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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 December 2022 (22.12.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/265652 A1

(51) International Patent Classification:
G06F 11/14 (2006.01) *G06F 11/36* (2006.01)

(21) International Application Number:
PCT/US2021/038011

(22) International Filing Date:
18 June 2021 (18.06.2021)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 10300 Energy Drive, Spring, Texas 77389 (US).

(72) Inventors: **FENG, Kang-Ning**; 10F, No.66, Jing Mao 2 Rd., NanGang District, Taipei City, 11568 (TW). **HUNG, Ming-Chang**; 10F, No. 66 Jing Mao 2 Rd, NanGang District, Taipei City, 11568 (TW). **HUANG, Wei-Chih**; 10F, No. 66 Jing Mao 2 Rd, NanGang District, Taipei City, 11568 (TW).

(74) Agent: **CARTER, Daniel J.** et al.; HP Inc., 3390 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528-9544 (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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW,

(54) Title: COMPONENT FIRMWARE REPLACEMENTS VIA NETWORKS

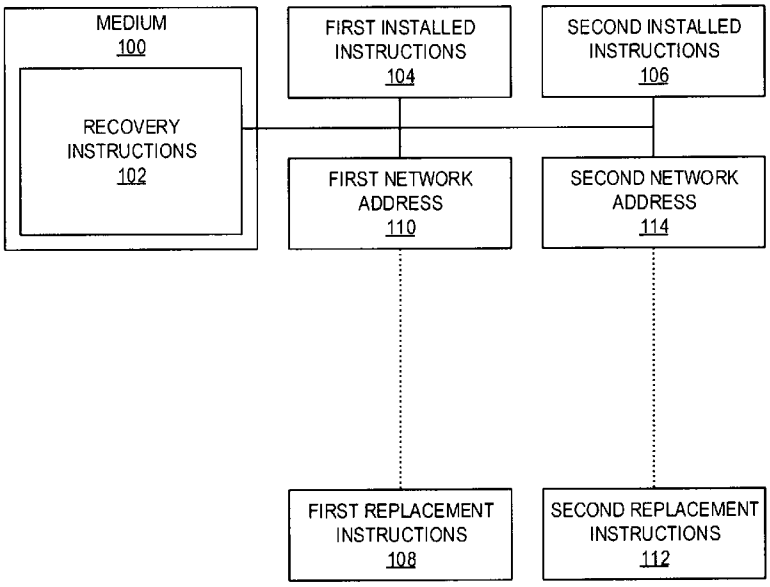

FIG. 1

(57) Abstract: An example non-transitory machine-readable medium includes recovery instructions that, when executed by a processor, cause the processor to detect a corruption of first installed instructions of first component firmware at a computing device and second installed instructions of second component firmware at the computing device. In response to detection of the corruption during pre-boot, if the first installed instructions are detected as corrupt, the recovery instructions request and receive first replacement instructions from a first network address, and replace in the computing device the first installed instructions with the first replacement instructions. If the second installed instructions are detected as corrupt, the recovery instructions request and receive second replacement instructions from a second network address different from the first network address, and replace in the computing device the second installed instructions with the second replacement instructions.

WO 2022/265652 A1

[Continued on next page]

SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

COMPONENT FIRMWARE REPLACEMENTS VIA NETWORKS

BACKGROUND

[0001] Computing devices include firmware to implement functions related to hardware components. It is often the case that a computing device is constructed from hardware components provided by a number of different manufacturers or vendors. Such parties may also provide component firmware to accompany the provided hardware.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] FIG. 1 is a block diagram of an example non-transitory machine-readable medium with component-firmware recovery instructions that detect a corrupted component firmware and obtain and install a replacement from a network address.

[0003] FIG. 2 is a block diagram of an example computing device that references a mapping to obtain and install a replacement for a corrupted component firmware.

[0004] FIG. 3 is a table of an example mapping of component firmware identifiers to network addresses of replacement component firmware.

[0005] FIG. 4 is a block diagram of an example recovery server that provides network addresses of replacement component firmware in response to requests.

[0006] FIG. 5 is a flowchart of an example method of obtaining replacement component firmware for component firmware detected to be corrupted.

[0007] FIG. 6 is a flowchart of an example method of installing replacement component firmware to replace component firmware detected to be corrupted.

[0008] FIG. 7 is a flowchart of an example method of obtaining replacement component firmware for component firmware detected to be corrupted and installing the replacement component firmware.

DETAILED DESCRIPTION

[0009] Component firmware may become corrupted, which may lead to decreased performance of a computing device, data loss, crash, or other significant malfunction.

[0010] Replacing component firmware is often a task for the end user or a service technician. It is often the case that a person manually downloads and installs new component firmware to replace a corrupted version.

[0011] Backup copies of component firmware may be stored on a computing device to act as replacement if corruption occurs. However, this takes storage space and such backup copies may become stale if not updated frequently.

[0012] Other approaches rely on an operating system (OS) of the computing device to detect and replace corrupted component firmware. However, this requires the OS to launch and severe component firmware corruption may prevent that. In addition, OS-based firmware recovery is OS-specific and may need to be implemented for different operating systems and versions thereof.

[0013] As such, as discussed herein, if component firmware becomes corrupted, then a mapping of component firmware may be referenced and a replacement for corrupted component firmware may be downloaded from a remote source, such as a hardware component manufacturer's server, a centrally managed server, or similar source. The mapping may be maintained remote from the computer by the manufacturer or provider of the computer. The mapping may represent a best known configuration (BKC) for elements of component firmware of the computer and may differ for different models of computer, versions of component firmware, version of Basic Input/Output

System (BIOS), *etc*. The mapping may be maintained at a recovery server, so that the BKC for various computing devices may be kept current and consolidated at one location.

[0014] A BIOS may carry out the detection and replacement of a corrupted component firmware. Detection and replacement may be performed during pre-boot of the computing device. As such, the OS need not participate in component firmware recovery. This may increase efficiency and speed recovery by avoiding launching the OS until recovery has been completed.

[0015] FIG. 1 shows an example non-transitory machine-readable medium 100 with component-firmware recovery instructions 102 that detect a corrupted component firmware and obtain and install a replacement from a network address.

[0016] The non-transitory machine-readable medium 100 may be part of a computing device, such as a desktop computer, notebook computer, all-in-one (AIO) computer, server, or similar device.

[0017] The non-transitory machine-readable medium 100 may include a non-volatile memory, such as a read-only memory (ROM), electrically-erasable programmable read-only memory (EEPROM), or flash memory that cooperates with a processor, such as a central processing unit (CPU) of a computing device, to execute the instructions. The non-transitory machine-readable medium 100 may include an electronic, magnetic, optical, or other physical storage device that encodes the instructions 102 that implement the functionality discussed herein.

[0018] The recovery instructions 102 may be directly executed, such as binary or machine code, and/or may include interpretable code, bytecode, source code, or similar instructions that may undergo additional processing to be executed. All of such examples may be considered executable instructions.

[0019] The recovery instructions 102 detect the corruption of instructions of installed component firmware. Multiple sets of instructions may be installed at a

computing device in non-volatile memory to implement multiple different component firmware that support different hardware components of the computing device. In various examples, each element of component firmware supports a hardware component, and each component firmware may be supplied by the manufacturer or supplier of the hardware component. Examples of hardware components include a serial bus controller, such as a Universal Serial Bus (USB™) controller or Thunderbolt™ controller, USB Power Delivery (PD) controller, camera, memory card reader, fingerprint reader, audio card, graphics card, storage device, and similar hardware devices. Component firmware may be provided on a non-transitory machine-readable medium, such as a non-volatile memory chip, carried by the hardware component. In some examples, component firmware may include initialization code and/or a driver for the hardware component.

[0020] Detection of corruption may be performed by the recovery instructions 102 receiving a signal, such as a hardware interrupt, from a hardware device. A hardware component may issue a signal to the processor to indicate an error, which may be the result of component firmware corruption. An interrupt, hardware vendor-provided protocol, or other signal may be available to the processor and thus to the instructions 102 by way of a status register, for example. The instructions 102 being executed by the processor may identify the signal as indicating firmware corruption. In other examples, the instructions 102 may compute a hash, cyclic redundancy check (CRC), or other value of the firmware, and then check such value against an accepted value to determine whether or not the component firmware has been corrupted.

[0021] In this example, first installed instructions 104 are provided for a first component firmware and second installed instructions 106 are provided for a second component firmware. The instructions 104, 106 are designated as installed in that they are installed at the computing device to support the associated hardware components. The first installed instructions 104 provide a first functionality to the computing device, such as initializing and/or providing a driver for a hardware component. The second installed instructions 106 provide

a different, second functionality, such as initializing and/or providing a driver for a different hardware component. Any number of sets of instructions 104, 106 may be provided to support any suitable number of hardware components, with two being discussed here merely for sake of brevity.

[0022] The recovery instructions 102 detect for corruption of the first and second installed instructions 104, 106. Detection of the corruption may inherently identify the cause of the corruption. For example, a hardware interrupt may identify the hardware component and thus the firmware.

[0023] If the first installed instructions 104 are detected as corrupt, the recovery instructions 102 request first replacement instructions 108 from a first network address 110. A computing device at the first network address 110 responds with the first replacement instructions 108, which are received by the computing device. The recovery instructions 102 then replace the first installed instructions 104 with the received first replacement instructions 108. After installation, the first replacement instructions 108 become the new first installed instructions to carry out the functionality of the first component firmware.

[0024] Likewise, if the second installed instructions 106 are detected as corrupt, the recovery instructions 102 request second replacement instructions 112 from a second network address 114. A computing device at the second network address 114 responds with the second replacement instructions 112, which are received by the computing device. The recovery instructions 102 then replace the second installed instructions 106 with the received second replacement instructions 112. After installation, the second replacement instructions 112 become the new second installed instructions to carry out the functionality of the second component firmware.

[0025] The recovery instructions 102 may include instructions to initiate and operate a network interface of the computing device, so as to allow access to the network addresses 110, 114 to obtain respective replacement instructions 108, 112 for any corrupt component firmware.

[0026] The network addresses 110, 114 may be stored in a mapping that is accessible by the recovery instructions 102 and that associates component firmware with network addresses. Component firmware may be uniquely identified by hardware device type, hardware device type provider/manufacturer, component firmware version, model of computing device, BIOS version, and similar information. The mapping may define best known configurations of component firmware for various combinations of such information. The mapping may be stored at the computing device, such as in the medium 100, or may be stored remotely.

[0027] The instructions 102 may form part of a BIOS that may initialize, control, or operate a computing device that contains the medium 100 prior to execution of an OS of the computing device. Instructions included within the BIOS may include software, firmware, microcode, or other programming that defines or controls functionality or operation of the BIOS. In some examples, the BIOS may be implemented using instructions, such as platform firmware of a computing device, executable by a processor, such as a central processing unit (CPU) of the computing device. The BIOS may operate or execute prior to the execution of the OS of the computing device. The BIOS may initialize, control, or operate components such as hardware components of a computing device and may load or boot the OS of computing device.

[0028] In some examples, a BIOS may provide or establish an interface between hardware devices or platform firmware of the computing device and an OS of the computing device, via which the OS of the computing device may control or operate hardware devices or platform firmware of the computing device. In some examples, a BIOS may implement the Unified Extensible Firmware Interface (UEFI) specification or another specification or standard for initializing, controlling, or operating a computing device.

[0029] The instructions 102 may be executed during pre-boot of the computing device, such as during a driver execution environment (DXE) phase of a UEFI boot sequence. Hence, corruption of component firmware may be

detected before OS launch. In various examples, after the replacement component firmware, the instructions 102 may reset the computing device and replace the component firmware on the subsequent pre-boot. The OS may then be subsequently launched without the corrupted component firmware. As such, the OS may be excluded from recovery of corrupted component firmware, which may reduce the time required to recover in that the time required to launch the OS is not expended. Resetting the computing device also allows the instructions 102 to verify that the replacement component firmware is not corrupt by predicting an additional cycle of corruption detection.

[0030] FIG. 2 is a block diagram of an example computing device 200 that references a mapping to obtain and install a replacement for a corrupted component firmware.

[0031] The computing device 200 includes a processor 202, memory 204, a non-transitory machine-readable medium 100, a chipset 206, a network interface 208, and a plurality of hardware devices 210, 212, 214.

[0032] The processor 202 may include a central processing unit (CPU), a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), or a similar device capable of executing instructions.

[0033] The memory 204 may include random-access memory (RAM) or similar working memory for use by the processor 202.

[0034] The chipset 206 may include an integrated circuit or circuits that connect and manage data flow among the processor 202, memory 204, medium 100, chipset 206, network interface 208, and hardware devices 210, 212, 214.

[0035] The network interface 208 may include hardware, such as a network adaptor card, network interface controller, or network-capable chipset, and may further include instructions, such firmware. The network interface 208 allows data to be communicated between the computing device 200 and a computer network 216, such as a local-area network (LAN), wide-area network (WAN),

virtual private network (VPN), the internet, or similar networks that may include wired and/or wireless pathways.

[0036] The hardware devices 210, 212, 214 include respective component firmware 220, 222, 224. Examples of hardware devices 210, 212, 214 include USB controllers, cameras, memory card readers, and so on, as discussed elsewhere herein. Each component firmware 220, 222, 224 includes instructions to initialize and/or provide a driver for the respective hardware devices 210, 212, 214.

[0037] The medium 100 may store a BIOS 226 for execution by the processor 202. The BIOS 226 may include component-firmware recovery instructions 228, which may include functionality discussed above with respect to the instructions 102. The BIOS 226 may further include a mapping network address 230 that identifies a recovery server 232 that hosts a mapping 234 that associates replacement component firmware with network addresses. The mapping 234 may be maintained remotely from the computing device 200 and the recovery instructions 228 may query the mapping 234 via a network 216.

[0038] The recovery server 232 may be a computing device that hosts the mapping 234 at a Uniform Resource Locator (URL) that identifies a protocol or scheme, domain, and path of the mapping 234. The mapping 234 may be a data file, a directory of files, a database responsive to queries, or similar data structure that can store and provide information.

[0039] The mapping network address 230 is the address of the mapping 234 as available on the network 216 and, as such, may be a URL expression. The mapping network address 230 may identify a file structure or a database that provides network addresses of replacement component firmware 240, 242, 244 made available via the network 216 on component-firmware servers 236, 238, 240. The mapping network address 230 may be configurable by the manufacturer or provider of the computing device 200 or by the user of the computing device 200.

[0040] In some examples, information that uniquely identifies each component firmware is arranged in a directory structure, such that a mapping network address 230 conforming to the following structure uniquely identifies a file:

[0041] https://www.example.com/model/BIOSversio/hw/FWversion/

[0042] where “model” is replaced with a model identifier of the computing device (*e.g.*, Notebook_Model_1),

[0043] “BIOSversion” is replaced with a version identifier of the BIOS (*e.g.*, 00.01.00),

[0044] “hw” is replaced with a hardware identifier (*e.g.*, Camera_A), and

[0045] “FWversion” is replaced with a version identifier of the component firmware of the hardware (*e.g.*, 1.02).

[0046] As such, the following specific mapping network address 230 may be a directory that contains a file that contains a network address of the respective replacement component firmware:

[0047] https://www.example.com/ Notebook_Mode_1/00.01.00/Camera_A/1.02/

[0048] The file at this location may contain the network address of the requested replacement component firmware 240, 242, 244, for example:

[0049] https://www.example1.com/Camera_A_1-02.dat

[0050] This file at the above network address may be the replacement component firmware 240, 242, 244 itself.

[0051] In other examples, the mapping network address 230 may identify a database that responds to a parameterized query, such as:

[0052] https://www.example.com/ ?model=Notebook_Model_1&BIOSversion=00.01.00&hw=Camera_A&FWversion=1.02

[0053] Such a query may be parsed and executed by the recovery server 224 against a database that contains a table of network addresses of replacement component firmware 240, 242, 244.

[0054] Hence, a structured request made with the mapping network address 230 returns a network address of the replacement component firmware 240, 242, 244. The recovery server 232 thus acts as a central directory of replacement component firmware 240, 242, 244 that may be stored on component-firmware servers 246, 248, 250.

[0055] The recovery server 232 may be managed by an organization that makes, sells, or services the computing device 200. The component-firmware servers 246, 248, 250 may be managed by organizations that make, sell, or service the hardware devices 210, 212, 214. As such, the servers 232, 246, 248, 250 may be in separate domains and may be operated independently of each other. Any suitable number of recovers servers 232 and component-firmware servers 246, 248, 250 may be used.

[0056] The recovery instructions 228 may monitor and detect corruption of component firmware 220, 222, 224 installed at the computing device 200. In response to detection of corruption, the instructions 228 identify the component firmware that is corrupt for example, by hardware name/type and component firmware version, and then construct a query with the mapping network address 230 using information such as the model of the computing device 200 and version of the BIOS 226. The instructions may then address such query to the recovery server 232 via the network interface 208 and network 216.

[0057] The recovery server 232, in response to receiving from the computing device 200 the query indicating the corrupted component firmware, then determines a network address for the corresponding replacement component

firmware 240, 242, 244 by referencing the mapping 234 of identifiers of component firmware 240, 242, 244 to network addresses. The recovery server 232 then transmits, via the network 216, an indication of the network address (*e.g.*, a URL) to the computing device 200.

[0058] The recovery instructions 228 receive the indication of the network address and request the corresponding replacement component firmware 240, 242, 244 from the respective server 246, 248, 250. The recovery instructions 228 then download and install the replacement component firmware 240, 242, 244.

[0059] To install the replacement component firmware 240, 242, 244, the recovery instructions 228 may store the replacement component firmware 240, 242, 244 at the medium 100 and set a flag indicating that new replacement component firmware 240, 242, 244 is available. The instructions 228 may then reset the computing device 200. After the reset, during a subsequent pre-boot, the BIOS 226 detects the flag, thus detecting the presence of the replacement component firmware 240, 242, 244, and installs the replacement component firmware 240, 242, 244.

[0060] FIG. 3 shows an example mapping 300 of component firmware identifiers to network addresses of replacement component firmware. The mapping 300 is illustrated as a table for sake of understanding. The mapping 300 may be implemented as database table, directory structure, or similar data structure, as discussed elsewhere herein. The mapping 300 is an example of the other mappings discussed herein, such as mapping 234.

[0061] The mapping correlates various combinations of computing device model 302, BIOS version 304, hardware component/device 306, and component firmware version 308 to uniquely definable download paths 310. Each download path 310 identifies a component firmware which may be used to replace a corrupted component firmware.

[0062] For example, the mapping 300 associates first replacement instructions of a first component firmware, as identified by hardware component 306 and firmware version 308, with a first network address 312 stored as a particular download path 310. Similarly, the mapping 300 associates different, second replacement instructions of a second component firmware, as identified by hardware component 306 and firmware version 308, with a second network address 314 stored as another particular download path 310. The network addresses 312, 314 may be at separate domains and may operate according to different protocols, such as Hypertext Transfer Protocol (HTTP), HTTP Secure (HTTPS), and File Transfer Protocol (FTP).

[0063] The mapping 300 may be updated as new models 302 with new/different components 306 are released. The mapping 300 may be updated as new BIOS versions 304 and component firmware versions 308 are released. Older information may be maintained, so that older computing devices may be supported.

[0064] A given set of component firmware, as identified by respective download paths 310, may be considered a best known configuration for the corresponding BIOS version 304 and model 302. An example of a best known configuration is shown at dashed box 316.

[0065] FIG. 4 shows an example recovery server 400 that provides network addresses of replacement component firmware in response to requests from computing devices, such as the computing device 200 of FIG. 2. The recovery server 400 may be used as the recovery server 232 of FIG. 2.

[0066] The recovery server 400 is a computing device that includes a processor 402, memory 404, chipset 406, network interface 408, and non-transitory machine-readable medium 410. Details of these components not repeated here may found elsewhere herein with regard to like-named components.

[0067] The medium 410 may store directing instructions 412 and a mapping 414.

[0068] The instructions 412 receive indications of corrupted component firmware from remote computing devices, via the network interface 408, and reference the mapping 414 to determine corresponding network addresses of replacement component firmware. The instructions 412 further transmit to the remote computing devices, via the network interface 408, indications of the network addresses.

[0069] The mapping 414 correlates identifiers of component firmware 416 to network addresses 418. As such, an indication of a corrupted component firmware received from a remote computing device may be used to obtain a network address 418 at which another server hosts the corresponding replacement component firmware indicated by the identifier 416.

[0070] A component firmware identifier 416 may include a version of the corrupted component firmware, a model of the remote computing device, and/or similar identifying data discussed elsewhere herein. Similar data may be received from a computing device requesting a network address 418 for a replacement component firmware.

[0071] The instructions 412 may further receive an indication of the network address 418 from a source of the replacement component firmware, such as a hardware manufacturer or vendor. Such indications may be provided by a form or other user interface element available to the source via a network. The instructions 412 may thus update the mapping 414 as new versions of component firmware are released or as storage locations of component firmware are changed.

[0072] The instructions 412 and mapping 414 may be implemented by a script and database, a directory structure, or similar technique.

[0073] FIG. 5 shows an example method 500 of obtaining replacement component firmware for component firmware detected to be corrupted. The

method 500 may be implemented as processor-executable instructions of a computing device, such as instructions of a BIOS.

[0074] At block 502, the BIOS is launched 502 and takes control of the computing device. Execution of the BIOS may include a Pre-EFI Initialization (PEI) phase and a DXE phase.

[0075] At block 504, the BIOS detects for corrupted component firmware by, for example, hardware interrupt, vendor-defined protocol, or similar signaling mechanism. This may occur during the DXE phase.

[0076] If no corrupted component firmware is identified, then the BIOS performs a boot device selection (BDS), at block 506, and completes its execution. The OS is then launched, at block 508. This concludes normal startup and the method 500 ends.

[0077] If a corrupted component firmware is detected, at block 510, the corrupted component firmware is identified by, for example, computing device model, hardware device identifier, component firmware version, and/or other data as discussed elsewhere herein. The identifying data may be added to a recovery list 512. The identifying data may also include an identifier or address of the respective hardware device for later reference when replacing component firmware.

[0078] Blocks 504 and 510 may be repeated for all component firmware of the computing device, via block 514, adding an identifier for any detected corrupted component firmware to the recovery list 512.

[0079] After all component firmware has been considered for corruption, such as by the BIOS reaching a specific point in execution, it is determined whether a network connection is available, at block 516.

[0080] If no network connection is available, the BIOS outputs a user alert, at block 518. An example user alert includes a dialog box or other user interface notification, to warn the user that a corrupted firmware component has been

detected. The contents of the recovery list 512 may be displayed to the user. The BIOS then completes execution, at block 506.

[0081] If a network connection is available, replacement component firmware corresponding to the corrupted component firmware is identified, at block 520. This may include transmitting, via the network, an identifier to a server that hosts a mapping 522 of component firmware identifier to network address and receiving, in response, an indication of a specific network address for a specific identifier of the corrupted component firmware. This may be done for each element of corrupted component firmware on the recovery list 512.

[0082] Then, at block 524, replacement component firmware is downloaded from a component firmware server 526 identified by an obtained network address. This may be done for each network address obtained at block 520 for each item of corrupted component firmware. Different servers 526 may be accessed.

[0083] At block 528, the downloaded replacement component firmware is stored in non-volatile memory. The recovery list 512 may also be stored in non-volatile memory, so as to indicate to the BIOS during next execution which component firmware has replacement component firmware to be installed. Alternatively, a simple flag may be set in non-volatile memory to inform the BIOS to check a location of non-volatile memory reserved for replacement component firmware.

[0084] Then, at block 530, the BIOS triggers a reset of the computing device.

[0085] FIG. 6 shows an example method 600 of installing replacement component firmware to replace component firmware detected to be corrupted. The method 600 may be implemented as processor-executable instructions of a computing device, such as instructions of a BIOS.

[0086] At block 502, the BIOS is launched 502 and takes control of the computing device, as discussed with respect to the method 500.

[0087] At block 602, the presence of any replacement component firmware is detected. This may include checking for a recovery list 512 and/or replacement component firmware in non-volatile memory. Alternatively, a flag may be checked. If no replacement component firmware is found to be present, then BIOS execution completes and the OS is launched at blocks 506 and 508, as discussed with respect to the method 500.

[0088] If replacement component firmware is detected, then recovery is performed at block 604. Recovery may include copying replacement component firmware to non-volatile storage associated with the respective hardware device. Such storage may be part of the hardware device. The recovery list 512 may be referenced to identify the respective hardware device.

[0089] Recovery is performed for each element on the recovery list 512, via block 604.

[0090] After all elements of replacement component firmware have been installed, then the downloaded copy of the replacement component firmware is deleted from non-volatile memory. The recovery list 512 is also deleted. Alternatively, a flag may be cleared. This prevents superfluous detection at next execution of block 602.

[0091] Then, a user notification may be issued, at block 610. A dialog box or similar user interface element may be displayed to the user to inform the user of the recovery and the affected items of component firmware and/or underlying hardware devices. Then, BIOS execution completes and the OS is launched at blocks 506 and 508.

[0092] The method 500 of detecting and obtaining replacement component firmware and the method 600 of installing obtained replacement component firmware may be implemented together in a BIOS or other set of instructions of a computing device. In various examples, execution of the methods 500, 600 may be coordinated, so that corrupted component firmware is replaced by

previously obtained replacement component firmware before detection is performed.

[0093] FIG. 7 shows an example method of obtaining replacement component firmware for component firmware detected to be corrupted, and installing the replacement component firmware. The method 700 may be implemented as processor-executable instructions of a computing device, such as instructions of a BIOS. Discussion of details already explained is not repeated here, and the description above may be referenced for like-numbered blocks.

[0094] BIOS execution begins at block 502.

[0095] At block 602, when replacement component firmware is detected, recovery is performed, at block 702, with reference to a recovery list 512. Block 702 may implement blocks 604, 606, 608, and 610 discussed above.

[0096] Next, at block 504, when corrupted component firmware is detected, replacement component firmware is obtained from a server or servers 526 with reference to a mapping 522 of component firmware identifier to network address, at block 704. A recovery list 512 may be generated. Block 704 may implement blocks 510, 514, 516, 520, 524, and 528 discussed above.

[0097] Subsequent to block 704, the computing device is reset at block 530, so that the method 700 is repeated. During next execution of the method 700, block 602 detects replacement component firmware and recovery is performed. Further, it is likely that block 504 will not detect further corrupted component firmware, hence, the BIOS will complete execution and the OS will be launched at blocks 506 and 508. If block 504 does detect further corrupted component firmware, the method 700 may continue for another cycle of obtaining replacement component firmware and resetting the computing device. The number of cycles may be limited to prevent an infinite loop and a user alert may be issued if the limit is exceeded.

[0098] In view of the above, it should be apparent that corrupted component firmware may be recovered without user or OS intervention. User experience may be improved, in that the user need not manually recover corrupted component firmware themselves, contact a service technician, or wait for the OS to launch to correct corrupted component firmware. Moreover, a specific OS that supports OS-based recovery need not be installed. Other operating systems including cloud-based operating systems are compatible with the techniques discussed herein. In addition, non-volatile storage space at the computing device may be saved by avoiding storing backup copies of component firmware locally. Further, the centralization of the mapping and distributed sources of replacement component firmware is highly scalable and relatively simple to keep updated, so that BKCs for numerous different configurations of computing devices may be readily maintained.

[0099] It should be recognized that features and aspects of the various examples provided above can be combined into further examples that also fall within the scope of the present disclosure. In addition, the figures are not to scale and may have size and shape exaggerated for illustrative purposes.

CLAIMS

1. A non-transitory machine-readable medium comprising recovery instructions that, when executed by a processor, cause the processor to:

detect a corruption of first installed instructions of a first component firmware at a computing device and second installed instructions of a second component firmware at the computing device;

in response to detection of the corruption during pre-boot:

if the first installed instructions are detected as corrupt, request and receive first replacement instructions from a first network address, and replace in the computing device the first installed instructions with the first replacement instructions; and

if the second installed instructions are detected as corrupt, request and receive second replacement instructions from a second network address different from the first network address, and replace in the computing device the second installed instructions with the second replacement instructions.

2. The non-transitory machine-readable medium of claim 1, wherein the instructions are further to query a mapping that associates the first replacement instructions with the first network address and that associates the second replacement instructions with the second network address.

3. The non-transitory machine-readable medium of claim 2, wherein the mapping is maintained remotely and the recovery instructions are to query the mapping via a network.

4. The non-transitory machine-readable medium of claim 2, wherein the mapping stores best known configurations (BKCs) of component firmware of different computing devices.

5. The non-transitory machine-readable medium of claim 1, wherein the first installed instructions and first replacement instructions provide a first

functionality to the computing device that is different from a second functionality provided by the second installed instructions and second replacement instructions.

6. The non-transitory machine-readable medium of claim 1, wherein the first network address is in a first domain and the second network address is in a second domain that is separate from the first domain.

7. The non-transitory machine-readable medium of claim 1, wherein the first network address is accessible by a first protocol and the second network address is accessible by a second protocol that is different from the first protocol.

8. A non-transitory machine-readable medium comprising instructions that, when executed by a processor, cause the processor to:

monitor a computing device to detect corruption of component firmware; and

in response to detection of the corruption:

identify the component firmware that has the corruption;

query a remotely maintained mapping that associates replacement component firmware with network addresses to obtain a network address;

request and receive replacement component firmware for the corrupted component firmware from the network address; and

replace in the computing device the corrupted component firmware with the replacement component firmware.

9. The non-transitory machine-readable medium of claim 8, wherein the instructions are executed during pre-boot of the computing device.

10. The non-transitory machine-readable medium of claim 8, wherein the instructions are further to:

reset the computing device after receiving the replacement component firmware; and

replace in the computing device the corrupted component firmware with the replacement component firmware after the reset.

11. The non-transitory machine-readable medium of claim 10, wherein the instructions are further to detect a presence of the replacement component firmware during boot and, in response, replace in the computing device the corrupted component firmware with the replacement component firmware.

12. A computing device comprising:

a network interface;

a processor connected to the network interface, the processor to:

receive from a remote computing device via the network interface an indication of a corrupted component firmware of the remote computing device;

determine a network address for a replacement component firmware by referencing a mapping of identifiers of component firmware to network addresses; and

transmit to the remote computing device via the network interface an indication of the network address.

13. The computing device of claim 12, wherein the processor is further to:

determine the network address for the replacement component firmware based on a version of the corrupted component firmware.

14. The computing device of claim 12, wherein the processor is further to:

receive from the remote computing device an identifier of a model of the remote computing device; and

determine the network address for the replacement component firmware based on the model of the remote computing device.

15. The computing device of claim 12, wherein the processor is further to receive an indication of the network address from a source of the replacement component firmware.

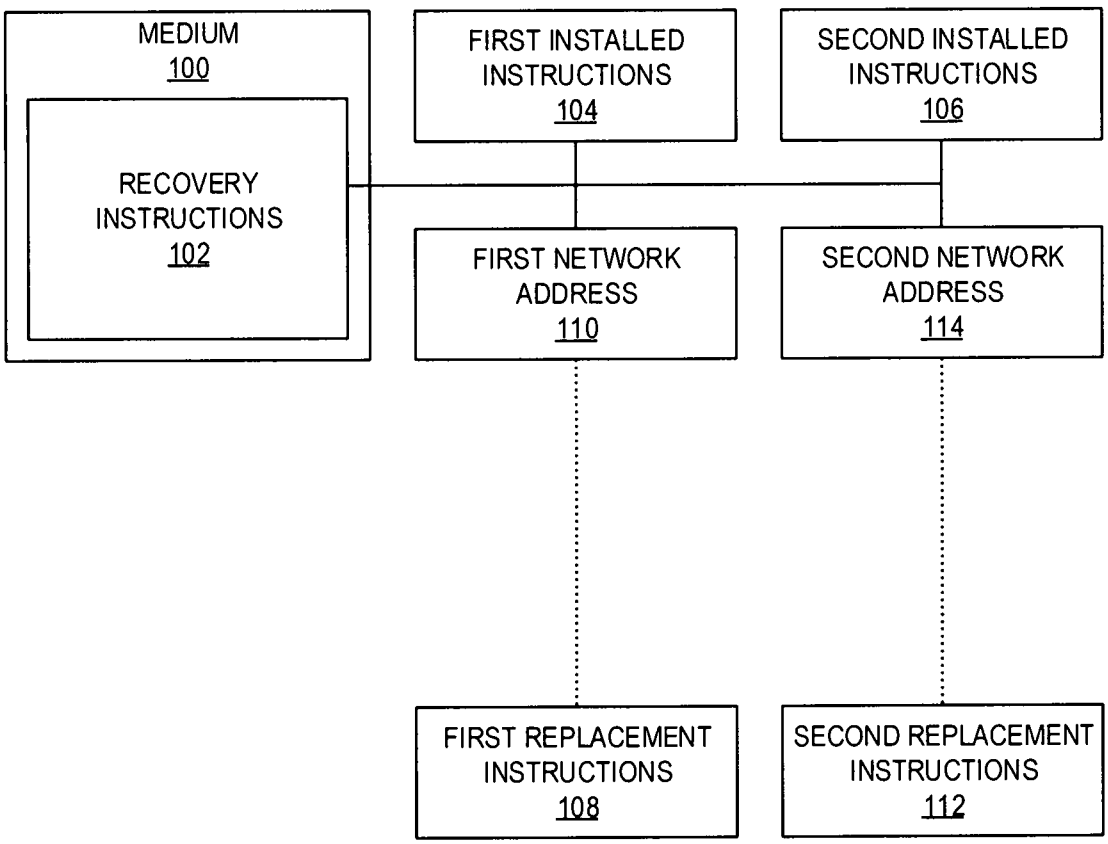
MEDIUM
100
RECOVERY INSTRUCTIONS
102
FIRST INSTALLED INSTRUCTIONS
104
SECOND INSTALLED INSTRUCTIONS
106
FIRST NETWORK ADDRESS
110
SECOND NETWORK ADDRESS
114
FIRST REPLACEMENT INSTRUCTIONS
108
SECOND REPLACEMENT INSTRUCTIONS
112

FIG. 1

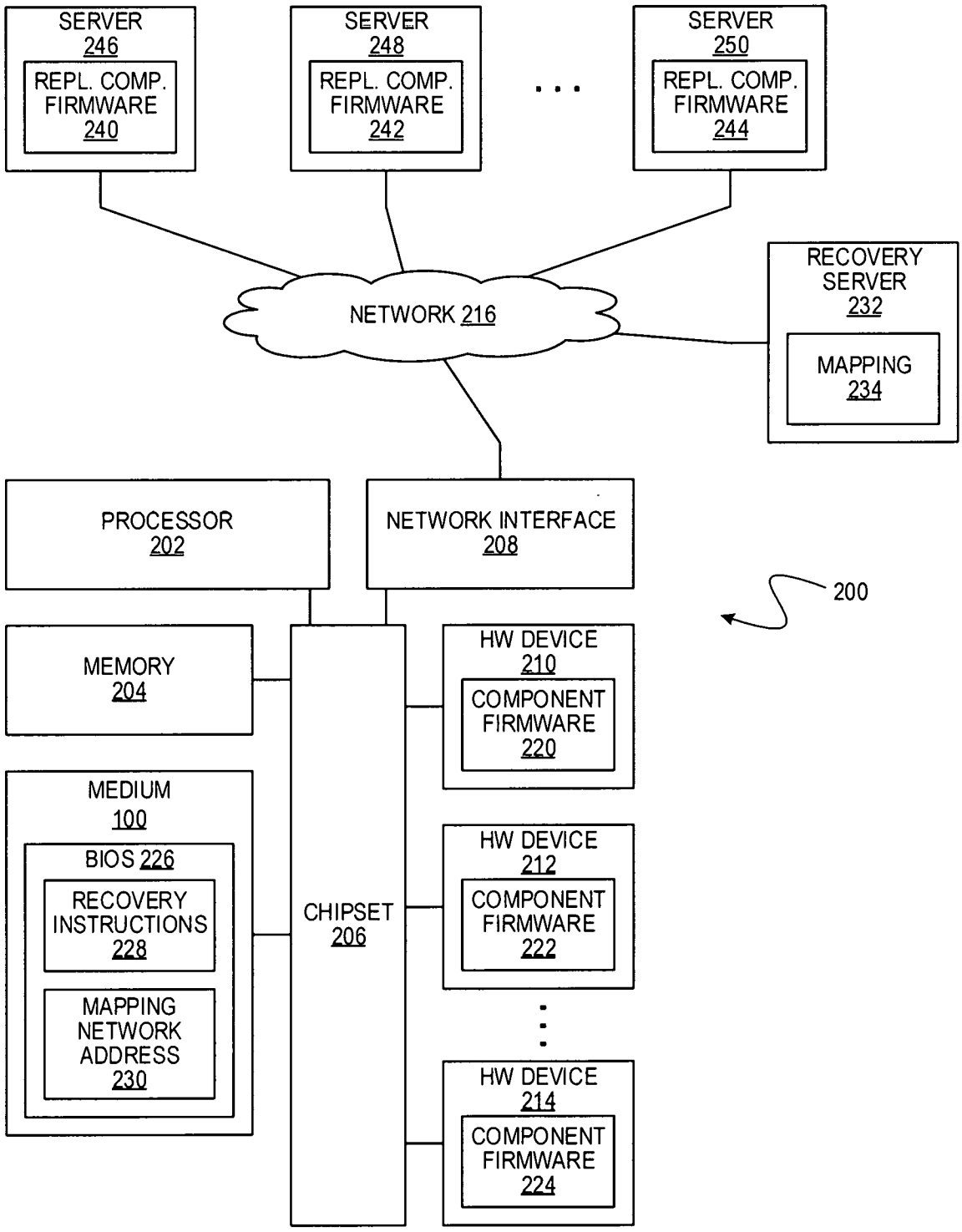
SERVER
246
REPL. COMP. FIRMWARE
240
SERVER
248
REPL. COMP. FIRMWARE
242
SERVER
250
REPL. COMP. FIRMWARE
244
NETWORK 216
RECOVERY SERVER
232
MAPPING
234
PROCESSOR
202
NETWORK INTERFACE
208
200
MEMORY
204
MEDIUM
100
BIOS 226
RECOVERY INSTRUCTIONS
228
MAPPING NETWORK ADDRESS
230
CHIPSET
206
HW DEVICE
210
COMPONENT FIRMWARE
220
HW DEVICE
212
COMPONENT FIRMWARE
222
HW DEVICE
214
COMPONENT FIRMWARE
224

FIG. 2

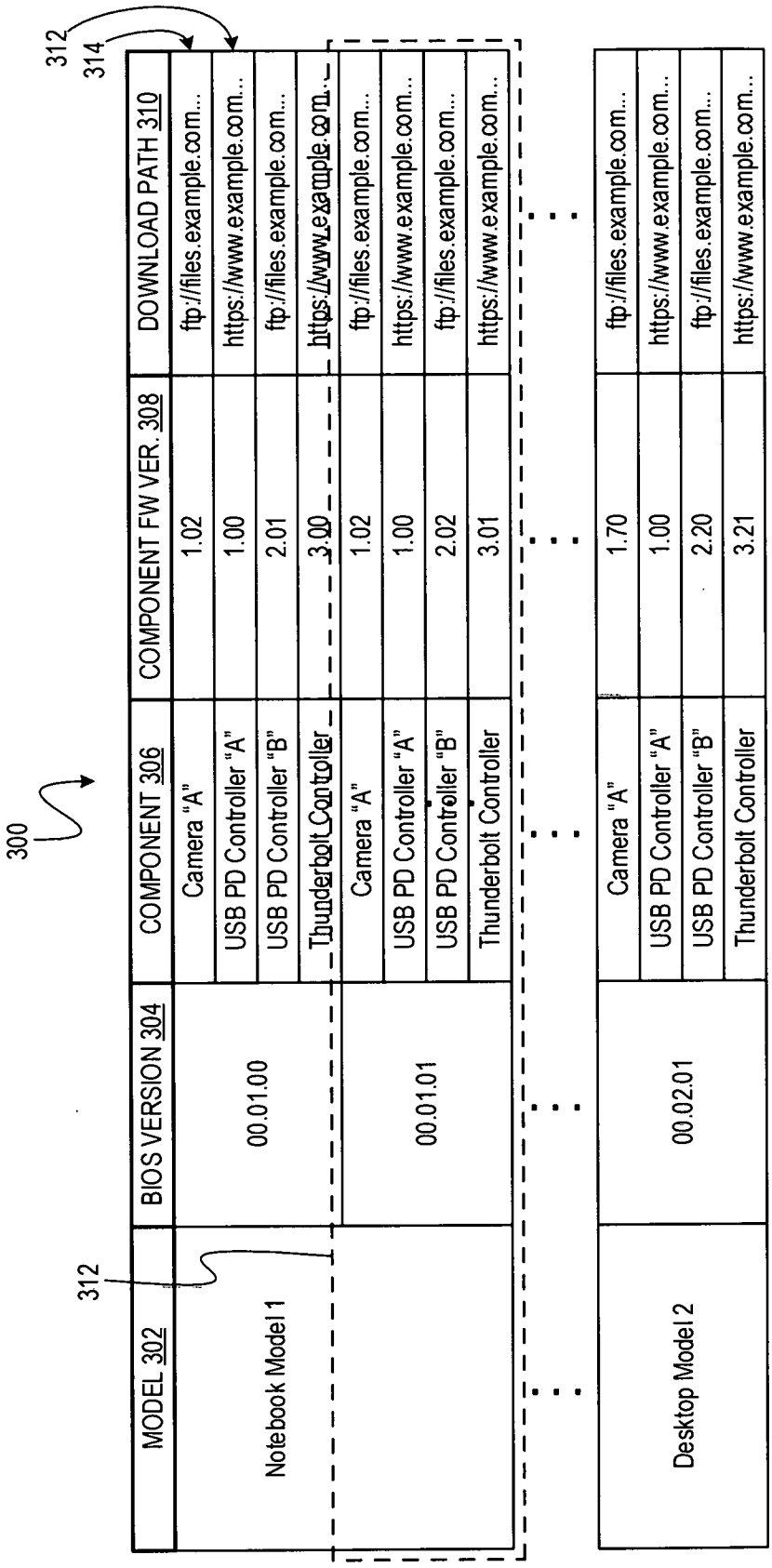
300
312
312
314
MODEL 302
BIOS VERSION 304
COMPONENT 306
COMPONENT FW VER. 308
DOWNLOAD PATH 310
Notebook Model 1
00.01.00
Camera "A"
1.02
ftp://files.example.com...
USB PD Controller "A"
1.00
https://www.example.com...
USB PD Controller "B"
2.01
ftp://files.example.com...
Thunderbolt Controller
3.00
https://www.example.com...
00.01.01
Camera "A"
1.02
ftp://files.example.com...
USB PD Controller "A"
1.00
https://www.example.com...
USB PD Controller "B"
2.02
ftp://files.example.com...
Thunderbolt Controller
3.01
https://www.example.com...
Desktop Model 2
00.02.01
Camera "A"
1.70
ftp://files.example.com...
USB PD Controller "A"
1.00
https://www.example.com...
USB PD Controller "B"
2.20
ftp://files.example.com...
Thunderbolt Controller
3.21
https://www.example.com...

FIG. 3

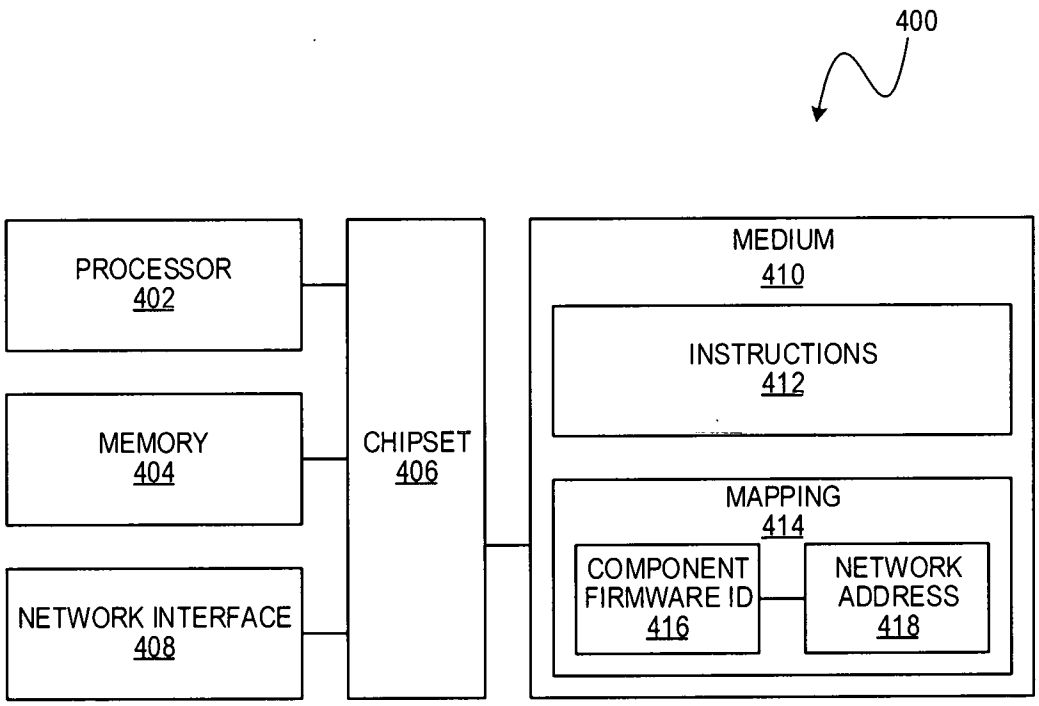
400
PROCESSOR
402
MEMORY
404
NETWORK INTERFACE
408
CHIPSET
406
MEDIUM
410
INSTRUCTIONS
412
MAPPING
414
COMPONENT FIRMWARE ID
416
NETWORK ADDRESS
418

FIG. 4

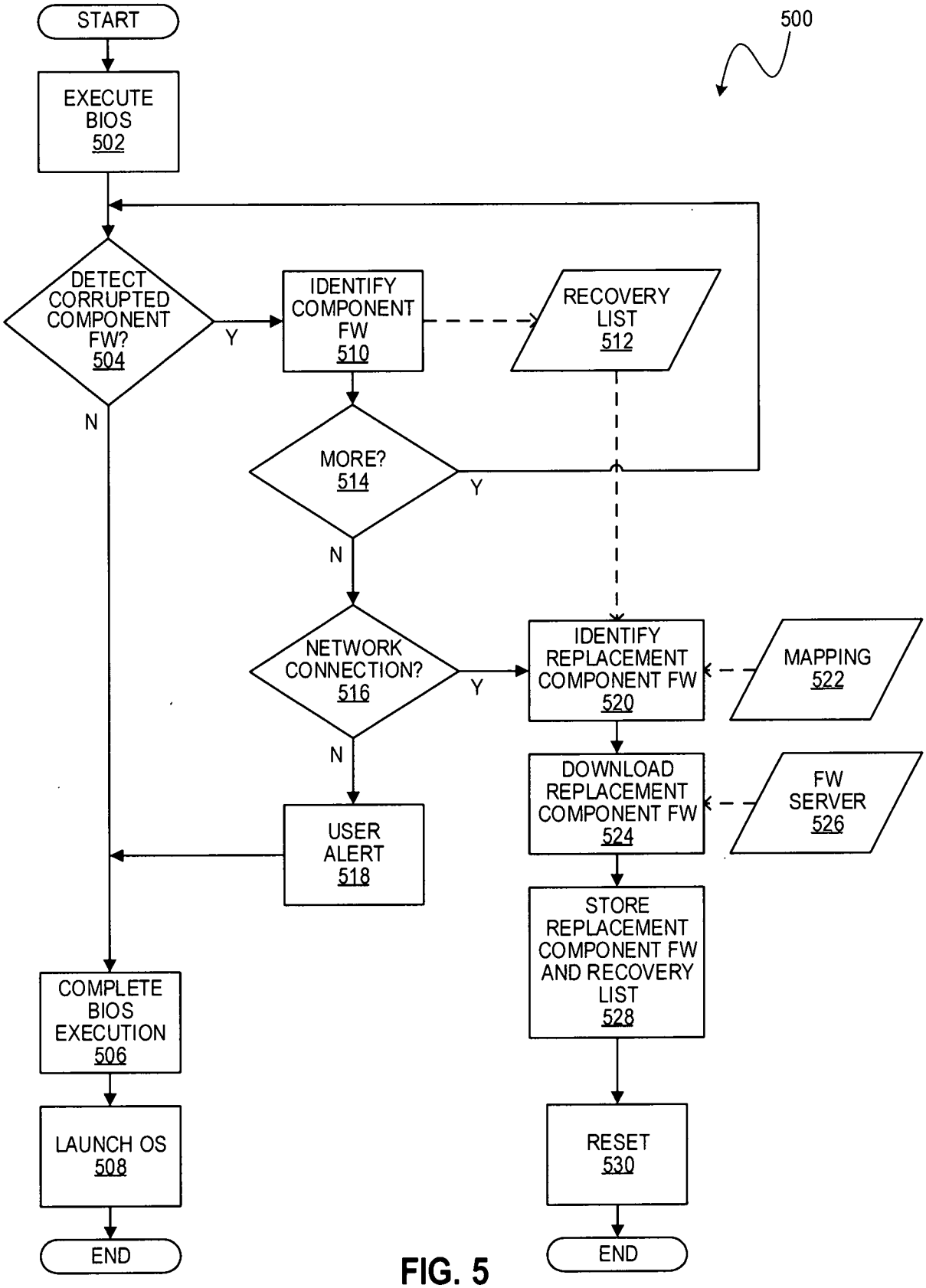
START
EXECUTE BIOS 502
500
DETECT CORRUPTED COMPONENT FW? 504
Y
N
IDENTIFY COMPONENT FW 510
RECOVERY LIST 512
MORE? 514
Y
N
NETWORK CONNECTION? 516
Y
N
IDENTIFY REPLACEMENT COMPONENT FW 520
MAPPING 522
DOWNLOAD REPLACEMENT COMPONENT FW 524
FW SERVER 526
STORE REPLACEMENT COMPONENT FW AND RECOVERY LIST 528
RESET 530
END
USER ALERT 518
COMPLETE BIOS EXECUTION 506
LAUNCH OS 508
END

FIG. 5

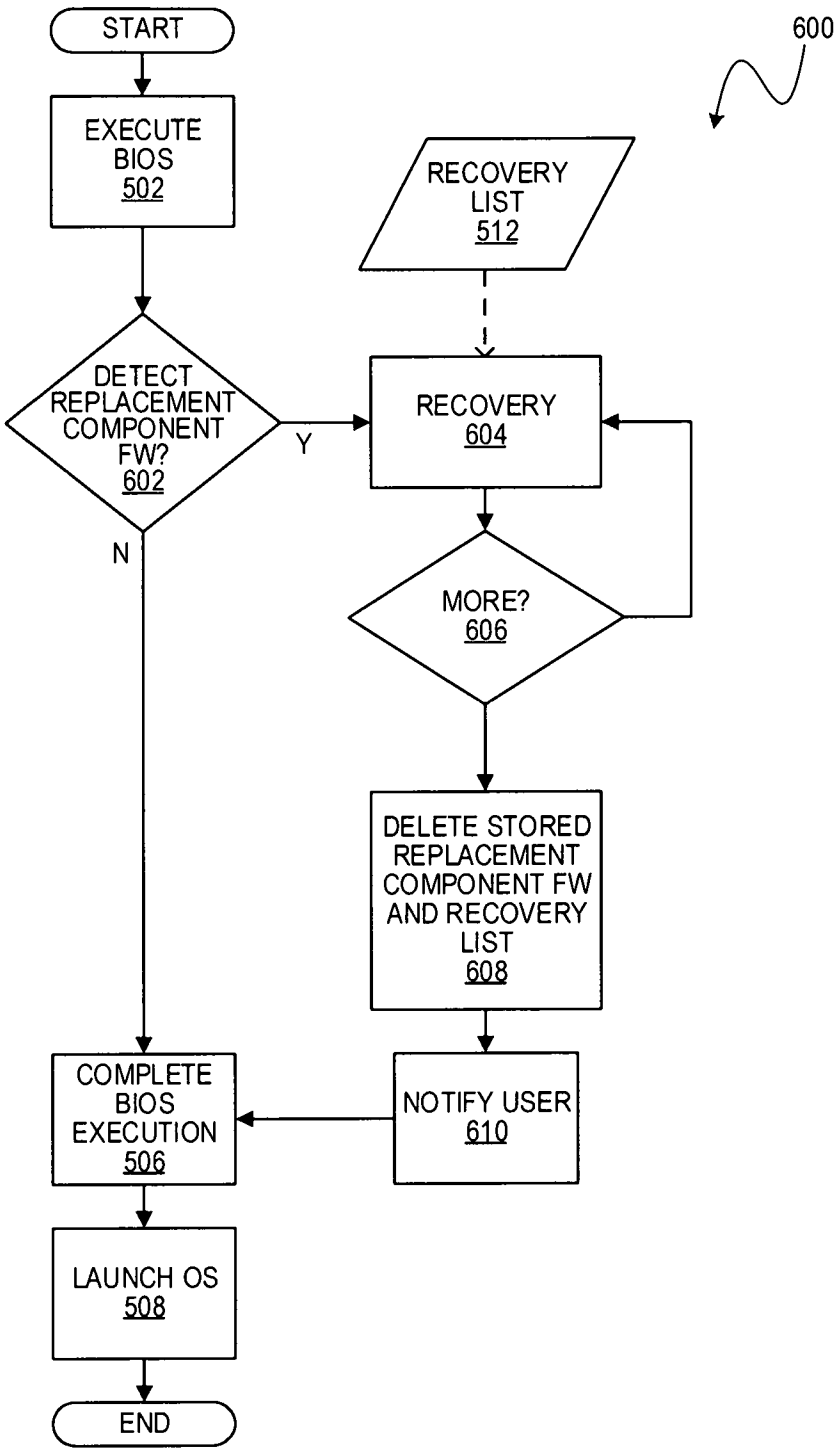
600
START
EXECUTE BIOS 502
RECOVERY LIST 512
DETECT REPLACEMENT COMPONENT FW? 602
Y
N
RECOVERY 604
MORE? 606
DELETE STORED REPLACEMENT COMPONENT FW AND RECOVERY LIST 608
NOTIFY USER 610
COMPLETE BIOS EXECUTION 506
LAUNCH OS 508
END

FIG. 6

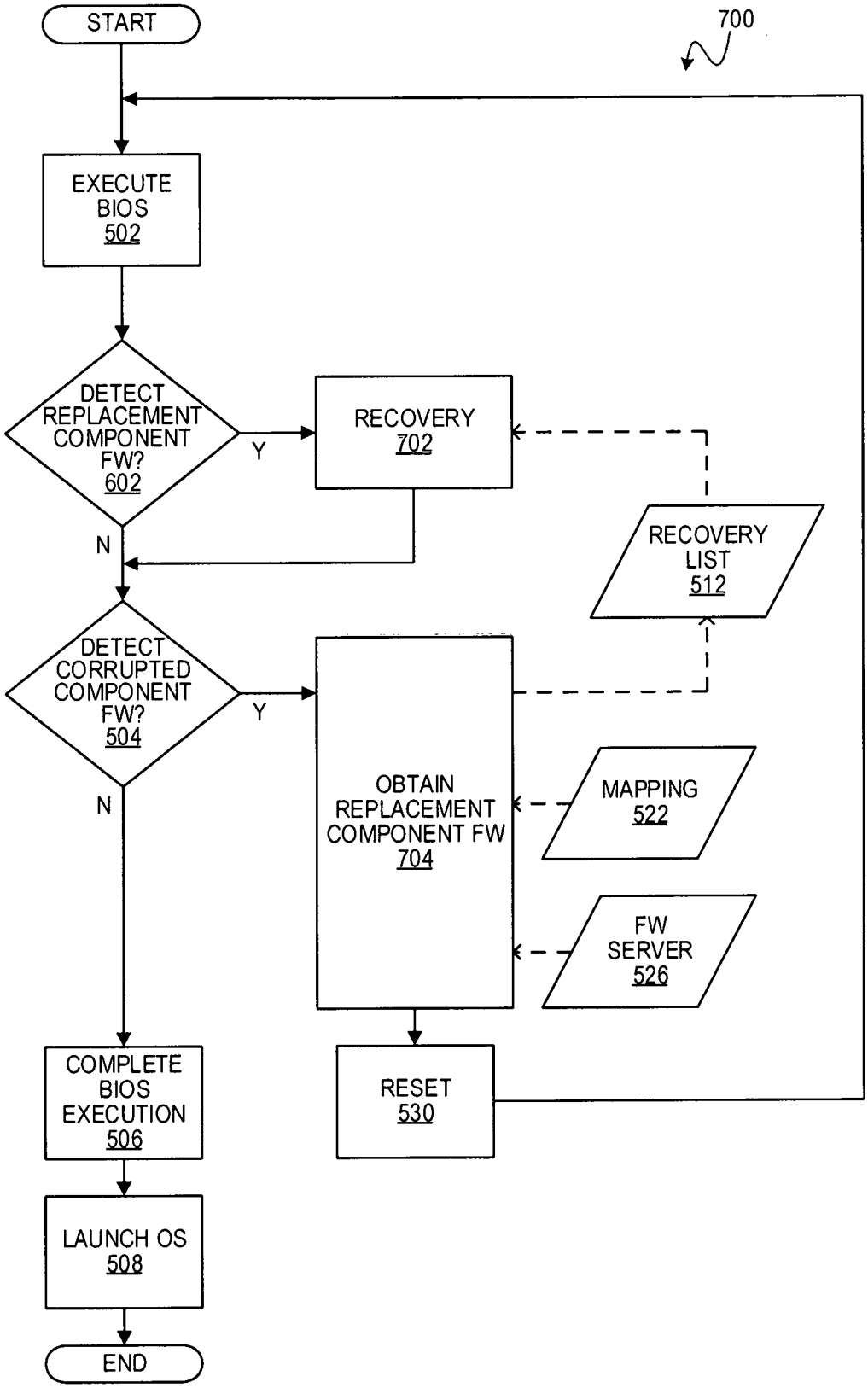
START
700
EXECUTE BIOS 502
DETECT REPLACEMENT COMPONENT FW? 602
Y
RECOVERY 702
N
RECOVERY LIST 512
DETECT CORRUPTED COMPONENT FW? 504
Y
OBTAIN REPLACEMENT COMPONENT FW 704
MAPPING 522
FW SERVER 526
N
COMPLETE BIOS EXECUTION 506
RESET 530
LAUNCH OS 508
END

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2021/038011

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/14 *(2006.01)*
G06F 11/36 *(2006.01)*

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)

G06F 11/00, 11/14, 11/36

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)

PatSearch (RUPTO Internal), USPTO, PAJ, Espacenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/0103347 A1 (SNEED G. CHRISTOPHER et al.) 27.05.2004, paragraphs [0004]-[0007], [0009], [0013], [0014], [0018], [0020], [0036], [0040], [0043], [0044], [0046]-[0048], [0050], [0053], [0056], [0057], [0060], [0062]	1-3, 5, 7-15
Y		4, 6
Y	US 2021/0240589 A1 (DELL PRODUCTS LP) 05.08.2021, paragraphs [0018], [0023]	4
Y	US 2018/0287996 A1 (HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP) 04.10.2018,paragraphs [0014], [0015].	6
A	US 2016/0098561 A1 (NOKOMIS, INC) 07.04.2016	1-15
A	US 2016/0373944 A1 (ORCHESTRA TECHNOLOGY, INC) 22.12.2016	1-15

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

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier document 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 October 2021 (28.10.2021)	18 November 2021 (18.11.2021)

Name and mailing address of the ISA/RU:	Authorized officer
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	K. Blednov Telephone No. 8(495)531-64-81