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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 July 2004 (29.07.2004)

PCT

(10) International Publication Number
WO 2004/063899 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number:
PCT/US2004/000694

(22) International Filing Date: 13 January 2004 (13.01.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/439,673 13 January 2003 (13.01.2003) US

(71) Applicant *(for all designated States except US)*: **BITFONE CORPORATION** [US/US]; 32451 Golden Lantern, Suite 301, Launa Niguel, CA 92677 (US).

(72) Inventors; and

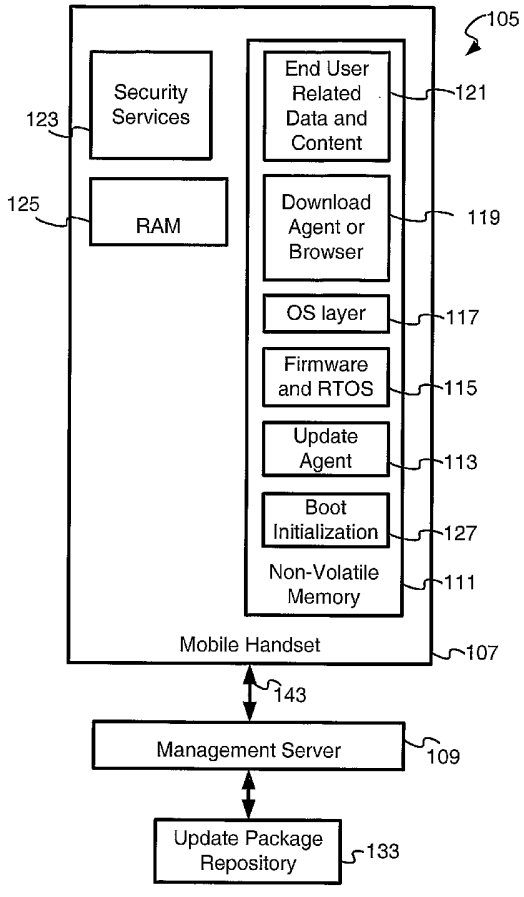
(75) Inventors/Applicants *(for US only)*: **GUSTAFSON, James, P.** [US/US]; 2100 Timberwood, Irvine, CA 92620 (US). **CHEN, Shao-Chun** [US/US]; 27662 Alliso Viejo Creed Road #7304, Alliso Viejo, CA 92656 (US). **PAKARINEN, Toni** [FI/US]; 2 Enterprise #5316, Aliso Viejo, CA 92656 (US). **NGUYEN, Do, P.** [US/US]; 7475 Keisha Terrace, San Diego, CA 92126 (US). **MAROLIA, Sunil** [US/US]; 32 Terra Vista, Data Point, CA 92629 (US). **HAMMERBERG, Karl, W.** [US/US]; 22501 Chase, Aliso Viejo, CA 92656 (US).

(74) Agent: **BORG, Kevin, E.**; 500 West Madison street, Suite 3400, Chicago, IL 60661 (US).

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

[Continued on next page]

(54) Title: MOBILE HANDSET CAPABLE OF UPDATING ITS UPDATE AGENT



(57) Abstract: A mobile handset in a mobile services network, with access to a plurality of services including a firmware / software update service, is also capable of updating its update agent using an appropriate update package. The update package is retrieved from an update package repository via a management server. In one embodiment, the existing update agent is copied to a backup section before the update agent itself is updated. On the subsequent reboot, the new update agent is employed unless it is determined that it is corrupted or ineffective, in which case the old update agent is reactivated.

WO 2004/063899 A2

TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

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

MOBILE HANDSET CAPABLE OF UPDATING ITS UPDATE AGENT

RELATED APPLICATIONS

[0001] This patent application makes reference to, claims priority to and claims benefit from United States Provisional Patent Application Serial No. 60/439,673, entitled "Mobile Handset Capable of Updating its Update Agent," filed on January 13, 2003.

[0002] The complete subject matter of the above-referenced United States Provisional Patent Application is hereby incorporated herein by reference, in its entirety. In addition, this application makes reference to United States Provisional Patent Application Serial No. 60/249,606, entitled "System and Method for Updating and Distributing Information", filed November 17, 2000, and International Patent Application Publication No. WO 02/41147 A1, entitled "Systems And Methods For Updating And Distributing Information," publication date March 23, 2002, the complete subject matter of each of which is hereby incorporated herein by reference, in its entirety.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0003] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0004] [Not Applicable]

BACKGROUND OF THE INVENTION

[0005] Electronic devices, such as mobile phones and personal digital assistants (PDA's), often contain firmware and application software that are either provided by the manufacturers of the electronic devices, by telecommunication carriers, or by third parties. These firmware and application software often contain software bugs. New versions of the firmware and software are periodically released to fix the bugs or to introduce new features, or both.

[0006] Problems may arise when informing a mobile handset of a need to update its firmware or software. Additionally, the mobile handset may utilize an update agent or driver in the update process, and the update agent or driver may also require updating.

Such updates may be complicated and a mobile handset may become inoperative if such an update should fail.

[0007] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of ordinary skill in the art through comparison of such systems with the present invention.

BRIEF SUMMARY OF THE INVENTION

[0008] Aspects of the present invention may be seen in a system that facilitates the updating of at least one of firmware and updating software in an electronic device, using updating information received via a communication network, the system comprising a non-volatile memory; a RAM; and security services for verifying the loaded updating information. The non-volatile memory comprises firmware; loading software for loading the updating information via the communication network; updating software for applying the loaded updating information to the at least one of firmware and updating software; and initializing software for initializing the electronic device. In an embodiment of the present invention, the updating software is capable of updating itself. The updating software is also capable of saving a back up copy of itself when it updates itself. In an embodiment of the present invention, the updating software can determine whether an updating process was successful, and if not, the updating software uses the back up copy of itself.

[0009] A method for updating at least one of firmware and updating software in an electronic device, using updating information received via a communication network, the method comprising initializing the electronic device; and determining whether at least one of the firmware and the updating software is to be updated. If it is determined that no updating needs to be done, the method further comprises performing a normal start up of the electronic device. Otherwise, if it is determined at least one of the firmware and the updating software needs to be updated, the method further comprises determining whether the updating software needs to be updated.

[0010] If it is determined that the updating software does not need updating and the firmware needs updating, the method further comprises updating the firmware using the updating information; and initializing the electronic device. Otherwise, if it is determined that the updating software needs to be updated, the method further comprises backing up the updating software; updating the updating software using the updating information to produce a new updating software; initializing the electronic device; and determining whether the updating of the updating software was successful.

[0011] If it is determined that updating the updating software was successful, the method further comprises enabling the use of the new updating software; and initializing the electronic device. Otherwise, if it is determined that updating the updating software was not successful, the method further comprises using the backed up updating software; and initializing the electronic device.

[0012] These and other features and advantages of the present invention may be appreciated from a review of the following detailed description of the present invention, along with the accompanying figures in which like reference numerals refer to like parts throughout.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0013] Fig. 1 illustrates a block diagram of an exemplary mobile services network, in accordance with an embodiment of the present invention.

[0014] Fig. 2a illustrates a flow diagram of an exemplary method of operating a mobile handset when it powers up or reboots to start a firmware/software update using an update agent, in accordance with an embodiment of the present invention.

[0015] **Fig. 2b** illustrates a flow diagram of another exemplary method of operating a mobile handset when it powers up or reboots to start a firmware/software update using an update agent, in accordance with an embodiment of the present invention.

[0016] Fig. 3a illustrates an exemplary memory map of a mobile handset, in accordance with an embodiment of the present invention.

[0017] Fig. 3b illustrates an exemplary memory map divided into 8 banks, in accordance with an embodiment of the present invention.

[0018] Fig. 3c illustrates an exemplary memory map using a tri-phase boot, in accordance with an embodiment of the present invention.

[0019] Fig. 3d illustrates another exemplary memory map using tri-phase boot, in accordance with an embodiment of the present invention.

[0020] Fig. 3e illustrates an exemplary memory map using tri-phase boot utilizing "Boot Block" memory, in accordance with an embodiment of the present invention.

[0021] Fig. 4 illustrates a flow diagram of an exemplary tri-phase boot process, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0022] The present invention relates generally to updates of firmware/software components in electronic devices such as, for example, mobile handsets, and specifically to the update agent in electronic devices being capable of updating itself. Although the following discusses aspects of the invention in terms of a mobile handset, it should be clear that the following discussion also applies to other mobile electronic devices such as, for example, personal digital assistants (PDAs), pagers, personal computers (PCs), and similar handheld electronic devices.

[0023] **Fig. 1** illustrates a block diagram of an exemplary mobile services network 105, in accordance with an embodiment of the present invention. The mobile services network 105 may comprise a mobile handset 107, a management server 109, and an update package repository 133. In an embodiment of the present invention, an update package may contain information needed to upgrade software/ firmware in the mobile handset 107 from one version to another. In an embodiment of the present invention, the mobile handset 107 may have access to services such as, for example, firmware/software update services. The mobile handset 107 may retrieve an update package from the management server 109 and confirm the authenticity of an update package before initiating the update process. In an embodiment of the present invention, the mobile handset 107 may utilize an update agent 113 in the update process. The mobile handset 107 may be linked to the management server 109 via a communication network 143. The communication network 143 may be a wireless or a wired network. In an embodiment of the present invention, the communication network 143 may be an existing network such as, for example, the Internet or a service (public) network such as, for example, a cellular wireless network, or a private network specifically designed for connecting a plurality of mobile handsets 107 and management servers 109.

[0024] In an embodiment of the present invention, the mobile handset 107 may comprise a non-volatile memory 111, a random access memory (RAM) 125, and security services 123. The non-volatile memory 111 of the mobile handset 107 may comprise an update agent 113, firmware and real-time operating system (RTOS) 115, an operating system (OS) layer 117, a download agent or browser 119, end-user-related data and content 121, and boot initialization 127.

[0025] In an embodiment of the present invention, the mobile handset 107 may download an update package from the update package repository 133 to update the update agent 113, and the mobile handset 107 may then reboot. The availability of update packages may be recorded in status information that may be stored in non-volatile memory 111 in the mobile handset 107. Upon a subsequent startup, the mobile handset 107 may execute the boot initialization 127, and then determine whether there is a need to execute the update agent 113 based on status information that may be available in the non-volatile memory 111. If the mobile handset 107 determines that the update agent 113 needs to execute the update process, the mobile handset 107 may invoke the update agent 113. In an embodiment of the present invention, the update agent 113 may determine that the update agent 113 itself needs to be updated. The update agent 113 may then create a backup of itself in a backup section such as, for example, a working flash bank (WFB) of the non-volatile memory 111 before updating the update agent 113, to provide for the recovery of the copy of the existing update agent 113 if the update process fails for any reason. Thus, the update agent 113 may provide updating in the mobile handset 107 in a fault tolerant manner utilizing a fault tolerant technique, an example of which may be found in International Patent Application Publication No. WO 02/41147 A1, entitled "Systems And Methods For Updating And Distributing Information," publication date March 23, 2002, the complete subject matter of which is hereby incorporated herein by reference, in its entirety.

[0026] **Fig. 2a** illustrates a flow diagram of an exemplary method of operating a mobile handset when it powers up or reboots to start a firmware/software update using an update agent, in accordance with an embodiment of the present invention. The following discussion of **Fig. 2a** makes reference to items shown in **Fig. 1**. At a start block 207, the processing may start when the mobile handset such as, for example, the mobile handset 107 of **Fig. 1** is powered up or rebooted. Then, at a next block 209, the mobile handset may execute the boot initialization code. At a next decision block 211, the mobile handset 107 may determine whether an update of firmware/update agent needs to be executed using the update agent such as, for example, the update agent 113 of **Fig. 1**. An appropriate update package may be retrieved via a management server such as, for example, the management server 109 of **Fig. 1**. If, at the decision block 211, the mobile handset 107 determines that an update is not needed, a regular startup of the mobile

handset 107 may be initiated at a next block 223, and the process may terminate at the end block 231.

[0027] If, however, the mobile handset 107 determines that an update is to be executed, then at a next decision block 213, the mobile handset may determine whether the existing update agent 113 is to be updated. If the mobile handset 107 determines that the update agent 113 is not to be updated, then, at a next block 227, a standard update process may be invoked. A reboot of the mobile handset 107 may then be initiated at the next block 229 before processing returns to the start block 207.

[0028] If, at the decision block 213, the mobile handset 107 determines that the update agent 113 needs to be updated, then, at a next block 215, the existing (old) update agent 107 may be backed up in a backup section of non-volatile memory 111 in the mobile handset 107. The update agent 107 may then be updated employing the contents of the update package that may have been retrieved earlier and stored in the non-volatile memory 111 or in the RAM 125. Then, at a next block 217, the mobile handset 107 may be rebooted. At a next decision block 219, the mobile handset 107 may determine which update agent should be used: the updated update agent (new) or the old update agent that may be available in the backup section of non-volatile memory 111. It may be necessary to use the old update agent in situation such as, for example, when the new update agent may be corrupted or ineffective. If the mobile handset 107 determines that the new update agent may be used, then a regular startup of the mobile handset 107 may be initiated at a next block 223, and the process may terminate at the end block 231.

[0029] If, at the decision block 219, the mobile handset 107 determines that the new update agent may be corrupted or unusable, then, at a next block 225, the old update agent from the backup section of the non-volatile memory 111 may be re-activated. In an embodiment of the present invention, the old update agent may be copied back to the default location for an update agent 107 within the non-volatile memory 111. A regular startup of the mobile handset 107 may then be initiated at a next block 223, and the process may terminate at the end block 231.

[0030] **Fig. 2b** illustrates a flow diagram of another exemplary method of operating a mobile handset when it powers up or reboots to start a firmware/software update using an update agent, in accordance with an embodiment of the present invention. The

description of the exemplary embodiment of **Fig. 2b** is similar to that of **Fig. 2a** with the exception that following block 215, the mobile handset 107 may be rebooted at block 229. Processing may then restart at a start block 207, and boot initialization code may be executed at a next block 209. Then, at a next decision block 233, the mobile handset 107 may determine whether an update was performed on the update agent 113. If the mobile handset 107 determines that the update agent 113 was not updated, the process goes back to a next block 211. If the mobile handset 107 determines that the update agent 113 was updated, the process proceeds to block 219, and continues as described hereinabove.

[0031] In an embodiment of the present invention, updates to the update agent 107 may be used to fix bugs in an earlier version of the update agent, keep the update agent 107 current to accommodate the latest technologies and improvements, keep up to date with changes in the updating process, accommodate changes in the interface to the download agent that may be used by the mobile handset to download the update packages, etc.

[0032] **Fig. 3a** illustrates an exemplary memory map 301 of a mobile handset such as, for example, the mobile handset 107 of **Fig. 1**, in accordance with an embodiment of the present invention. The memory map 301 represents a 512kB flash memory section that may use uniform physical blocks of 64kB each. The flash memory may be split into banks, each bank 64kB in size. Thus 8 64kB banks would result, shown in **Fig. 3a** as bank 0 to bank 7.

[0033] **Fig. 3b** illustrates an exemplary memory map 303 divided into 8 banks, in accordance with an embodiment of the present invention. In an embodiment of the present invention, an update agent such as, for example, the update agent 113 of **Fig. 1**, of a mobile handset such as, for example, the mobile handset 107, may be positioned within the first bank, bank 0. A WFB may be placed in a free area of flash memory such as, for example, bank 7, and an image of the mobile handset firmware, which normally starts at address location 0x0 in bank 0, may be displaced to the second bank, bank 1. In an embodiment of the present invention, the positioning of the update agent may be done at a binary level, such that, the updated agent may not be compiled with the image of the mobile handset firmware. Instead, the update agent may be compiled independently and may not reference anything in the image of the mobile handset firmware directly. In an embodiment of the present invention, the image of the mobile handset firmware may be

of any size, so long as the image of the mobile handset firmware does not grow into the space occupied by the WFB.

In an embodiment of the present invention, a tri-phase boot may be utilized in updating the update agent. **Fig. 3c** illustrates an exemplary memory map 305 using a tri-phase boot, in accordance with an embodiment of the present invention. In such an embodiment of the present invention, checksums may be associated with the first bank, bank 0, containing the update agent, and with the WFB bank, bank 7. A "Boot Checker" bank may also be added. In an embodiment of the present invention, each checksum may be used to validate the update agent that is associated with that bank. For example, the CS in the first bank, bank 0, is the checksum for the update agent in that bank. The checksum may be defined by the system designer according to the specific requirements or needs of the system and it may be a simple sum or a more complicated value such as, for example, cyclic redundancy code (CRC), MD5 checksum, hash code, etc. In an embodiment of the present invention, an update may have been already attempted, and the WFB may hold a copy of the original update agent along with its associated CS. The checksum may be computed for the original update agent firmware image soon after the compilation of the original update agent firmware image.

[0034] The "Boot Checker" may be a small piece of code that is the first code run upon boot-up. In an embodiment of the present invention, this code may include device specific functionality with the purpose of checking for a bank containing a valid update agent before branching to the update agent, because the updated original update agent may have been corrupted in a prior updating process.

[0035] Fig. 4 illustrates a flow diagram of an exemplary tri-phase boot process, in accordance with an embodiment of the present invention. After a minimal initialization in a mobile handset at block 407, the first bank, bank 0, may be checked for validity at block 409. The method chosen to calculate the checksum for this validity check may reside within the "Boot Checker" bank. If bank 0 were valid, it may be assumed that bank 0 contains the correct update agent, at block 411. The bank may include, in addition to the update agent, an initialization code that may have been used earlier.

[0036] In an embodiment of the present invention, if bank 0 were not valid, it may indicate that an update of the update agent was attempted but was interrupted. The WFB

may then be checked for validity, at block 413, since the original update agent may be located in the WFB. The "Boot Checker" bank may then branch to the WFB at block 415.

[0037] In an embodiment of the present invention, the Tri-Phase Boot method may involve a process similar to that of updating the image of the mobile handset firmware. This approach may involve the update agent itself applying the update in the device. In an embodiment of the present invention, the checksum of the update agent may be applied to the update agent firmware image before generating the update package. In an embodiment of the present invention, an external tool such as, for example, an update package generator with a generator user interface may be used to calculate the checksum of the update agent firmware image. Using a generator and an update agent for updating the update agent, proper provisioning, security, and fault-tolerance may be maintained.

[0038] **Fig. 3d** illustrates another exemplary memory map 307 using tri-phase boot, in accordance with an embodiment of the present invention. In an embodiment of the present invention, the first bank, bank 0, may be used as the "Boot Checker" and the update agent and image of the mobile handset firmware may be shifted forward by one bank, to banks 1 and 2, respectively.

[0039] **Fig. 3e** illustrates an exemplary memory map using tri-phase boot utilizing "Boot Block" memory, in accordance with an embodiment of the present invention. In an embodiment of the present invention, the "Boot Block" may be comprised of 8kB blocks. A small boot checker may be placed in block 0. The update agent may be positioned starting at bank 1 and may, as a result, occupy up to about 56kB of space. In an embodiment of the present invention, the need for extra flash memory blocks is eliminated.

[0040] In an embodiment of the present invention, implementation of the boot checker may be done in Read Only Memory (ROM), mask ROM, or another flash device including internal flash.

[0041] Although the discussion hereinabove provided exemplary illustrations of memory maps with variables and components in specific locations, it should be clear that locations and sizes of memory blocks may be altered based on the requirements and the design of the specific systems.

[0042] While the present invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present invention without departing from its scope. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed, but that the present invention will include all embodiments falling within the scope of the appended claims.

CLAIMS

What is claimed is:

1. A system that facilitates the updating of at least one of firmware and updating software in an electronic device, using updating information received via a communication network, the system comprising:

a non-volatile memory comprising:

firmware;

loading software for loading the updating information via the communication network;

updating software for applying the loaded updating information to the at least one of firmware and updating software; and

initializing software for initializing the electronic device;

a RAM; and

security services for verifying the loaded updating information.

2. The system according to claim 1 wherein the loading software receives the updating information via the communication network.

3. The system according to claim 1 wherein the updating software is capable of updating itself.

4. The system according to claim 3 wherein the updating software saves a back up copy of the updating software in the non-volatile memory before updating itself.

5. The system according to claim 3 wherein the updating software saves a back up copy the updating software in the RAM before updating itself.

6. The system according to claim 1 wherein the updating software uses a fault tolerant technique in the updating process.

7. The system according to claim 1 wherein the verifying utilizes cyclic redundancy code.

8. The system according to claim 1 wherein the communication network comprises a wireless network.

9. The system according to claim 1 wherein the initialization software determines if the applying of the loaded updating information is successful.

10. The system according to claim 9 wherein the initialization software uses a back up copy of the at least one of firmware and updating software if the applying of the loaded updating information is not successful.

11. The system according to claim 1 wherein the initialization software continues an interrupted applying of the loaded updating information.

12. A method for updating at least one of firmware and updating software in an electronic device, using updating information received via a communication network, the method comprising:

initializing the electronic device; and

determining whether at least one of the firmware and the updating software is to be updated.

13. The method according to claim 12 wherein, if it is determined that no updating needs to be done, the method further comprises performing a normal start up of the electronic device.

14. The method according to claim 12 wherein, if it is determined at least one of the firmware and the updating software needs to be updated, the method further comprises determining whether the updating software needs to be updated.

15. The method according to claim 14 wherein, if it is determined that the updating software does not need updating and the firmware needs updating, the method further comprises:

updating the firmware using the updating information; and

initializing the electronic device.

16. The method according to claim 14 wherein, if it is determined that the updating software needs to be updated, the method further comprises:

backing up the updating software;

updating the updating software using the updating information to produce a new updating software;

initializing the electronic device; and

determining whether the updating of the updating software was successful.

17. The method according to claim 16 wherein the backing up of the updating software is done by storing the updating software in a non-volatile memory in the electronic device.

18. The method according to claim 16 wherein the backing up of the updating software is done by storing the updating software in a RAM in the electronic device.

19. The method according to claim 16 wherein, if it is determined that updating the updating software was successful, the method further comprises:

enabling use of the new updating software; and

initializing the electronic device.

20. The method according to claim 16 wherein, if it is determined that updating the updating software was not successful, the method further comprises:

enabling use of the backed up updating software; and

initializing the electronic device.

21. The method according to claim 12 wherein the communication network is wireless.

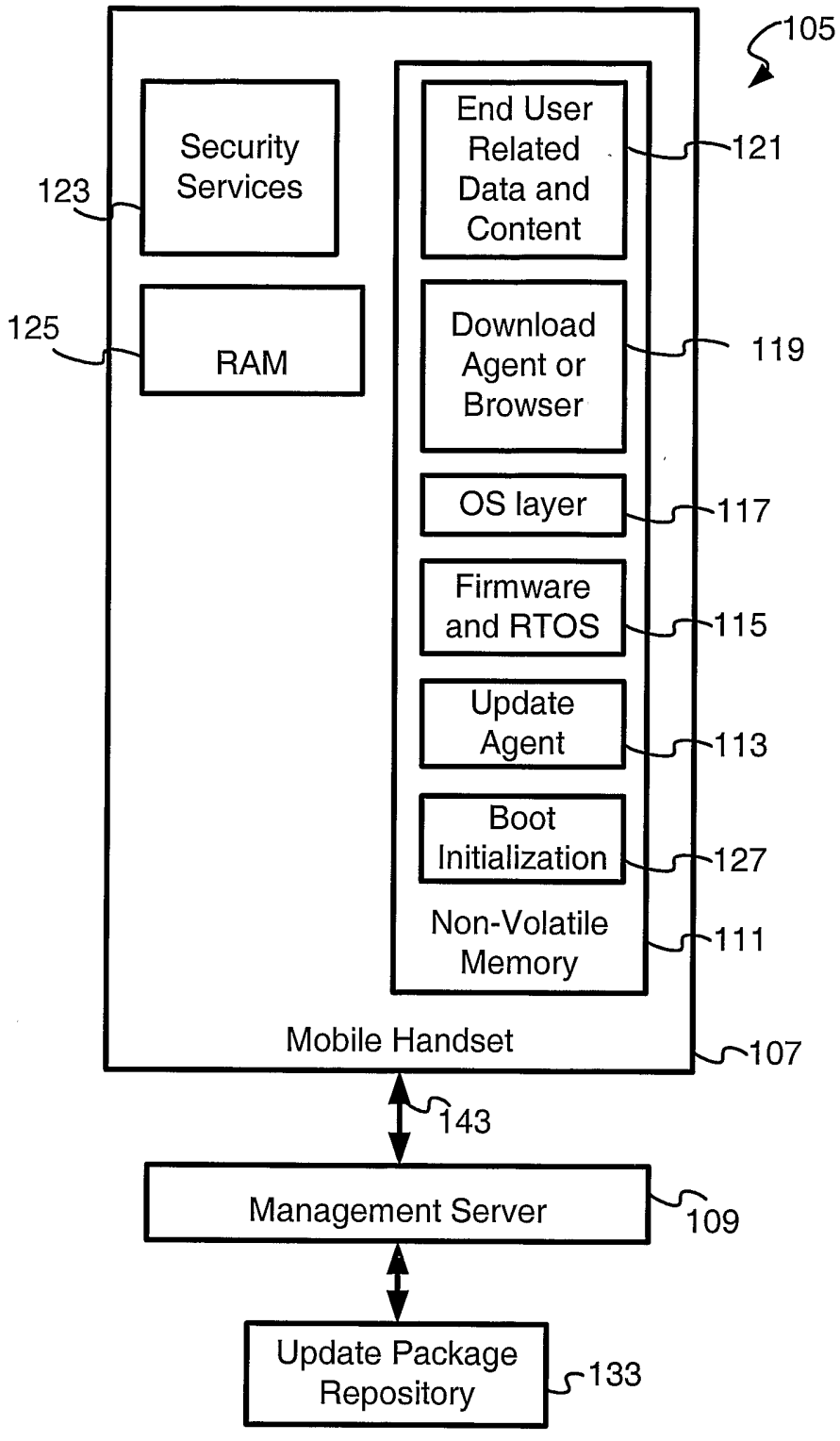

105
Security Services
123
RAM
125
End User Related Data and Content
121
Download Agent or Browser
119
OS layer
117
Firmware and RTOS
115
Update Agent
113
Boot Initialization
127
Non-Volatile Memory
111
Mobile Handset
107
143
Management Server
109
Update Package Repository
133

Fig. 1

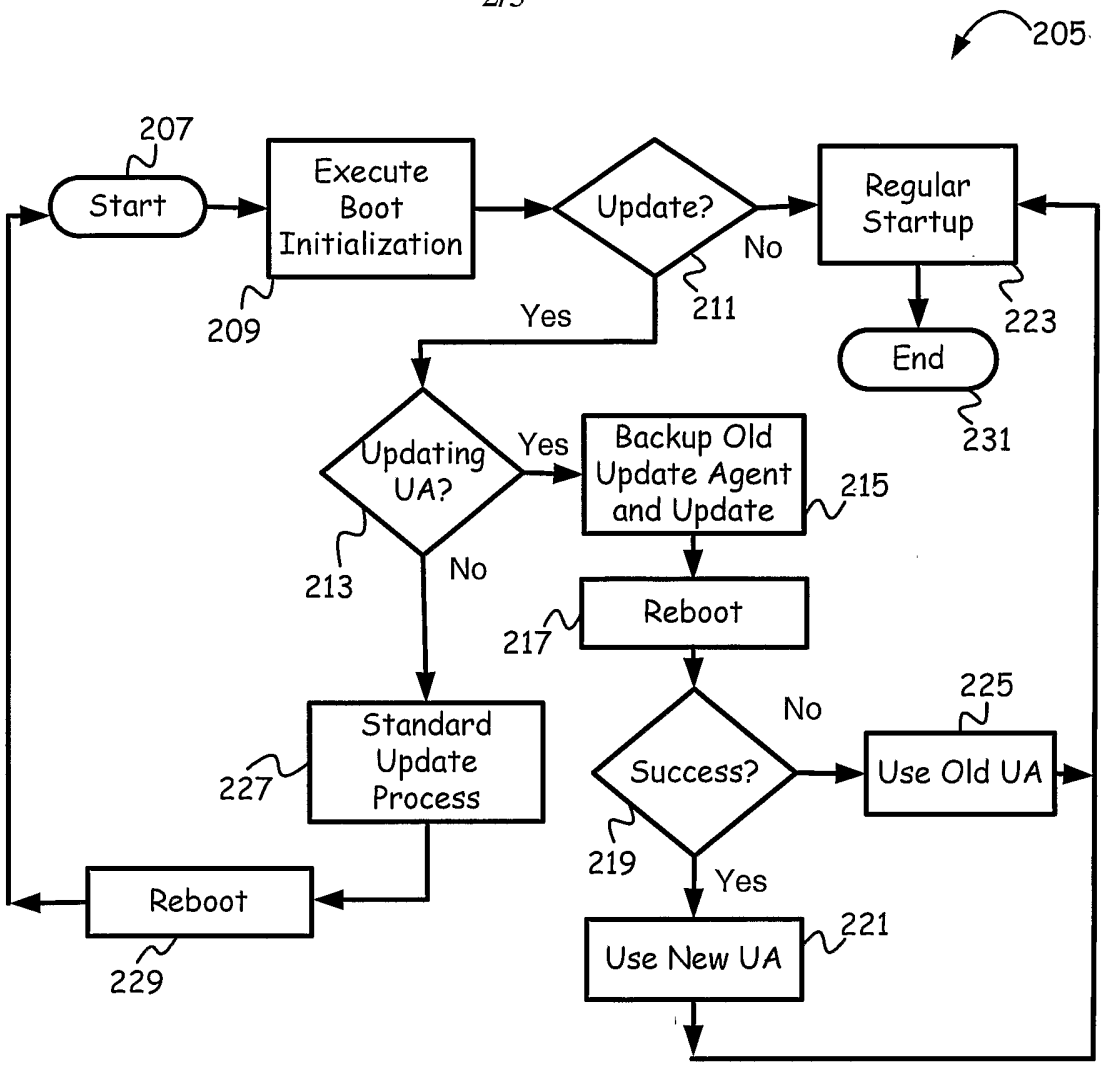

205
207
Start
Execute Boot Initialization
209
Update?
No
211
Regular Startup
223
End
231
Yes
Updating UA?
213
Yes
Backup Old Update Agent and Update
215
No
Reboot
217
Standard Update Process
227
No
225
Success?
Use Old UA
219
Yes
Use New UA
221
Reboot
229

Fig. 2A

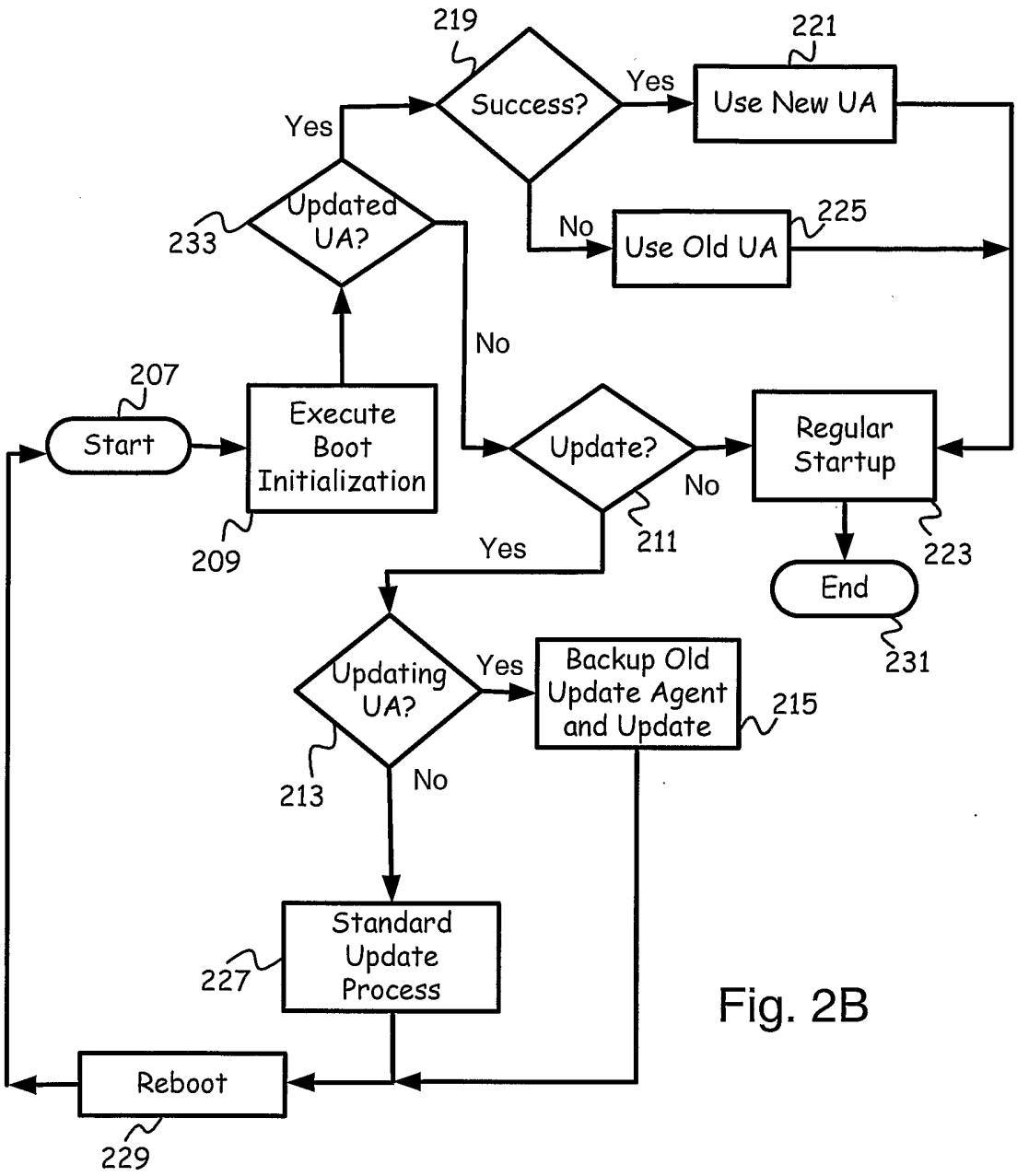

219
Success?
Yes
221
Use New UA
Yes
Updated UA?
233
No
Use Old UA
225
No
207
Start
Execute Boot Initialization
209
Update?
No
211
Regular Startup
223
Yes
End
231
Updating UA?
Yes
Backup Old Update Agent and Update
215
213
No
Standard Update Process
227
Reboot
229

Fig. 2B

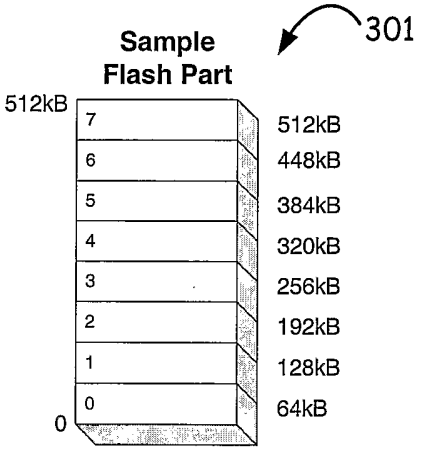


Fig. 3A

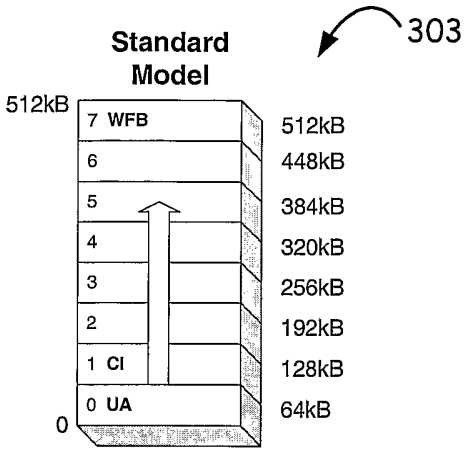


Fig. 3B

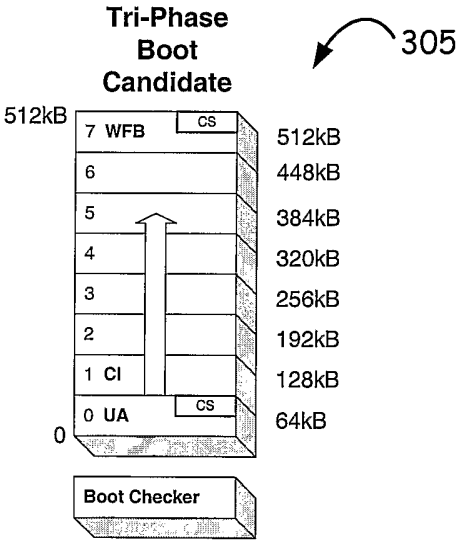


Fig. 3C

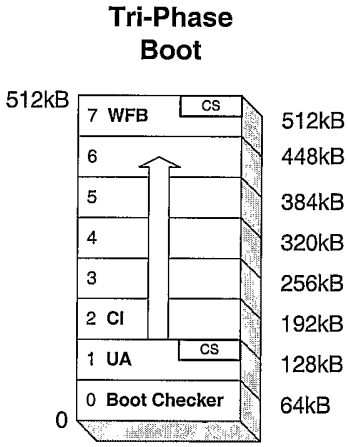


Fig. 3D

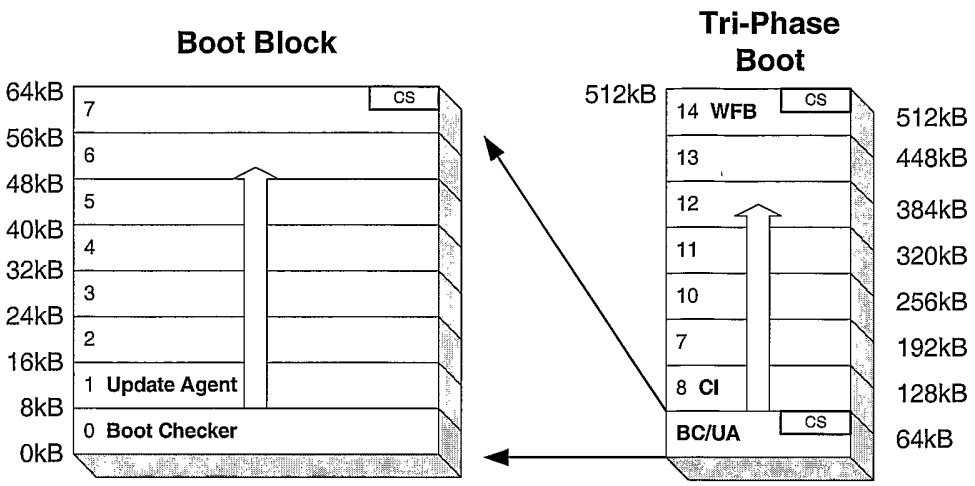

Boot Block
Tri-Phase Boot
64kB
56kB
48kB
40kB
32kB
24kB
16kB
8kB
0kB
7
6
5
4
3
2
1 Update Agent
0 Boot Checker
CS
512kB
14 WFB
13
12
11
10
7
8 CI
BC/UA
512kB
448kB
384kB
320kB
256kB
192kB
128kB
64kB

Fig. 3E

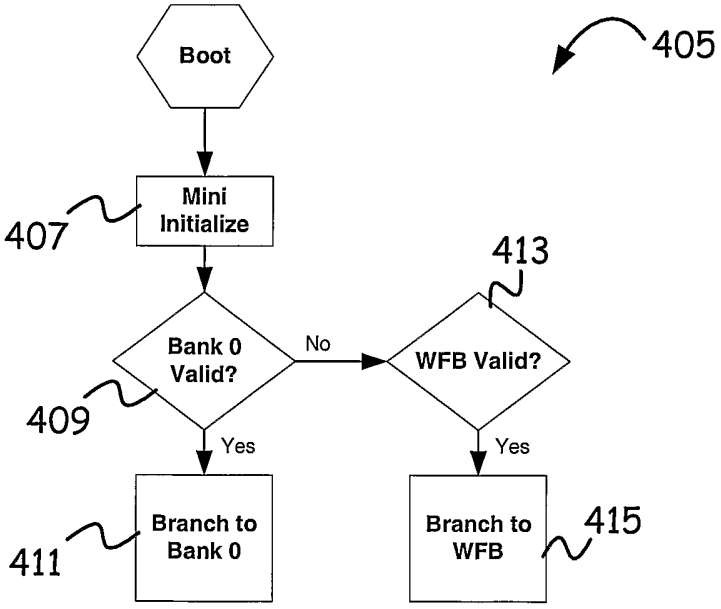

Boot
405
Mini Initialize
407
Bank 0 Valid?
409
No
WFB Valid?
413
Yes
Yes
Branch to Bank 0
411
Branch to WFB
415

Fig. 4