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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 February 2010 (18.02.2010)

PCT

(10) International Publication Number
WO 2010/019568 A1

(51) **International Patent Classification**:
*G01C 21/00* (2006.01)

(21) **International Application Number**:
PCT/US2009/053402

(22) **International Filing Date**:
11 August 2009 (11.08.2009)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
61/188,564 11 August 2008 (11.08.2008) US

(71) **Applicant** *(for all designated States except US)*: **TELCORDIA TECHNOLOGIES, INC.** [US/US]; One Telcordia Drive 5G116, Piscataway, NJ 08854-4157 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **SHIM, Hyong** [US/US]; 76 Liberty Ridge Road, Basking Ridge, NJ 07920 (US). **PIETROWICZ, Stanley** [US/US]; 65 Country View Drive, Freehold, NJ 07728 (US). **ZHANG, Tao** [US/US]; 247 Harmon Avenue, Fort Lee, NJ 07027 (US). **WULLERT, John, R., II** [US/US]; 1106 Mayflowr Court, Martinsville, NJ 08836 (US).

(74) **Agents**: **FEIG, Philip, J.** et al.; Telcordia Technologies, Inc., One Telcordia Drive 5G116, Piscataway, NJ 08854-4157 (US).

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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published**:

— *with international search report (Art. 21(3))*

(54) **Title**: SYSTEM AND METHOD FOR USING NETWORKED MOBILE DEVICES IN VEHICLES

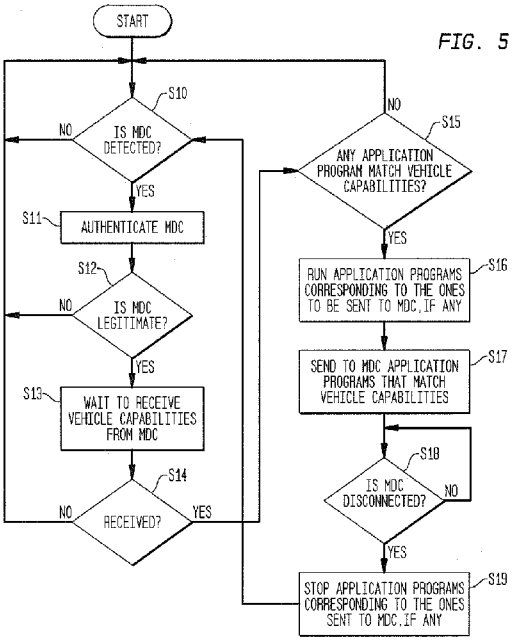



FIG. 5

(57) **Abstract**: A system and method for using networked mobile devices in a vehicle m a tightly integrated manner is presented. The vehicle has an OBE, a mobile device client, and vehicle components, and the mobile device has a mobile device proxy and applications, such that the mobile device client and the mobile device proxy communicate, enabling dynamic transfer of the applications to the OBE and execution of the applications on the mobile device and the OBE using the plurality of vehicle components at runtime. In one embodiment, the mobile device client and the mobile device proxy authenticate each other. The authentication can be performed using digital certificates. The mobile device client can communicate the vehicle components on the vehicle to the mobile device proxy. The mobile device client and the mobile device proxy can communicate using Bluetooth. The vehicle components can include dashboard displays, speakers, and voice I/O systems.

WO 2010/019568 A1

# SYSTEM AND METHOD FOR USING NETWORKED MOBILE DEVICES IN VEHICLES

## CROSS REFERENCE TO RELATED APPLICATIONS

**[0001]** The present invention claims the benefit of U.S. provisional patent application 61/188,564 filed August 11, 2008, the entire contents and disclosure of which are incorporated herein by reference as if fully set forth herein.

## FIELD OF THE INVENTION

**[0002]** The present invention relates generally to wireless networks.

## BACKGROUND OF THE INVENTION

**[0003]** The in-vehicle use of mobile, handheld network devices, e.g., mobile telephones, portable mp3 players, personal digital assistants, etc., has dramatically increased over the past few years. However, it is difficult and even dangerous to use such devices in the car. To interact and control his or her device, the user currently has little choice but to learn a new user interface (UI) system in the car, or resort to using the display and controls on the device, which are often too small and awkward for in-car use and present a serious safety hazard.

**[0004]** In the area of in-vehicle network computing, Ford Sync allows use of voice commands. In some cars, the user can use the car's audio controls to control the media playing of certain portable MP3 players. For example, in Ford Sync, either application programs execute directly on mobile devices or application data is transferred to the vehicle to be used by the vehicle's application programs. Therefore, the user has to learn the vehicle's native UI mechanism.

**[0005]** In the area of general network computing, both Sun's JINI and UPnP allow dynamic discovery of functional capabilities of various components in a given computing environment. In turn, this feature allows a new device that has not been previously known to the computing environment to use the capabilities of the other devices and make its capability available to other devices. In JINI and UPnP, the application interface and/or driver software is moved from the host network to a new device that connects to the network, and the device runs the received software in order to use the services of the host network. In both UPnP and JINI, the user should still employ the UI controls of the mobile device to use the services of vehicle components, which is inconvenient and dangerous.

[0006] As a result, there is a growing need to better support vehicle drivers and passengers to seamlessly and intuitively use networked handheld devices in the car.

## SUMMARY OF THE INVENTION

[0007] A novel method for allowing application services of mobile devices to run on vehicles by way of dynamic capability discovery and application program transfer at runtime is presented. Currently, in-vehicle dashboard screens, speakers, and other components are designed to support in-vehicle applications only. In general, the vehicle is regarded as a closed system that provides a fixed (or static) set of application services and static UI controls. The present invention provides a novel way of "opening up" the vehicle's system so that it can allow a wide variety of third-party applications to use its display and other devices on an as-needed basis, by dynamically moving application software code from mobile devices to the vehicle at runtime. Thus, in the inventive system and method, application software of the mobile device is transferred from the device to the vehicle to run on the vehicle.

[0008] An inventive system and method for using networked mobile devices in a vehicle is presented. The vehicle has an in-vehicle computer, generally known in the art as an OBE, a mobile device client, and a plurality of vehicle components, and the mobile device has a mobile device proxy and one or more applications, such that the mobile device client and the mobile device proxy communicate, enabling dynamic transfer of the one or more applications to the OBE and execution of the one or more applications on the mobile device and the OBE using the plurality of vehicle components at runtime. The mobile device client authenticates the mobile device proxy and the mobile device proxy authenticates the mobile device client. The authentication can be performed using digital certificates. The mobile device client communicates the vehicle components on the vehicle to the mobile device proxy. The mobile device client and the mobile device proxy can communicate using Internet Protocol (IP) over Bluetooth. The vehicle components can include dashboard displays with or without user interface (UI) mechanisms, speakers, and voice I/O systems.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The invention is further described in the detailed description that follows, by reference to the noted drawings by way of non-limiting illustrative embodiments of the invention, in which like reference numerals represent similar parts throughout the

drawings. As should be understood, however, the invention is not limited to the precise arrangements and instrumentalities shown. In the drawings:

Figure 1 is an architectural overview of the inventive system;

Figure 2 is a schematic of vehicle on-board equipment in an embodiment of the inventive system;

Figure 3 is a schematic of a mobile device in an embodiment of the inventive system;

Figure 4 is flow diagram for transferring vehicle capabilities; and

Figure 5 is a flow diagram for transferring application programs.

## DETAILED DESCRIPTION

**[0010]** The inventive system and method enables a networked mobile device to engage a vehicle, permitting the device's user to interface with and control the audio, display, user interface (UI) and other systems of the vehicle by way of dynamically discovering such systems on the vehicle and moving application code from the mobile device to the vehicle on demand.

**[0011]** Figure 1 presents an architectural overview of the present invention. In Figure 1, a vehicle is equipped with an in-vehicle system, called On-Board Equipment (OBE) **10**. For all the practical purposes of the present invention, the OBE **10** is a computer system specifically built for in-vehicle use. OBE **10** may consist of a vehicle dashboard display system **12**, which provides a graphical display area for vehicle applications and touch-screen capabilities for getting input from vehicle occupants. It may also provide a voice input / output (IO) system **14** that allows the user to issue verbal commands to interact with vehicle applications. It may also include one or more other UI system(s) **16**, such as soft keys, buttons, for instance on the steering wheel, or other areas of the vehicle, a mouse, a keyboard, and/or an optical device that tracks where the driver is focusing with respect to the vehicle dashboard display **12**.

**[0012]** While the OBE **10** in Figure 1 is shown as a single device, its functions and interfaces may be implemented in a distributed manner on multiple physical devices. Like a general-purpose computer, the OBE **10** runs an operating system **18**, such as Linux, that provides a runtime environment for vehicle applications and special software, called drivers, which vehicle applications can use to interact with vehicle IO, audio, and other

systems. The concepts of operating systems, drivers, applications, and use of drivers are well understood in the field of the present invention.

**[0013]** As shown in Figure 1, a special-purpose software, called the Mobile Device Client (MDC) **20**, is installed and executes on the OBE **10**. As the name implies, the MDC **20** is a client-side application that works with a server-side software of the present invention, called the Mobile Device Proxy (MDP) **22**, to tightly integrate the vehicle's OBE **10** and the mobile device **24**. The MDC **20** operates to let the MDP **22** know about the vehicle's dashboard display **12**, voice I/O **14**, and other systems **16** and to move and execute one or more applications from the user's mobile device **24**. In one embodiment, the MDC **20** is pre-installed on the OBE **10** of a vehicle when the vehicle is manufactured. However, other methods are also feasible, including a vehicle service station installing the MDC **20** on the OBE **10**, for example via an USB interface, upon the customer's request subsequent to vehicle manufacture and purchase.

**[0014]** As shown in Figure 1, the OBE **10** is capable of communicating with the user's mobile device **24** via a network connection **26**. Specifically, the OBE **10** can communicate with the mobile device **24** via a Bluetooth connection, and the MDC **20** on the OBE **10** communicates with the MDP **22** on the mobile device **24** using a TCP/IP connection over Bluetooth. Implementing applications that use TCP/IP over Bluetooth is well-understood in the field of this invention. Other wireless technologies may also be used to interconnect the OBE **10** and the mobile device **24**. Other transport layer protocols beyond TCP, such as UDP, may be also used for the MDP **22** and MDC **20** to communicate with each other.

**[0015]** In Figure 1, the user's mobile device **24** consists of the operating system (OS) **28**, storage device **30**, and a number of wireless communications capabilities **32**. As discussed above, the mobile device **24** runs a special-purpose application, called the Mobile Device Proxy (MDP) **22**, which works with the MDC **20** on the OBE **10** to run the applications from the mobile device **24** on the OBE **10** of the vehicle by means of dynamic discovery of the vehicle's capabilities and dynamic transfer of application code from the mobile device **24** to the vehicle OBE **10**.

**[0016]** The inventive methods of using special-purpose applications, i.e., MDC **20** and MDP **22**, on the vehicle and mobile device **24**, respectively, to communicate vehicle system information to the mobile device **24** and move application code from the mobile device **24** to the vehicle to execute thereon are unique in the field of mobile computing in

a vehicle environment. Advantageously, in the present invention, all the applications on the mobile device **24** can be available to run on the vehicle (assuming that the vehicle provides necessary device drivers that the applications need), while in prior art approaches, the user is limited to only those applications provided by the vehicle. For example, in other approaches, if the user has an application, e.g., APP-A, on the mobile device **24**, but the vehicle does not have a functionally-similar application that can process APP-A's data and/or provide appropriate drivers for APP-A, the user cannot use APP-A on the vehicle. However, the present invention allows the user to use APP-A on such a vehicle by moving the executable application code and/or data of APP-A from the mobile device **24** to the vehicle, and executing the code with the data thereon. Note in this example that APP-A is provided by the mobile device manufacturer. Therefore, when running on the vehicle, APP-A can provide a similar look-and-feel to when it runs on the mobile device **24**, which can dramatically reduce the learning curve of the user for using APP-A on the vehicle.

**[0017]** Figure 2 illustrates the MDC **20** implemented on the OBE **10** in an embodiment of the present invention. In Figure 2, the OBE **10** provides an application runtime environment, called the OSGi Framework. OSGi Framework is an industry standard technology for executing and managing the lifecycle of a particular type of Java applications, called OSGi Bundles. The OSGi Framework and Bundles have been used to implement a variety of vehicle applications for the OBE by the Vehicle Infrastructure Integration Consortium (also called IntelliDrive). The concepts of the OSGi Framework and Bundles are well understood by those who practice in the field of the present invention.

**[0018]** As shown in Figure 2, the MDC **20** is implemented as an OSGi Bundle in the preferred embodiment of the present invention. This way, the MDC **20** can take advantage of the default system services provided by the OSGi Framework to realize its functions, including discovering (Java) application packages and services exported by other OSGi Bundles on the OSGi Framework. For example, one of the standard services available on every OSGi Framework is the Device Access service, which allows other applications to programmatically discover the devices and the corresponding drivers available on the OSGi Framework and bind them to use the discovered devices in their operations. Therefore, the MDC **20** can use the OSGi Device Access service to keep track of available devices and drivers on the vehicle at runtime. Likewise, the MDC **20** can also

use the OSGi Service Registry, which is another standard service on every OSGi Framework, to keep track of services and application packages exported by other Bundles on the OSGi Framework.

**[0019]** As an OSGi Bundle, the MDC **20** also has the ability to dynamically install, start, and stop other Bundles at runtime by means of the application programming interfaces (APIs) of OSGi BundleContext and Bundle objects. Specifically, upon receiving an application bundle from the MDP **22**, described below, the MDC **20** first stores the received Bundle in the file system of the vehicle OBE **10**. To install the received Bundle on the vehicle, the MDC **20** then uses the installBundle() method of the BundleContext object provided by the vehicle's OSGi Framework; it passes the URL to the location of the received Bundle on the file system to installBundle(). Subsequently, it invokes the start() method on the Bundle object that is returned by installBundle() to start the newly installed Bundle. Likewise, the MDC **20** can use the stop(), update(), and uninstall() methods of the same Bundle object to stop the operation of, update the software of, and uninstall the received Bundle, respectively.

**[0020]** In Figure 3, the MDP **22** is shown as an application that runs in the operating system (OS) **28** of the mobile device **24**. Unlike the MDC **20**, which is an OSGi Bundle (in Java), the MDP **22** is an native application which is specifically implemented to run in the OS **28** of the mobile device **24**. As shown in Figure 3, the MDP **22** manages a set of OSGi Bundles that can be sent to the vehicle OBE **10** and can execute there. For each OSGi Bundle, the MDP **22** keeps track of dependencies the OSGi Bundle may have on other applications. In one embodiment, the dependency includes the names of packages (in Java) the OSGi uses in its operation. It may also include version information for each dependent package. MDP **22** uses the dependency information to determine if a given OSGi Bundle stored in storage **30** on the mobile device **24** can run on the vehicle OBE **10**.

**[0021]** In the embodiment in Figure 2 and Figure 3, the MDC **20** and MDP **22** use the TCP/IP over Bluetooth **34** to communicate with the MDP **22** on the mobile device **24**. Specifically, the MDC **20** periodically broadcasts a "Hello" message to a well-known port number, to which the MDP **22** listens for MDC **20** messages. Upon receiving the message from the MDC **20**, the MDP **22** responds by sending its own "Hello" message to another well-known port number, to which the MDC **20** listens for MDP **22** messages. This response message from the MDP **22** to the MDC **20** is addressed to the IP address that the MDP **22** retrieves from the IP header of MDC's broadcast message. The response

message also includes a port number, to which the MDP **22** listens for further communications with the MDC **20**, and an identifier, which uniquely identifies the MDP's mobile device **24**. UDP can be used both for the broadcast messages from the MDC **20** to the MDP **22** and for the response message from the MDP **22** to the MDC **20**. Alternatively, TCP, or other appropriate communication techniques, can be used for communicating these messages between the MDC **20** and MDP **22**.

**[0022]** Figure 2 and Figure 3 show both the MDC **20** and MDP **22** having asymmetric cryptographic keys and their digital certificates **36**, which are used to perform mutual authentication and secure the communication, as described in the present disclosure, between the MDC **20** and MDP **22**. The mechanisms for installing, distributing, updating, and revoking digital certificates **36** are well understood in the field of the present invention. In one embodiment of the present invention, the MDC **20** digitally signs its "Hello" broadcast message, which allows the MDP **22** to validate the signature of the received the "Hello" message and authenticate the MDC **20**. Likewise, the MDP **22** digitally signs its "Hello" response message, which allows the MDC **20** to validate the signature of the received "Hello" response message and authenticate the MDP **22**. The algorithms for signing messages and verifying signatures with digital signatures are well-known in the field of the present invention. Data used for creation, exchange, or use of symmetric cryptographic encryption and integrity protection of transfer of application code and data can also be communicated between the MDC **20** and MDP **22** by way of having the MDC **20** and MDP **22** run TLS, DTLS, or shared key management protocols. TLS and DTLS are industry standard communications protocols for establishing and transferring data in secure sessions and are well understood in the field of the present invention.

**[0023]** Operation of the inventive system is now described. The MDC **20** executes to send a list of vehicle capabilities, in terms of application packages exported by OSGi Bundles on the vehicle, and to receive and run OSGi Bundles from the MDP **22** on the mobile device **24**. Once the MDC **20** detects the MDP **22**, as described above, the MDC **20** attempts to determine if the detected MDP **22** is legitimate, that is MDC **20** performs authentication regarding MDP **22**. In one embodiment, the MDP **22** can digitally sign its "Hello" response message to the MDC **20**, which, in turn, validates the signature before proceeding.

**[0024]** If the MDC **20** successfully authenticates the MDP **22**, MDC **20** sends to the MDP **22** the list of vehicle capabilities, such as application packages and services exported by the OSGi Bundles on the vehicle OBE **10**. The list is contained in a message, called "CAP", addressed to the IP address that the MDC **20** retrieves from the IP header of the MDP "Hello" response message and the port number included in the same "Hello" response message. The "CAP" message also includes the port number, to which the MDC **20** listens for potential file transfer of OSGi Bundles from the MDP **22**, and the location information on the file system of the OBE **10** where the OSGi Bundles are to be stored.

**[0025]** A File Transfer Protocol, e.g., FTP or secure FTP, can be used to transfer any OSGi Bundle files from the mobile device **24** to the vehicle, in which the MDC **20** plays the role of an FTP server and the MDP **22** the role of an FTP client. Other approaches are also feasible, including a well-known file transfer over Secure Shell (SSH), HTTP, or HTTPS, which uses the Transport Layer Security. UDP transfers can also be supported using Datagram Transport Layer Security.

**[0026]** After sending the "CAP" message, the MDC **20** waits to receive an acknowledgement message from the MDP **22** that it has received the "CAP" message. If the acknowledgement is received within a (configurable) period of wait time, the MDC **20** proceeds to listen for a FTP connection request from the MDP **22** at the port number included in the "CAP" message. Otherwise, the MDC **20** sends the "CAP" message again and waits for the acknowledgement. If the MDC **20** still does not receive the acknowledgement after a (configurable) number of retries, MDC **20** proceeds to wait for the current mobile device **24** to disconnect from the vehicle, the process of which is described below.

**[0027]** When the MDC **20** receives an FTP connection request from the MDP **22** within a (configurable) period of wait time, MDC **20** proceeds to execute the FTP protocol with the MDP **22** to receive application programs, that is, OSGi Bundles, from the mobile device **24**. Subsequent to the successful transfer of the OSGi Bundles, the MDC **20** stores them on the file system of the vehicle OBE **10** and installs and starts them on the OSGi Framework of the vehicle. If the FTP transfer fails, the MDC **20** retries the process of waiting for FTP connection requests from the MDP **22** and executing the FTP protocol. If the FTP transfer still fails after a (configurable) number of retries, MDC **20** proceeds to wait for the current mobile device **24** to disconnect from the vehicle.

**[0028]** It may happen that the MDC **20** does not receive any OSGi bundles after a (configurable) number of retries to wait for FTP connection requests from the MDP **22**. In such a case, MDC **20** assumes that the present mobile device **24** does not have any applications to run on the vehicle and waits for the mobile device **24** to disconnect from the vehicle. The MDP **22** can also send the MDC **20** a message indicating that no applications are available.

**[0029]** MDC **20** detects the disconnect of the mobile device **24** by periodically sending "Still There" messages to the same port number used for the "Hello" message broadcast. If MDP **22** does not respond with the corresponding "Still Here" messages within a (configurable) period of time for a (configurable) number of retries, the MDC **20** decides that the mobile device **24** has been disconnected from the vehicle. In such a case, the MDC **20** stops any running OSGi Bundles that have been received from the mobile device **24** and goes back to waiting for new MDPs.

**[0030]** Figure 4 is a flow diagram of the algorithm that the MDC **20** executes to send the list of vehicle capabilities and to receive and process OSGi Bundles from the MDP **22** on the mobile device **24**. In step S1, MDC **20** determines whether MDP **22** is detected. If no MDP **22** is detected (S1=NO), MDC **20** continues to try to detect MDP **22**, repeating step S1.

**[0031]** If MDP **22** is detected (S1=YES), in step S2, MDP **22** is authenticated, as discussed above. If MDP is not authentic or legitimate (S3=NO), then processing returns to step S1.

**[0032]** If MDP **22** is legitimate (S3=YES), in step S4 MDC **20** sends vehicle capabilities to this legitimate MDP **22**. MDC **20** waits to receive application programs from MDP **22** in step S5. In step S6, MDC **20** checks for acknowledgement of a "CAP" message. If the message is acknowledged (S6=YES), MDC **20** receives and runs the application programs from MDP **22** in step S7. When the programs are run (in S7) or when no acknowledgement is received (S6=NO), MDC **20** determines whether MDP **22** is disconnected in step S8. If MDP **22** is disconnected (S8=YES), programs from MDP **22** are stopped in step S9. If MDP is not disconnected (S8=NO), MDC **20** continues to check whether MDP **22** is disconnected in step S8.

**[0033]** Operation of the MDP **22** on the mobile device **24**, which runs to transfer applications to the MDC **20** on the vehicle, is as follows. Upon receiving the "Hello" message from the MDC **20**, the MDP **22** validates the signature on the message to

authenticate the MDC **20**. If successful, MDP **22** sends its "Hello" response message as described previously.

**[0034]** Subsequently, MDP **22** waits for the "CAP" message from the MDC **20**. If the message is not received for a (configurable) period of wait time after a (configurable) time of retries, the MDP **22** goes back to waiting for and processing the "Hello" messages. Otherwise, the MDP **22** sends an acknowledgement for the receipt of the "CAP" message, retrieves the list of application packages and services from the received "CAP" message, and tries to identify OSGi Bundles that can execute on the information, if applicable, of the application packages and services available on the vehicle.

**[0035]** If an OSGi Bundle for an application that can run on the vehicle exists, the MDP **22** prepares to send the Bundle to the MDC **20**. Such preparation includes configuring and running some associated components of the application on the mobile device **24**. For example, a text messaging application may exist on a mobile phone **24** whose OSGi Bundle consists of a software component designed to use the vehicle's voice recognition system to allow the user to verbally dictate text messages for a given recipient and a graphical user interface (GUI) component to display text messages being composed on the vehicle's dashboard display. When the user says "Send Message," yet another component of the text messaging Bundle sends the composed message and the recipient information to the mobile device **24**, e.g., on the IP over Bluetooth connection **34**, which, in turn, sends the message out to the recipient using the message transmission mechanism of the mobile device **24**.

**[0036]** This type of an application requires that some application components run on the mobile device **24** at the same time as the components execute on the vehicle. In the above example of the described text messaging application, some application components should run on the mobile device **24** that receive the composed messages from the vehicle and send the messages to recipients. To support this type of an application, the MDP **22** maintains a configuration file which identifies, for each OSGi Bundle, the corresponding application components and any configuration and initialization data for them. If an OSGi Bundle can execute on the vehicle, the MDP **22** first configures and runs the corresponding application components on the mobile device **24** and then transfers the OSGi Bundle to the MDC **20**.

**[0037]** Once all the OSGi Bundles have been transferred to the MDC **20**, the MDP **22** waits until the mobile device **24** is disconnected from the vehicle. It does so by listening

for and responding to the "Still There" messages from the MDC **20**. When disconnected, the MDP **22** stops all the application components that have been running in support of the OSGi Bundles that have been transferred to the vehicle and goes back to waiting for the "Hello" message from the MDC **20**.

**[0038]** Figure 5 is a flow diagram of the algorithm the MDP **22** on the mobile device **24** runs to transfer applications to the MDC **20** on the vehicle. In step S10, MDP **22** determines whether MDC **20** is detected, that is, whether a "Hello" message has been received. If no MDC **20** is detected (S10=NO), MDP **22** continues to listen for a message, repeating step S10.

**[0039]** If MDC **20** is detected (S10=YES), it is authenticated in step S11 using the procedure described above. If the detected MDC **20** is legitimate (S12=YES), then MDP **22** waits to receive vehicle capabilities from MDC **20** in step S13. Otherwise, if MDC **20** is not legitimate (S12=NO), MDP **22** returns to step S10 and listens to detect another "Hello" message.

**[0040]** In step S14, MDP **22** determines whether the vehicle capabilities have been received from the MDC **20**. If the vehicle capabilities have not been received (S14=NO), processing returns to step S10 and MDP **22** listens for another "Hello" message.

**[0041]** If vehicle capabilities have been received (S14=YES), MDP **22** determines whether any application programs match these vehicle capabilities, in step S15. If not, processing returns to step S10 and MDP **22** listens for another "Hello" message.

**[0042]** If there are one or more application programs that match vehicle capabilities (S15=YES), then, in step S16, MDP **22** runs any application programs that correspond to those being sent to MDC **20**. In step S17, MDP **22** sends to MDC **20** any application programs that match vehicle capabilities.

**[0043]** In step S18, MDP **22** determines whether MDC **20** is disconnected. If MDC **20** is not disconnected (S18=NO), MDP **22** continues to check whether MDC **20** is disconnected by repeating step S18. If MDP **22** determines that MDC **20** is disconnected (S18=YES), application programs in MDP **22** corresponding to those in MDC **20** are stopped in step S19.

**[0044]** Several extensions on the described method are possible. For example, the communication between the MDC **20** and MDP **22** can be secured by employing an industry-standard secure transport protocol such as TLS or DTLS as previously mentioned. In addition, the MDC **20** can communicate with multiple mobile devices **24** at

the same time. This is feasible by keeping track of the IP address of each mobile device **24** and the associated identifier information in the "Hello" response message from the MDP **22**.

**[0045]** To facilitate faster MDP **22** and MDC **20** interactions after the initial encounter and download of applications between the vehicle and mobile device, the MDC **20** may keep a cache of applications on the OBE **10**. The OSGi Framework allows Java applications to reside in the framework without starting, for instance, between vehicle uses. During the next interaction between the MDP **22** and MDC **20**, such as the next time the driver enters the vehicle, the MDC **20** will send the MDP **22**, in the "Hello" response message, a list of available applications associated with that mobile device **24** that are already in the OBE **10**. The MDC **20** can use the MDP's digital certificate **36** to identify the mobile device **24** and associate applications with it.

**[0046]** The list of available applications sent to the MDP **22** can also include information to identify the application version. If the application that the mobile device **24** wishes to run is already installed on the OBE **10**, but not running, the MDP **22** need only identify which application the MDC **20** should start and avoid an unnecessary download. It can do so by sending a "START" message to the MDC **20** identifying a list of applications to start after running any corresponding application components on the mobile device **24** as previously described. Receipt of the "START" message will cancel the MDC **20** wait for an FTP connection request from the MDP **22**.

**[0047]** If the MDP **22** determines that the MDC's application is out of date based on the application version, MDP **22** will connect via FTP and transfer the new application with instructions to perform an OSGi Update of the application's Bundle. In the interactions above and previously described, each application is signed, which helps the MDC **20** to establish authenticity and determine if any alterations of the application have occurred during the download process.

**[0048]** Further, the described interactions between the MDC **20** and MDP **22** can occur anytime the mobile device **24** is within the radio range of the vehicle whether the vehicle has not started, is starting up, or has started. In the preferred embodiment of the invention, the vehicle has a wireless key device, and the MDC **20** begins to listen for MDPs and responds only after the vehicle has successfully authenticated a user who has the wireless key device. This can be achieved if the vehicle provides a service on the OSGi

Framework that publishes the vehicle lock and unlock status to other applications on the same OSGi Framework, and the MDC **20** subscribes to the service.

**[0049]** Various aspects of the present disclosure may be embodied as a program, software, or computer instructions embodied in a computer or machine usable or readable medium, which causes the computer or machine to perform the steps of the method when executed on the computer, processor, and/or machine. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform various functionalities and methods described in the present disclosure is also provided.

**[0050]** The system and method of the present disclosure may be implemented and run on a general-purpose computer or special-purpose computer system. The computer system may be any type of known or will be known systems and may typically include a processor, memory device, a storage device, input/output devices, internal buses, and/or a communications interface for communicating with other computer systems in conjunction with communication hardware and software, etc.

**[0051]** The terms "computer system" and "computer network" as may be used in the present application may include a variety of combinations of fixed and/or portable computer hardware, software, peripherals, and storage devices. The computer system may include a plurality of individual components that are networked or otherwise linked to perform collaboratively, or may include one or more stand-alone components. The hardware and software components of the computer system of the present application may include and may be included within fixed and portable devices such as desktop, laptop, and server. A module may be a component of a device, software, program, or system that implements some "functionality", which can be embodied as software, hardware, firmware, electronic circuitry, or etc.

**[0052]** The embodiments described above are illustrative examples and it should not be construed that the present invention is limited to these particular embodiments. Thus, various changes and modifications may be effected by one skilled in the art without departing from the spirit or scope of the invention as defined in the appended claims.

What is claimed is:

1. A system for using networked mobile devices in a vehicle,

said vehicle comprising:

an in-vehicle computer;

a mobile device client; and

a plurality of vehicle components; and

said mobile device comprising:

a mobile device proxy; and

one or more applications and data;

wherein the mobile device client and the mobile device proxy communicate, enabling dynamic transfer of the one or more applications and the data to the in-vehicle computer and execution of the one or more applications on the mobile device and the in-vehicle computer using the plurality of vehicle components at runtime.

2. The system according to claim 1, wherein the mobile device client authenticates the mobile device proxy and the mobile device proxy authenticates the mobile device client.

3. The system according to claim 1, wherein the mobile device client communicates the plurality of vehicle components on the vehicle to the mobile device proxy.

4. The system according to claim 1, wherein the mobile device client communicates information about a plurality of applications cached on the vehicle to the mobile device proxy.

5. The system according to claim 1, wherein the mobile device proxy determines a plurality of applications to transfer to the vehicle.

6. The system according to claim 1, wherein the mobile device proxy dynamically configures a plurality of applications to be transferred to the vehicle.

7. The system according to claim 1, wherein the mobile device client and the mobile device proxy communicate using at least one of a Bluetooth connection, an Internet Protocol (IP) connection, and a secure connection.

8. The system according to claim 1, wherein the mobile device client and the mobile device proxy communicate to dynamically create cryptographic keys for encryption, integrity protection, and other security protection of the communications, or to use existing cryptographic keys for encryption, integrity protection, and other security protection of the communications.

9. The system according to claim 1, wherein the mobile device client detects that the mobile device proxy has disconnected and the mobile device proxy detects that the mobile device client has disconnected.

10. The system according to claim 1, wherein the mobile device client and the mobile device proxy communicate while the vehicle has not started up, is starting up, or has started up.

11. The system according to claim 1, wherein the plurality of vehicle components comprises one or more of dashboard displays with or without built-in user interface mechanisms, speakers, and voice I/O systems.

12. The system according to claim 1, wherein one mobile device client communicates with a plurality of mobile device proxies at a same time.

13. A method for using networked mobile devices in a vehicle, comprising steps of:

communicating between a mobile device client on the vehicle and a mobile device proxy on the mobile device;

dynamically transferring one or more applications and data from the mobile device to the vehicle; and

executing at runtime one or more applications on the mobile device and on the vehicle using a plurality of vehicle components on the vehicle.

14. The method according to claim 13, further comprising:

authenticating the mobile device client by the mobile device proxy; and

authenticating the mobile device proxy by the mobile device client.

15. The method according to claim 13, wherein the step of communicating further comprises sending the plurality of vehicle components from the vehicle to the mobile device proxy.

16. The method according to claim 13, wherein the step of communicating further comprises sending information about a plurality of applications cached on the vehicle to the mobile device proxy.

17. The method according to claim 13, further comprising a step of determining a plurality of applications to transfer to the vehicle.

18. The method according to claim 13, further comprising a step of dynamically configuring a plurality of applications to be transferred to the vehicle.

19. The method according to claim 13, wherein the step of communicating is performed using at least one of a Bluetooth connection, an Internet Protocol (IP) connection, and a secure connection.

20. The method according to claim 13, the step of communicating further comprising one of:

dynamically creating cryptographic keys for encryption, integrity protection, and other security protection of the communications; and

using existing cryptographic keys for encryption, integrity protection, and other security protection of the communications.

21. The method according to claim 13, further comprising:

detecting disconnection of the mobile device proxy by the mobile device client; and

detecting disconnection of the mobile device client by the mobile device proxy.

22. The method according to claim 13, wherein the step of communicating occurs while the vehicle has not started up, is starting up, or has started up.

23. The method according to claim 13, wherein the plurality of vehicle components comprises one or more of dashboard displays with or without built-in user interface mechanisms, speakers, and voice I/O systems.

24. The method according to claim 13, wherein one mobile device client communicates with a plurality of mobile device proxies at a same time.

25. A computer readable medium having computer readable program for operating on a computer for using networked mobile devices in a vehicle, said program comprising instructions that cause the computer to perform steps of:

communicating between a mobile device client on the vehicle and a mobile device proxy on the mobile device;

dynamically transferring one or more applications and data from the mobile device to the vehicle; and

executing at runtime one or more applications on the mobile device and on the vehicle using a plurality of vehicle components on the vehicle.

26. The program according to claim 25, further comprising:

authenticating the mobile device client by the mobile device proxy; and

authenticating the mobile device proxy by the mobile device client.

27. The program according to claim 25, wherein the step of communicating further comprises sending the plurality of vehicle components from the vehicle to the mobile device proxy.

28. The program according to claim 25, wherein the step of communicating further comprises sending information about a plurality of applications cached on the vehicle to the mobile device proxy.

29. The program according to claim 25, further comprising a step of determining a plurality of applications to transfer to the vehicle.

30. The program according to claim 25, further comprising a step of dynamically configuring a plurality of applications to be transferred to the vehicle.

31. The program according to claim 25, wherein the step of communicating is performed using at least one of a Bluetooth connection, an Internet Protocol (IP) connection, and a secure connection.

32. The program according to claim 25, the step of communicating further comprising one of:

dynamically creating cryptographic keys for encryption, integrity protection, and other security protection of the communications; and

using existing cryptographic keys for encryption, integrity protection, and other security protection of the communications.

33. The program according to claim 25, further comprising:

detecting disconnection of the mobile device proxy by the mobile device client; and

detecting disconnection of the mobile device client by the mobile device proxy.

34. The program according to claim 25, wherein the step of communicating occurs while the vehicle has not started up, is starting up, or has started up.

35. The program according to claim 25, wherein the plurality of vehicle components comprises one or more of dashboard displays with or without built-in user interface mechanisms, speakers, and voice I/O systems.

36. The program according to claim 25, wherein one mobile device client communicates with a plurality of mobile device proxies at a same time.

FIG. 1

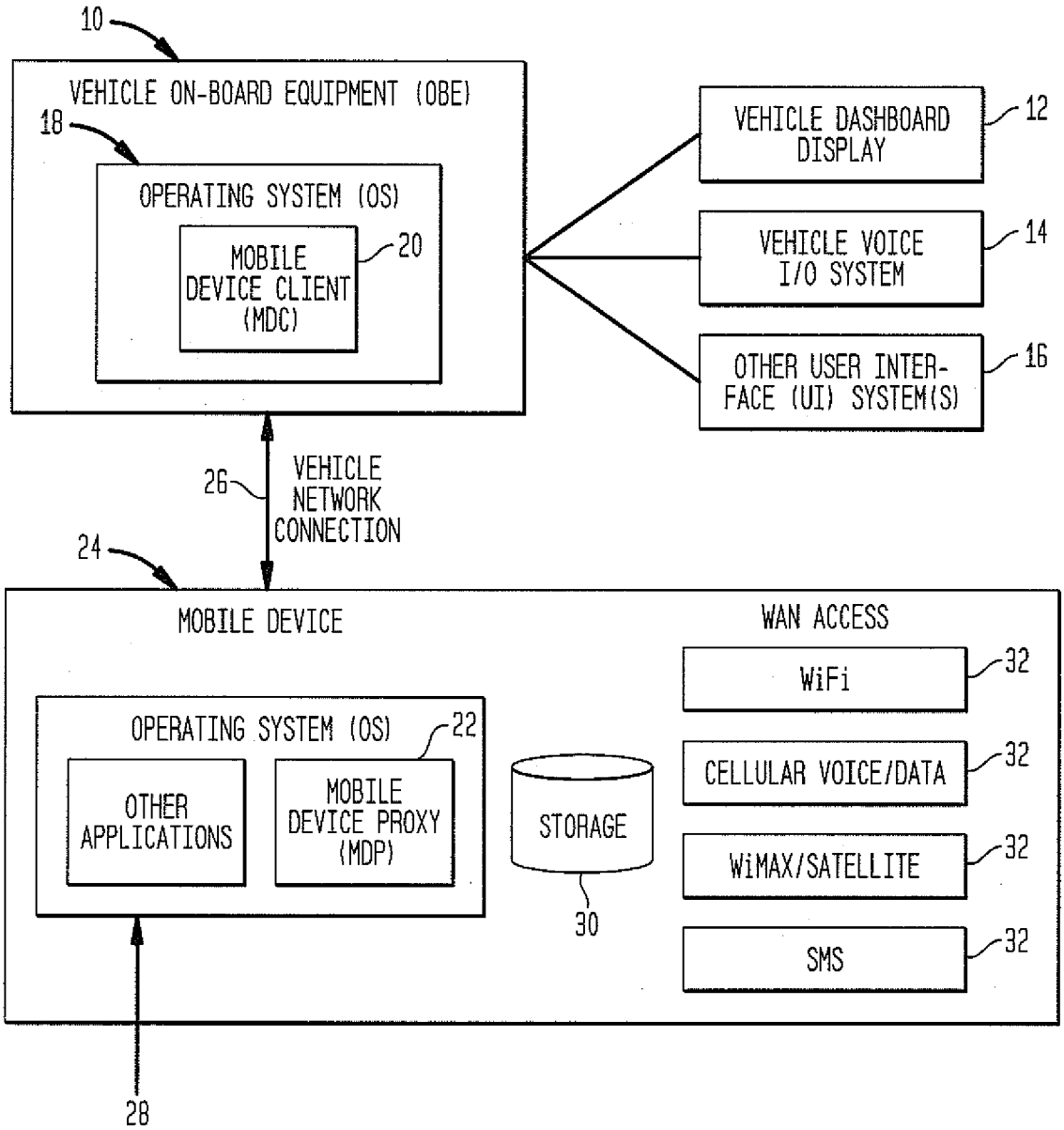

10
VEHICLE ON-BOARD EQUIPMENT (OBE)
18
OPERATING SYSTEM (OS)
MOBILE DEVICE CLIENT (MDC)
20
VEHICLE DASHBOARD DISPLAY
12
VEHICLE VOICE I/O SYSTEM
14
OTHER USER INTER-FACE (UI) SYSTEM(S)
16
26
VEHICLE NETWORK CONNECTION
24
MOBILE DEVICE
WAN ACCESS
WiFi
32
CELLULAR VOICE/DATA
32
WiMAX/SATELLITE
32
SMS
32
OPERATING SYSTEM (OS)
22
OTHER APPLICATIONS
MOBILE DEVICE PROXY (MDP)
STORAGE
30
28

FIG. 2

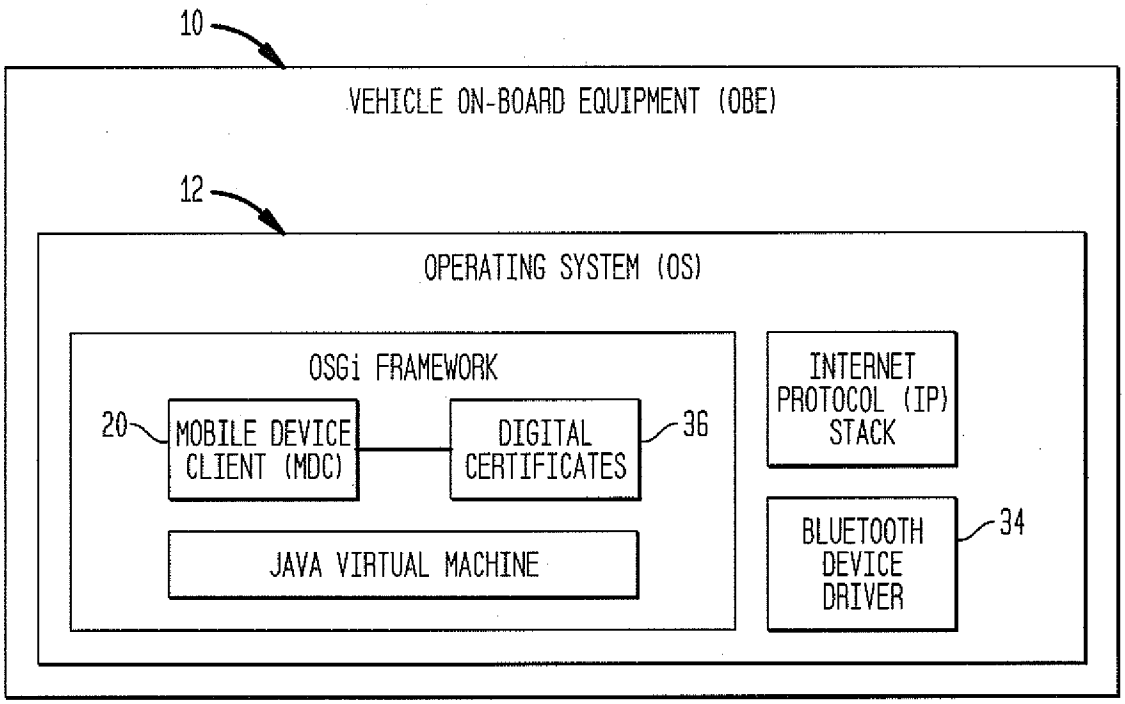

10
VEHICLE ON-BOARD EQUIPMENT (OBE)
12
OPERATING SYSTEM (OS)
OSGi FRAMEWORK
20
MOBILE DEVICE CLIENT (MDC)
DIGITAL CERTIFICATES
36
JAVA VIRTUAL MACHINE
INTERNET PROTOCOL (IP) STACK
BLUETOOTH DEVICE DRIVER
34

FIG. 3

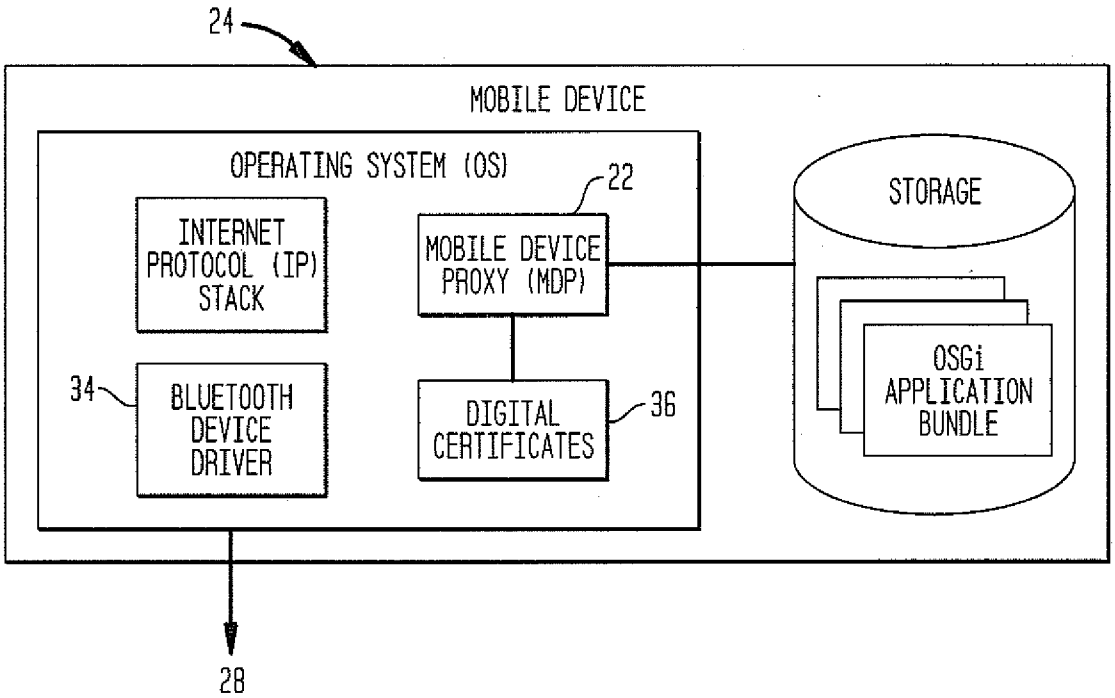

24
MOBILE DEVICE
OPERATING SYSTEM (OS)
22
INTERNET PROTOCOL (IP) STACK
MOBILE DEVICE PROXY (MDP)
STORAGE
OSGi APPLICATION BUNDLE
34
BLUETOOTH DEVICE DRIVER
DIGITAL CERTIFICATES
36
28

FIG. 4

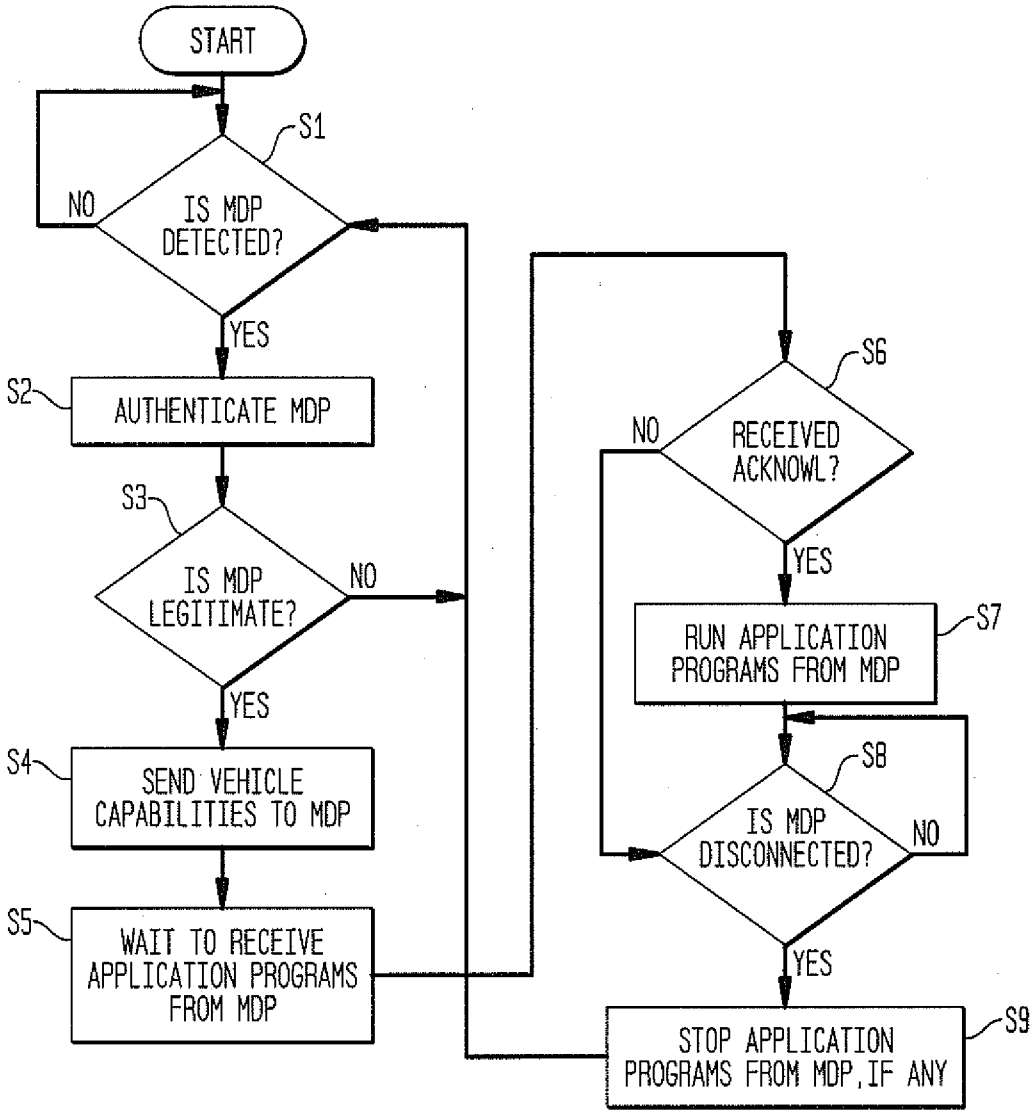
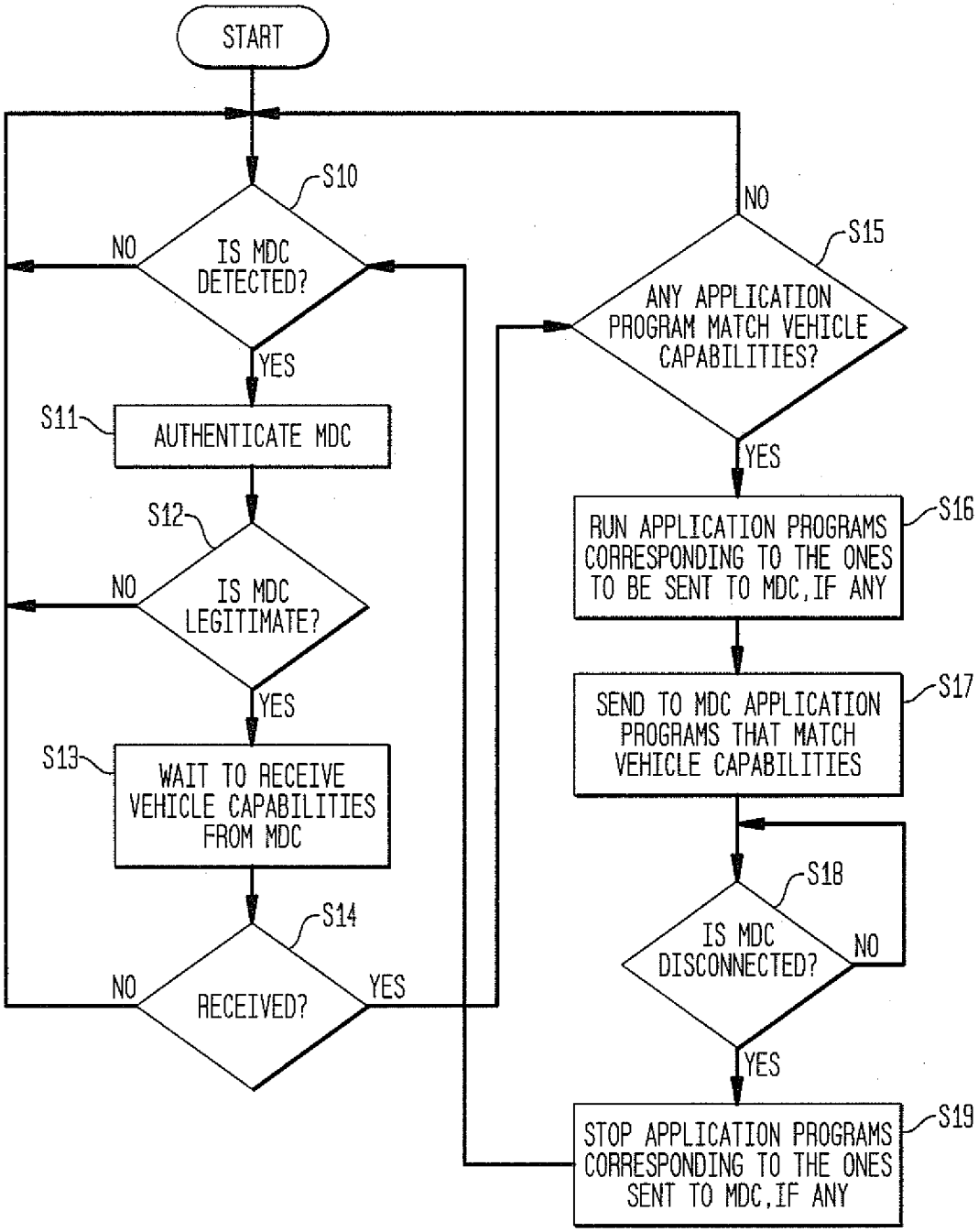

START
S1
IS MDP DETECTED?
NO
YES
S2
AUTHENTICATE MDP
S3
IS MDP LEGITIMATE?
NO
YES
S4
SEND VEHICLE CAPABILITIES TO MDP
S5
WAIT TO RECEIVE APPLICATION PROGRAMS FROM MDP
S6
RECEIVED ACKNOWL?
NO
YES
S7
RUN APPLICATION PROGRAMS FROM MDP
S8
IS MDP DISCONNECTED?
NO
YES
S9
STOP APPLICATION PROGRAMS FROM MDP, IF ANY

FIG. 5


START
S10
IS MDC DETECTED?
NO
YES
S11
AUTHENTICATE MDC
S12
IS MDC LEGITIMATE?
NO
YES
S13
WAIT TO RECEIVE VEHICLE CAPABILITIES FROM MDC
S14
RECEIVED?
NO
YES
NO
S15
ANY APPLICATION PROGRAM MATCH VEHICLE CAPABILITIES?
YES
S16
RUN APPLICATION PROGRAMS CORRESPONDING TO THE ONES TO BE SENT TO MDC, IF ANY
S17
SEND TO MDC APPLICATION PROGRAMS THAT MATCH VEHICLE CAPABILITIES
S18
IS MDC DISCONNECTED?
NO
YES
S19
STOP APPLICATION PROGRAMS CORRESPONDING TO THE ONES SENT TO MDC, IF ANY

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2009/053402

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G01C 21/00 (2009.01)
USPC - 701/210
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - G01C 21/00 (2009.01)
USPC - 340/903, 993; 701/210

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
USPTO WEST System (US, USPG-PUB, EPO), Google Patent, PatBase

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | US 2002/0177928 A1 (MORIGUCHI et al) 28 November 2002 (28.11.2002) entire document | 1-34 |
| Y | US 2007/0063037 A1 (SILVERBROOK et al) 22 March 2007 (22.03.2007) entire document | 1-34 |
| Y | US 2004/0203379 A1 (WITKOWSKI et al) 14 October 2004 (14.10.2004) entire document | 2, 8, 14, 20, 26 and 32 |

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 28 September 2009 | 02 OCT 2009 |
| Name and mailing address of the ISA/US<br>Mail Stop PCT, Attn: ISA/US, Commissioner for Patents<br>P.O. Box 1450, Alexandria, Virginia 22313-1450<br>Facsimile No. 571-273-3201 | Authorized officer:<br>Blaine R. Copenheaver<br>PCT Helpdesk: 571-272-4300<br>PCT OSP: 571-272-7774 |