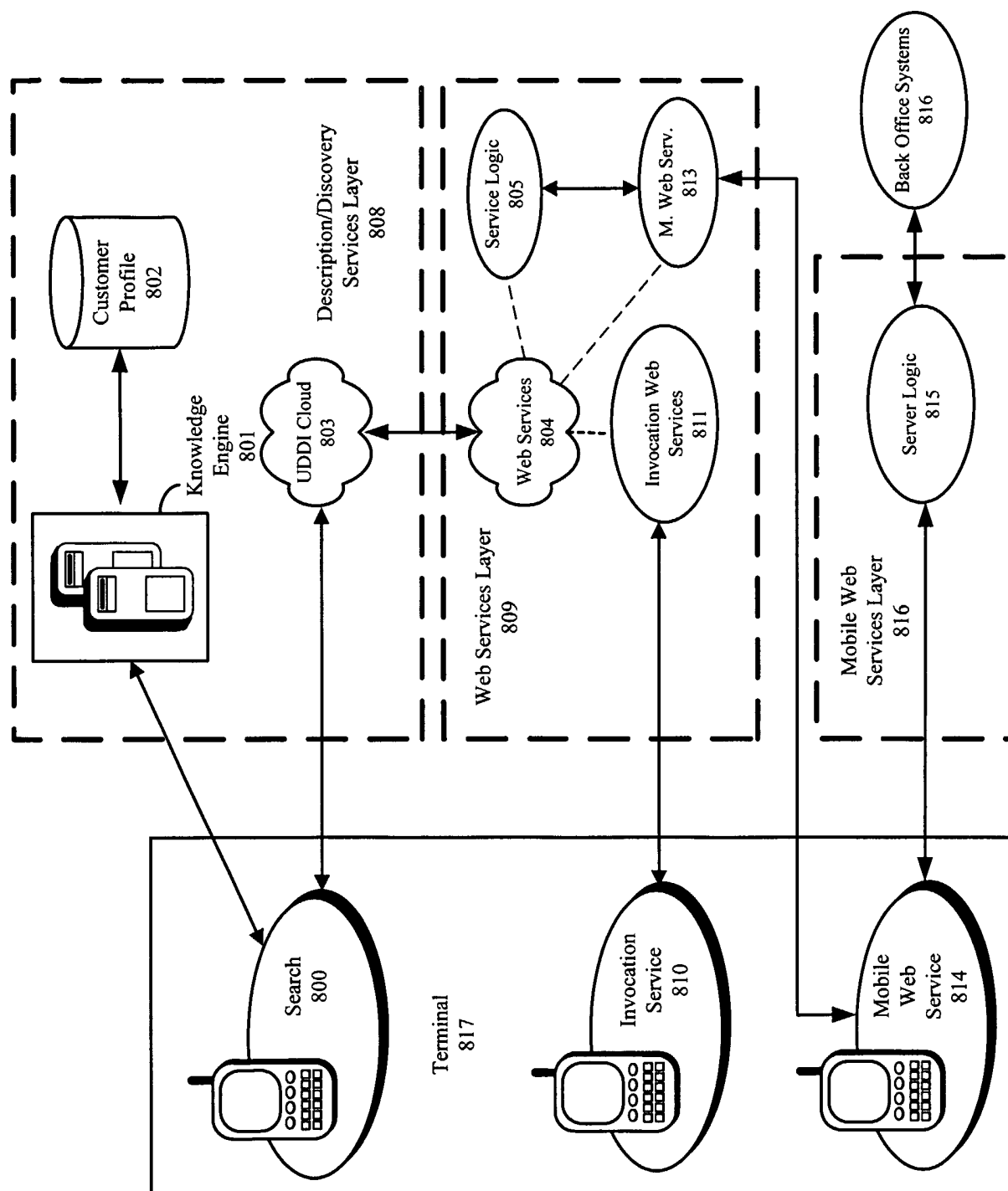


FIG. 8B