



- (51) International Patent Classification:
G06F 9/445 (2006.01)
- (21) International Application Number:
PCT/CN2011/070018
- (22) International Filing Date:
4 January 2011 (04.01.2011)
- (25) Filing Language:
English
- (26) Publication Language:
English
- (71) Applicant (for all designated States except US): **MOTOROLA MOBILITY, INC.** [US/US]; 600 North US Highway 45, Libertyville, Illinois 60048 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **ZHAO, Liang** [CN/CN]; Motorola Beijing, No. 1 Wang Jing East Road, Chao Yang District, Beijing 100102 (CN). **HUANG, Bo** [CN/CN]; No. 108 Jian Guo Road, Chao Yang District, Beijing 100022 (CN). **LIN, Hai-Tao** [CN/CN]; No. 108 Jian Guo Road, Chao Yang District, Beijing 100022 (CN). **YU, Lin-Lin** [CN/CN]; Motorola Beijing, No. 1 Wang Jing East Road, Chao Yang District, Beijing 100102 (CN).
- (74) Agent: **CHINA SINDA INTELLECTUAL PROPERTY LIMITED**; B11th Floor, Focus Place, 19 Financial Street, Beijing 100033 (CN).

- (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, TH, 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, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(54) Title: HYBRID OPERATING SYSTEM MEDIA INTEGRATION

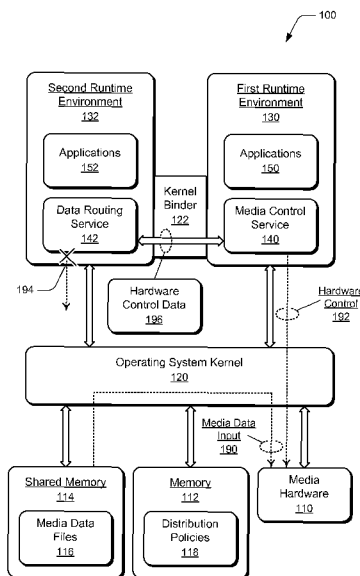


FIG. 1

(57) Abstract: In embodiments of hybrid operating system media integration, a media control service in a first runtime environment manages the distribution of a media data file from the first runtime environment to media hardware that renders the media data. A data routing service in a second runtime environment routes an additional media data file and hardware control data from the second runtime environment to the media control service in the first runtime environment. The media control service further manages distribution of the additional media data file to the media hardware.

WO 2012/092706 A1

HYBRID OPERATING SYSTEM MEDIA INTEGRATION

BACKGROUND

[0001] Mobile devices, such as portable computers and mobile phones, may be implemented with a hybrid operating system that includes multiple runtime environments operating with one operating system kernel. Unlike a virtual machine that emulates a computing device within another controlling operating system, the runtime environments of a hybrid operating system can each access and control device hardware. For example, in a LinuxTM hybrid operating system, an AndroidTM application in a first runtime environment and an UbuntuTM application in a second runtime environment may both access and control audio hardware with independent hardware control signals via the Linux Kernel. This may result in an audio conflict, such as in a mobile phone when audio for an incoming phone call in the Android environment overlaps or conflicts with audio for a multimedia video file in the Ubuntu environment.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Embodiments of hybrid operating system media integration are described with reference to the following drawings. The same numbers are used throughout the drawings to reference like features and components:

FIG. 1 illustrates an example system in which embodiments of hybrid operating system media integration can be implemented.

FIG. 2 further illustrates components of the example system in embodiments of hybrid operating system media integration.

FIG. 3 illustrates an example electronic device in which embodiments of hybrid operating system media integration can be implemented.

FIG. 4 illustrates example method(s) of hybrid operating system media integration in accordance with one or more embodiments.

FIG. 5 illustrates various components of an example device that can implement embodiments of hybrid operating system media integration.

DETAILED DESCRIPTION

[0003] In embodiments of hybrid operating system media integration, an electronic device can be implemented with a hybrid operating system, such as a GNU/Linux operating system that includes a Linux Kernel, an Android™ runtime environment, and a separate GNU/Linux runtime environment. The operating system kernel communicates and/or transfers data between the runtime environments and the memory, processors, and media hardware of the device. A media control service in the Android runtime environment manages and controls the distribution of media data, such as audio data and/or video data, from applications in the Android runtime environment to media hardware that renders the media data. The media hardware may include speakers to render audio and/or a display component (or monitor) to display video. A data routing service in the GNU/Linux runtime environment routes additional media data files and hardware control data from the GNU/Linux runtime

environment to the media control service in the Android runtime environment. The media control service in the Android runtime environment then manages and controls the distribution of the additional media data to the media hardware.

[0004] In an example, an Ubuntu MP3 player (*e.g.*, an application in the GNU/Linux environment) may be playing an audio file at a background audio level. If the GNU/Linux runtime environment has access to control the audio hardware in an electronic device (*e.g.*, a mobile phone) as described above in the Background section, then an audio controller in the GNU/Linux runtime environment can send control signals to turn a device speaker on and set the volume to the background audio level to playback the audio file. When an incoming phone call is received by a call application (*e.g.*, an application in the Android runtime environment), a user of the device may initiate turning off the Ubuntu MP3 player to take the phone call. However, turning off the Ubuntu MP3 player may trigger a control signal to turn the device speaker off, while the Android call application attempts to initiate turning the device speaker on and setting a particular volume for an audible phone call ring alert. The embodiments of hybrid operating system media integration, as described herein, preempts this type of audio conflict as well as other media hardware conflicts, where two or more software applications in different runtime environments attempt to control the media hardware in an electronic device to render or playback media data.

[0005] While features and concepts of hybrid operating system media integration can be implemented in any number of different devices, systems, environments, and/or configurations, embodiments of hybrid operating system media integration are described in the context of the following example devices, systems, and methods.

[0006] FIG. 1 illustrates an example system 100 in which embodiments of hybrid operating system media integration can be implemented. The example system 100 may be implemented as a variety of mobile or non-mobile environments in any type of fixed or mobile device. For example, a dual runtime environment may be implemented as a gaming and computing device, as an electronic book (eBook) and television media device, as a mobile phone and computing device, or as any other set of two or more runtime environments. Additionally, the example system can be implemented as an electronic device with any combination of differing components as further described with reference to the example electronic device shown in FIG. 5.

[0007] The example system 100 may be generally described with reference to layers of abstraction, such as in an electronic and/or computing device. The system includes media hardware 110 that renders and/or captures media data, such as audio data and/or video data. The media hardware may include an audio speaker that renders audio when input with audio data, and may include a display component that displays video when input with video data. The media hardware can include various audio speakers, such as a private speaker, a hands-free speaker, a wireless earpiece speaker with a Bluetooth transceiver, a headphone jack, and the like. The media hardware may also include a microphone for capturing audio signals, a camera for capturing video or still picture signals, vibration devices for capturing vibrational signals, haptic feedback devices for generating vibrations when input with haptic data, visual outputs, and/or any other media receiving and/or rendering device.

[0008] In this example, the system also includes memory 112 and shared memory 114, each implemented to store or otherwise maintain various data. For

example, the shared memory 114 stores media data files 116, such as audio data files and/or video data files. The memory 112 stores media distribution policies 118 that can be used to manage distribution of the media data files to the media hardware. The memory and shared memory can be implemented as any type of memory and/or suitable electronic data storage including memory units removable from the electronic device, such as a flash memory data storage device (*e.g.*, USB flash drive), a compact disk (CD), or a digital versatile disk (DVD).

[0009] The example system 100 includes elements of a hybrid operating system, such as an operating system kernel 120 and a kernel binder 122. The system also includes a first runtime environment 130 and a second runtime environment 132. In implementations, a LinuxTM hybrid operating system may be implemented as a GNU/Linux operating system that includes a Linux Kernel (*e.g.*, the operating system kernel), an AndroidTM runtime environment (*e.g.*, the first runtime environment), and a GNU/Linux runtime environment (*e.g.*, the second runtime environment). The operating system kernel 120 communicates and/or transfers data between the runtime environments and the memory, processors, and media hardware of the example system. Embodiments of a hybrid operating system are not limited to Linux-based systems and environments. Other operating systems that may be used to implement hybrid operating system media integration could include Disk Operating System (DOS), Microsoft WindowsTM operating systems, and Apple Mac OSTM operating systems. Additionally, alternate implementations can include more than two runtime environments of a hybrid operating system.

[0010] The first runtime environment 130 includes a media control service 140 that can be implemented as computer-executable instructions, such as a

software application, and executed by one or more processors to implement the various embodiments described herein. In embodiments, the media control service can be implemented as an Android Audio Flinger service (*e.g.*, in an Android runtime environment) for audio management and distribution of audio data files to the media hardware. The second runtime environment 132 includes a data routing service 142 that can also be implemented as computer-executable instructions, such as a software application, and executed by one or more processors to implement the various embodiments described herein. In embodiments, the data routing service can be implemented as an Advanced Linux Sound Architecture (ALSA) service (*e.g.*, in a GNU/Linux runtime environment).

[0011] The first runtime environment 130 also includes software applications 150, such as Android applications, and the second runtime environment 132 also includes software applications 152, such as Ubuntu™ applications. The software applications in their respective runtime environments may include, but are not limited to, any one or combination of a call/dialer application, camera (video or still images), audio recorder, MP3 player, WAVE audio file player, TIFF still images viewer, AVI audio-video player, Flash audio-video player, QuickTime audio-video player, or MP4 audio-video player. Depending on specific implementations and functions of the various software applications, the applications generally retrieve and/or generate media data, as well as retrieve and/or generate hardware control data that controls the media hardware 110 to render the corresponding media data, such as audio data and/or video data. The media data that is retrieved and/or generated by the software applications can be stored or otherwise maintained in the shared memory 114 as the media data files 116. The hardware

control data may include data signals to turn the media hardware on and off, set a volume level or gain, set a brightness level, mix multiple media data files, turn muting on and off, and any other type of hardware control data signals.

[0012] The media control service 140 in the first runtime environment 130 manages the distribution of media data files from the software applications 150 in the first runtime environment to the media hardware 110. Media data from the media data files 116 is routed for input to the media hardware at 190, and the media control service communicates hardware control signals 192 through the operating system kernel 120 to initiate and control the media hardware that renders the media data. The data routing service 142 in the second runtime environment 132 is implemented to preempt direct distribution of media data files and hardware control data from the software applications 152 in the second runtime environment to the media hardware at 194. Note that, although a playback example is shown here, hardware control signals 192 can also be used to control media hardware 110 to capture media data and store it in the shared memory 114.

[0013] In embodiments, the data routing service 142 is implemented to route additional media data files from the second runtime environment 132 to the media control service 140 in the first runtime environment 130 where the media control service also manages the distribution of the additional media data files to the media hardware 110. This prevents a media hardware control conflict and/or a media rendering conflict if both the first runtime environment 130 and the second runtime environment 132 were to initiate the same media hardware for media playback, or if one runtime environment 130 were to turn off the media hardware when the other

runtime environment 132 is using or initiates use of the media hardware. Not only can playback conflicts be avoided, but recording conflicts can also be avoided.

[0014] The data routing service 142 in the second runtime environment 132 utilizes the shared memory 114 to store the additional media data files from the applications 152, and the additional media data files are then accessible to the operating system (*e.g.*, the operating system kernel 120) from the shared memory. The data routing service 142 is also implemented to route hardware control data 196 that is associated with the additional media data files from the second runtime environment to the media control service 140 in the first runtime environment. The hardware control data is routed via the kernel binder 122 between the second runtime environment and the first runtime environment. The media control service 140 can then selectively initiate the hardware control signals 192 from the hardware control data 196 to manage the distribution of the media data files 116 that are routed from the second runtime environment by the data routing service.

[0015] FIG. 2 further illustrates components 200 of the example system 100 as described with reference to FIG. 1. The media control service 140 includes code libraries 230. As described above, the media control service may be implemented as an Android Audio Flinger service (*e.g.*, in an Android runtime environment). In an example implementation of an Android runtime environment, the code libraries 230 include “libmedia” 232, “libcutils” 233, “libbinder” 234, “liblog” 235, and “libutils” 236. The data routing service 142 can be implemented with one or more of the code libraries of the media control service at 238. As described above, the data routing service may be implemented as an Advanced Linux Sound Architecture

(ALSA) (e.g., in a GNU/Linux runtime environment). The Android Audio Flinger libraries are ported to the Linux/Ubuntu environment, and a new ALSA PCM plug-in (pulse code modulation for audio data) and a new ALSA CTL plug-in (media hardware control) can be built from the Android code libraries.

[0016] FIG. 3 illustrates an example of an electronic device 300 in which embodiments of hybrid operating system media integration can be implemented. The electronic device 300 can be implemented as any type of fixed or mobile device, and can be implemented with any combination of differing components as further described with reference to the example electronic device shown in FIG. 5. Additionally, the electronic device 300 includes multiple runtime environments, such as described with reference to FIG. 1.

[0017] For example, the electronic device 300 includes media hardware 310 that renders media data, such as audio data and/or video data. The media hardware may include an audio speaker that renders audio when input with audio data, and may include a display component that displays video when input with video data. Note that media hardware 310 may be integrated into the electronic device 300 or may be a peripheral connected through a port or through a wireless transceiver. The media hardware in the electronic device can include various audio speakers, such as a private speaker, a hands-free speaker, a wireless earpiece speaker with a Bluetooth transceiver, a headphone jack output, and the like. The media hardware may also include a microphone, a camera, vibration devices, haptic feedback devices, visual outputs, and/or any other media receiving and/or rendering device.

[0018] The electronic device 300 also includes memory 312 and shared memory 314, each implemented to store or otherwise maintain various data. For example, the shared memory 314 stores media data files 316, such as audio data files and/or video data files. The memory 312 stores media distribution policies 318 that can be used to manage distribution of the media data files to the media hardware. The memory and shared memory can be implemented as any type of memory and/or suitable electronic data storage as described with reference to FIG. 1.

[0019] The electronic device 300 includes elements of a hybrid operating system 320, such as an operating system kernel 322 and a kernel binder 324. The hybrid operating system includes a first runtime environment 330 and a second runtime environment 332. In implementations, the hybrid operating system may be implemented as a GNU/Linux operating system that includes a Linux Kernel (*e.g.*, the operating system kernel), an Android™ runtime environment (*e.g.*, the first runtime environment), and a GNU/Linux runtime environment (*e.g.*, the second runtime environment). The operating system kernel 322 communicates and/or transfers data between the runtime environments and the memory, processors, and media hardware of the electronic device. As previously mentioned, embodiments of a hybrid operating system are not limited to Linux-based systems and environments. Additionally, alternate implementations can include more than two runtime environments of a hybrid operating system.

[0020] The first runtime environment 330 includes a media control service 340, and the second runtime environment 332 includes a data routing service 342. The media control service and the data routing service can each be implemented as computer-executable instructions, such as software applications, and

executed by one or more processors to implement the various embodiments described herein. In embodiments, the media control service is implemented as an Android Audio Flinger service (*e.g.*, in an Android runtime environment) for audio management and distribution of audio data files to the media hardware. Additionally, the data routing service is implemented as an Advanced Linux Sound Architecture (ALSA) service (*e.g.*, in a GNU/Linux runtime environment).

[0021] The runtime environments also include software applications, such as Android applications 350 in the first runtime environment 330 and Ubuntu applications 352 in the second runtime environment 332. The software applications in the respective runtime environments may include any one or combination of the software applications described with reference to FIG. 1. The media data that is retrieved and/or generated by the applications can be stored or otherwise maintained in the shared memory 314 as the media data files 316.

[0022] The media control service 340 in the first runtime environment 330 manages the distribution of media data files from the Android applications 350 to the media hardware 310. Media data from the media data files 316 is routed for input to the media hardware, and the media control service communicates hardware control signals through the operating system kernel 322 to initiate and control the media hardware that renders the media data. The data routing service 342 in the second runtime environment 332 is implemented to route additional media data files from the Ubuntu applications 352 to the media control service 340 in the first runtime environment 330 where the media control service also manages the distribution of the additional media data files to the media hardware 310.

[0023] The data routing service 342 in the second runtime environment 332 utilizes the shared memory 314 to store the additional media data files from the Ubuntu applications 352, and the additional media data files are then accessible to the operating system (*e.g.*, the operating system kernel 322) from the shared memory. The data routing service 342 is also implemented to route hardware control data 390 that is associated with the additional media data files from the second runtime environment to the media control service 340 in the first runtime environment. The hardware control data is routed via the kernel binder 324 between the second runtime environment and the first runtime environment. The media control service 340 can then initiate (or ignore) hardware control signals from the hardware control data 390 to manage the distribution of the media data files 316 that are routed from the second runtime environment by the data routing service. The media control service 340 checks the distribution policies 318 to determine whether to ignore or pass hardware control data 390 from the second runtime environment to the media hardware 310.

[0024] As described with reference to FIG. 2, the media control service 340 may be implemented as an Android Audio Flinger service (*e.g.*, in an Android runtime environment), and includes code libraries that may also be used to implement the data routing service 342, which may be implemented as an Advanced Linux Sound Architecture (ALSA) service (*e.g.*, in a GNU/Linux runtime environment). The Android Audio Flinger libraries are ported to the Linux/Ubuntu environment, and a new ALSA PCM plug-in (pulse code modulation for audio data) and a new ALSA CTL plug-in (media hardware control) can be built from the Android code libraries.

[0025] Example method 400 is described with reference to FIG. 4 in accordance with one or more embodiments of hybrid operating system media integration. Generally, any of the services, functions, methods, procedures, components, and modules described herein can be implemented using software, firmware, hardware (e.g., fixed logic circuitry), manual processing, or any combination thereof. A software implementation represents program code that performs specified tasks when executed by a computer processor. The example methods may be described in the general context of computer-executable instructions, which can include software, applications, routines, programs, objects, components, data structures, procedures, modules, functions, and the like. The program code can be stored in one or more computer-readable storage media devices, both local and/or remote to a computer processor. The methods may also be practiced in a distributed computing environment by multiple computer devices. Further, the features described herein are platform-independent and can be implemented on a variety of computing platforms having a variety of processors.

[0026] FIG. 4 illustrates example method(s) 400 of hybrid operating system media integration. The order in which the method blocks are described are not intended to be construed as a limitation, and any number or combination of the described method blocks can be combined in any order to implement a method, or an alternate method.

[0027] At block 402, a first media data file and first hardware control data for media hardware is received at a first runtime environment. For example, the first runtime environment 130 (FIG. 1) receives a media data file and hardware control data from a software application 150 in the first runtime environment. In a specific

example, an Android call application 350 (FIG. 3) of the first runtime environment 330 receives an incoming call signal that points to a ringer alert audio file and a specific volume control level.

[0028] At block 404, a second media data file and second hardware control data for the media hardware is received at a second runtime environment. For example, the second runtime environment 132 receives a media data file and hardware control data from a software application 152 in the second runtime environment. In a specific example, an Ubuntu Internet browser application 352 of the second runtime environment 332 receives streaming movie video and audio data with user-selectable audio control settings.

[0029] At block 406, the second media data file is routed from the second runtime environment to the first runtime environment. For example, the data routing service 142 in the second runtime environment 132 routes a media data file 116 from the second runtime environment to the first runtime environment 130 via the shared memory 114, which is accessible by the operating system kernel 120. In the specific example, the data routing service 342 in the second runtime environment 332 routes the video and audio data files based on the Ubuntu Internet browser application 352 instructions.

[0030] At block 408, the second hardware control data is routed from the second runtime environment to the first runtime environment. For example, the data routing service 142 in the second runtime environment 132 routes the hardware control data 196 from the second runtime environment to the first runtime environment 130 via the kernel binder 122 between the second runtime environment and the first runtime environment. In the continuing example, the data routing

service 342 sends speaker and volume setting control signals through the kernel binder 324 from the second runtime environment 332 to the first runtime environment 330.

[0031] At block 410, the media hardware is managed using the first hardware control data and/or the second hardware control data in compliance with media distribution policies. For example, the media control service 140 in the first runtime environment 130 manages the media hardware 110 using the hardware control signals 192 (*e.g.*, derived from the first hardware control data and/or the second hardware control data). In implementations, the media control service 140 manages the media hardware in compliance with the media distribution policies 118 by selecting the first hardware control data and suppressing the second hardware control data, by selecting the second hardware control data and suppressing the first hardware control data, or by combining the first hardware control data and the second hardware control data. In the continuing example described with reference to the electronic device 300 shown in FIG. 3, the distribution policies 318 prioritize incoming call alerts over all other audio data. Then, the media control service 340 would select the first hardware control data from the Android call application 350 for the incoming call to turn on a particular audio speaker at a particular volume; suppress the second hardware control data from the Ubuntu Internet browser application 352 for the streaming movie audio; and couple the ringer alert audio data file to the media hardware 310.

[0032] The media distribution policies 118 can be quite complex and may prioritize audio data and loudspeaker hardware control data from particular software applications or specific runtime environments over other audio data and loudspeaker

hardware control data. The prioritization of audio information may be completely different from the prioritization of video information. For example, the video data and display hardware control data may continue to be sent from the Ubuntu Internet browser application 352 in the second runtime environment 332 while the audio data and loudspeaker hardware control data from the second runtime environment 332 is suppressed.

[0033] FIG. 5 illustrates various components of an example electronic device 500 that can be implemented as any device described with reference to any of the previous FIGs. 1-4. The electronic device may be implemented as any one or combination of a fixed or mobile device, in any form of a consumer, computer, portable, user, communication, phone, navigation, gaming, media playback, and/or electronic device.

[0034] The electronic device 500 includes communication transceivers 502 that enable wired and/or wireless communication of device data 504, such as received data, data that is being received, data scheduled for broadcast, data packets of the data, etc. Example transceivers include wireless personal area network (WPAN) radios compliant with various IEEE 802.15 (sometimes referred to as BluetoothTM) standards, wireless local area network (WLAN) radios compliant with any of the various IEEE 802.11 (sometimes referred to as WiFiTM) standards, wireless wide area network (WWAN) radios for cellular telephony, wireless metropolitan area network (WMAN) radios compliant with various IEEE 802.15 (sometimes referred to as WiMAXTM) standards, and wired local area network (LAN) Ethernet transceivers.

[0035] The electronic device 500 may also include one or more data input ports 506 via which any type of user data, media content, and/or inputs can be received, such as user-selectable inputs, messages, music, television content, recorded video content, and any other type of audio, video, and/or image data received from any content and/or data source. The data input ports may include USB ports, coaxial cable ports, and other serial or parallel connectors (including internal connectors) for flash memory, DVDs, CDs, and the like. These data input ports may be used to couple the electronic device 500 to components, peripherals, or accessories such as microphones or cameras.

[0036] The electronic device 500 includes one or more processors 508 (*e.g.*, any of microprocessors, controllers, and the like), which process computer-executable instructions to control operation of the device. Alternatively or in addition, the electronic device can be implemented with any one or combination of software, hardware, firmware, or fixed logic circuitry that is implemented in connection with processing and control circuits, which are generally identified at 510. Although not shown, the electronic device can include a system bus or data transfer system that couples the various components within the device. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures.

[0037] The electronic device 500 also includes one or more memory devices 512 that enable data storage, examples of which include random access memory (RAM), non-volatile memory (*e.g.*, read-only memory (ROM), flash memory, EPROM, EEPROM, etc.), and a disk storage device. A disk storage device

may be implemented as any type of magnetic or optical storage device, such as a hard disk drive, a recordable and/or rewriteable disc, any type of a digital versatile disc (DVD), and the like. The electronic device may also include a mass storage media device.

[0038] A memory device 512 provides data storage mechanisms to store the device data 504, other types of information and/or data, and various device applications 514 (*e.g.*, software applications). For example, an operating system 516 can be maintained as software instructions within a memory device and executed on processors 508. The device applications may also include a device manager, such as any form of a control application, software application, signal-processing and control module, code that is native to a particular device, a hardware abstraction layer for a particular device, and so on. In embodiments of hybrid operating system media integration, the operating system 516 can establish one or more runtime environments 518 as described herein. For example, a first runtime environment includes a media control service 520, and a second runtime environment includes a data routing service 522.

[0039] The electronic device 500 also includes an audio and/or video processing system 524 that generates audio data for an audio system 526 and/or generates display data for a display system 528. The audio system and/or the display system may include any devices that process, display, and/or otherwise render audio, video, display, and/or image data. Display data and audio signals can be communicated to an audio component and/or to a display component via an RF (radio frequency) link, S-video link, HDMI (high-definition multimedia interface), composite video link, component video link, DVI (digital video interface), analog

audio connection, or other similar communication link, such as media data port 530. In implementations, the audio system and/or the display system are external components to the electronic device. Alternatively, the audio system and/or the display system are integrated components of the example electronic device, such as an integrated touch-screen.

[0040] Although embodiments of hybrid operating system media integration have been described in language specific to features and/or methods, the subject of the appended claims is not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as example implementations of hybrid operating system media integration.

CLAIMS

1. A media system, comprising:
 - media hardware configured to render media data;
 - a media control service in a first runtime environment, the media control service configured to manage distribution of a media data file from the first runtime environment to the media hardware; and
 - a data routing service in a second runtime environment, the data routing service configured to route an additional media data file from the second runtime environment to the media control service in the first runtime environment, the media control service further configured to manage distribution of the additional media data file to the media hardware.

2. The media system as recited in claim 1, further comprising:
 - an operating system configured for establishing the first runtime environment and the second runtime environment.

3. The media system as recited in claim 2, further comprising:
 - a shared memory configured to store the media data file from the first runtime environment and the additional media data file from the second runtime environment, where the media data file and the additional media data file are accessible to the operating system from the shared memory.

4. The media system as recited in claim 1, wherein the data routing service is further configured to route hardware control data that is associated with the additional media data file from the second runtime environment to the media control service in the first runtime environment.

5. The media system as recited in claim 4, further comprising:
a kernel binder configured for transferring the hardware control data from the second runtime environment to the first runtime environment.

6. The media system as recited in claim 1, wherein the media hardware includes an audio speaker.

7. The media system as recited in claim 1, wherein the media hardware includes at least one of a microphone, a display component, or a camera.

8. The media system as recited in claim 1, wherein the media control service complies with media distribution policies to manage rendering of the media data file and the additional media data file.

9. The media system as recited in claim 1, wherein the media control service includes code libraries, and wherein the data routing service includes one or more of the code libraries of the media control service.

10. An electronic device, comprising:

a first runtime environment configured for execution with an operating system kernel;

a media control service in the first runtime environment, the media control service configured to manage distribution of a media data file with at least audio data or video data from the first runtime environment;

a second runtime environment configured for execution with the operating system kernel; and

a data routing service in the second runtime environment, the data routing service configured to route an additional media data file to the media control service in the first runtime environment.

11. The electronic device as recited in claim 10, further comprising:

a shared memory configured to store the additional media data file from the second runtime environment, and wherein the additional media data file is routed to the media control service via the shared memory.

12. The electronic device as recited in claim 10, wherein the data routing service is further configured to route hardware control data that is associated with the additional media data file from the second runtime environment to the media control service in the first runtime environment.

13. The electronic device as recited in claim 12, wherein the hardware control data is routed from the second runtime environment to the first runtime environment via a kernel binder.

14. The electronic device as recited in claim 12, wherein the media control service is further configured to comply with media distribution policies to suppress the hardware control data that is associated with the additional media data file.

15. The electronic device as recited in claim 10, further comprising:
an operating system configured for establishing the first runtime environment and the second runtime environment.

16. The electronic device as recited in claim 10, wherein the media control service includes code libraries, and wherein the data routing service includes one or more of the code libraries of the media control service.

17. A method, comprising:

receiving a first media data file and first hardware control data for media hardware at a first runtime environment;

receiving a second media data file and second hardware control data for the media hardware at a second runtime environment;

routing the second media data file from the second runtime environment to the first runtime environment;

routing the second hardware control data from the second runtime environment to the first runtime environment; and

managing the media hardware at the first runtime environment using the first hardware control data or the second hardware control data.

18. The method as recited in claim 17, wherein managing the media hardware includes complying with media distribution policies to perform at least one of:

selecting the first hardware control data and suppressing the second hardware control data;

selecting the second hardware control data and suppressing the first hardware control data; or

combining the first hardware control data and the second hardware control data.

19. The method as recited in claim 17, wherein the second media data file is routed via a shared memory from the second runtime environment to the first runtime environment.

20. The method as recited in claim 17, wherein the second hardware control data is routed via a kernel binder from the second runtime environment to the first runtime environment.

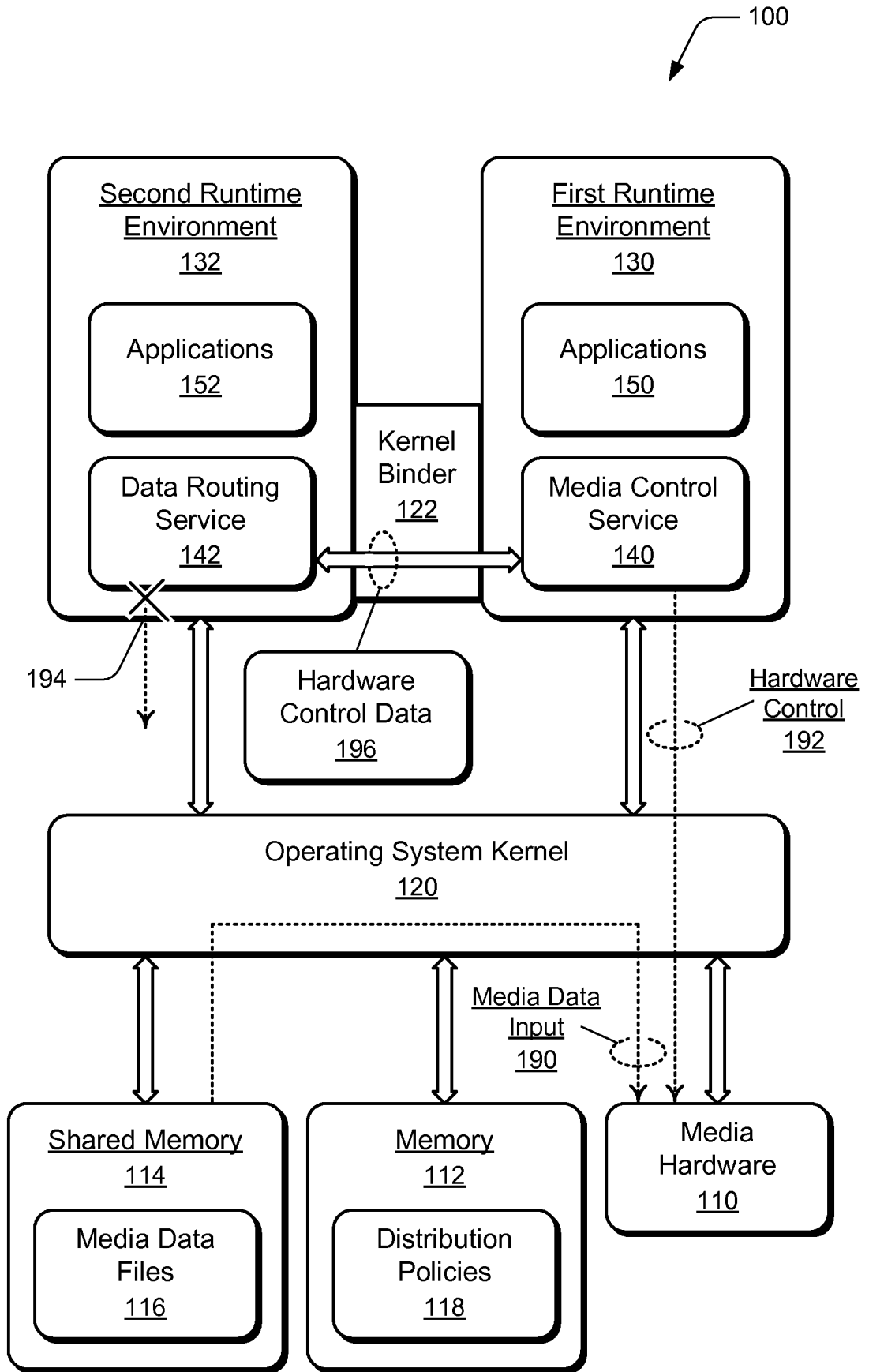


FIG. 1

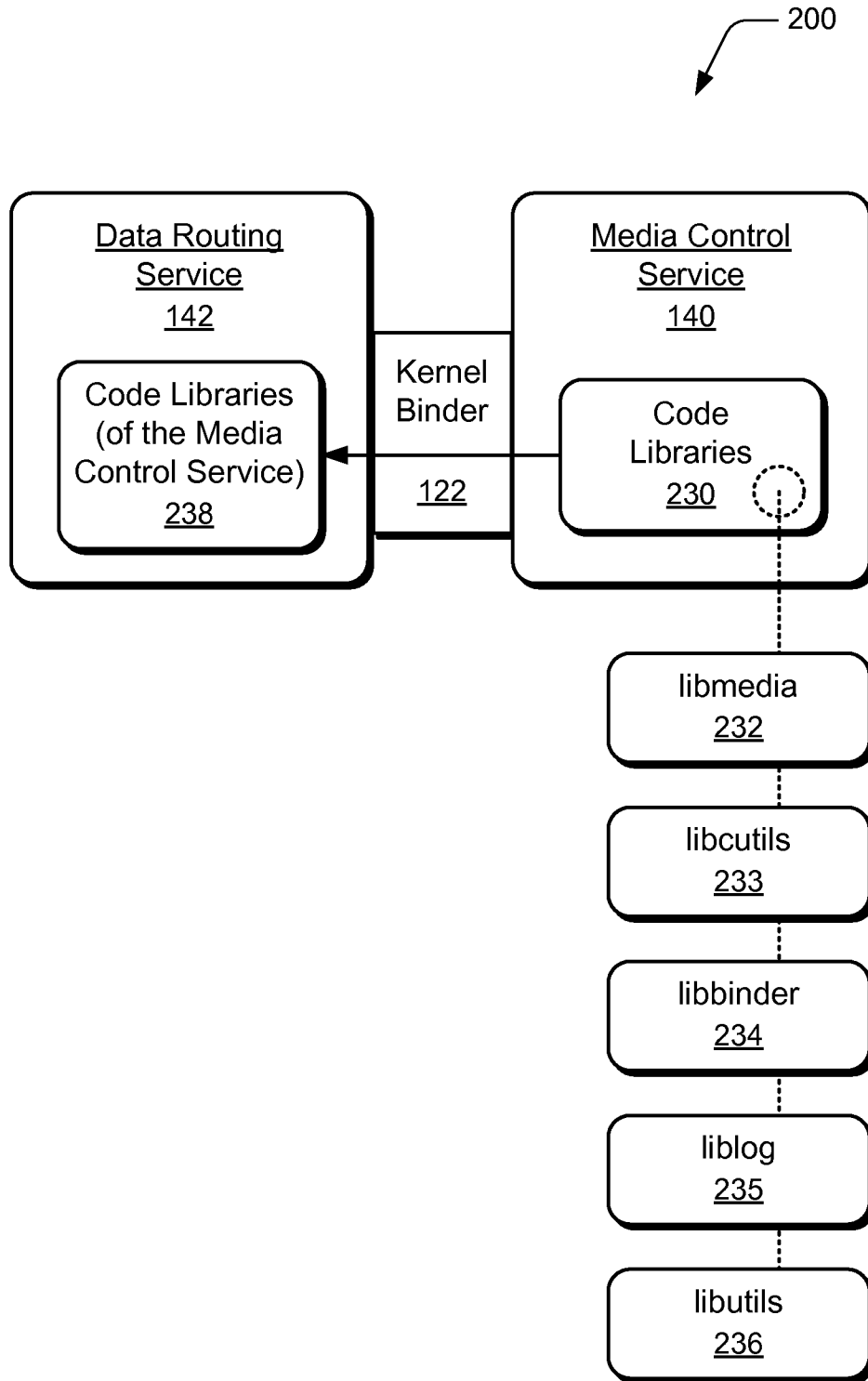


FIG. 2

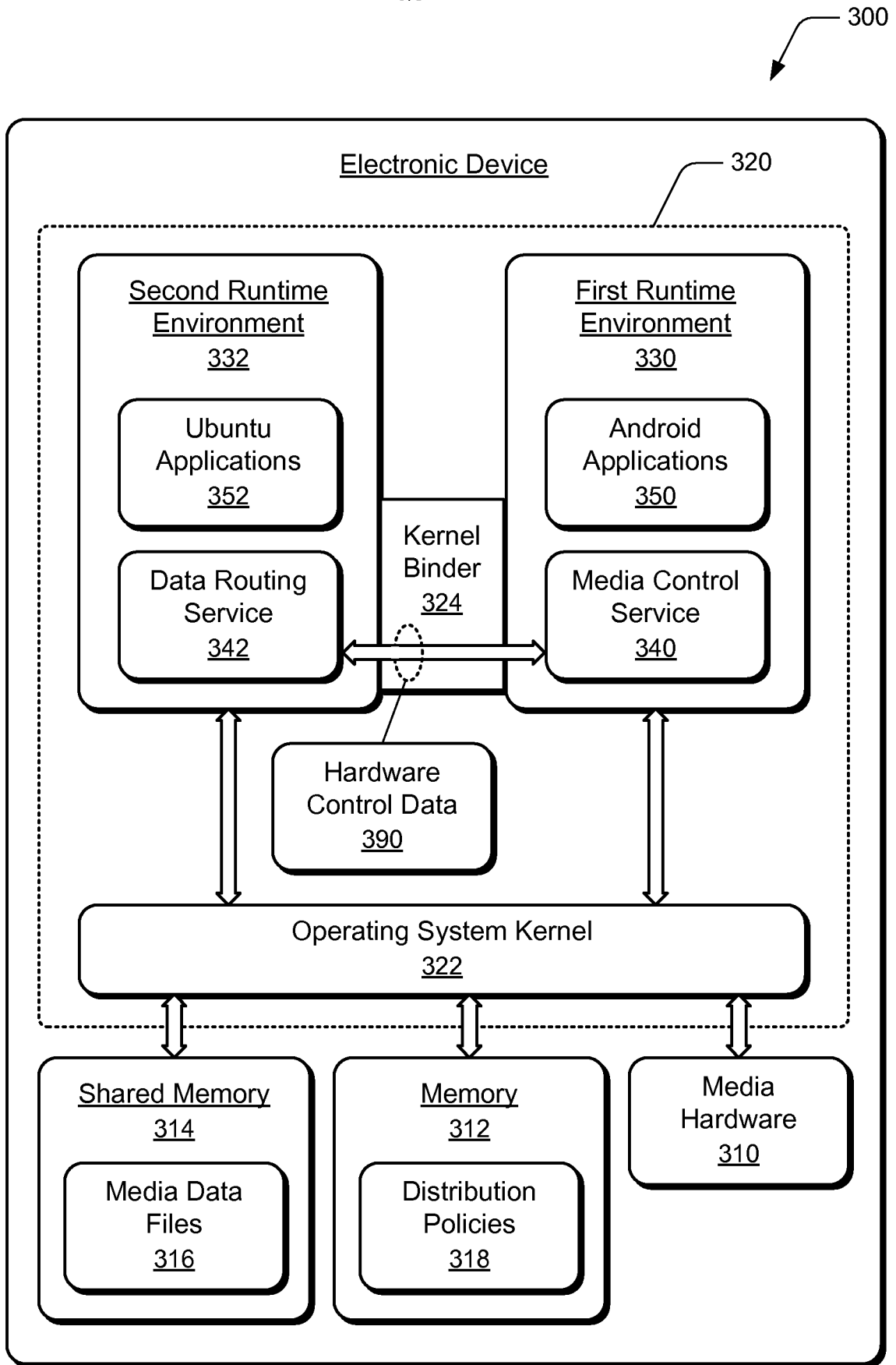
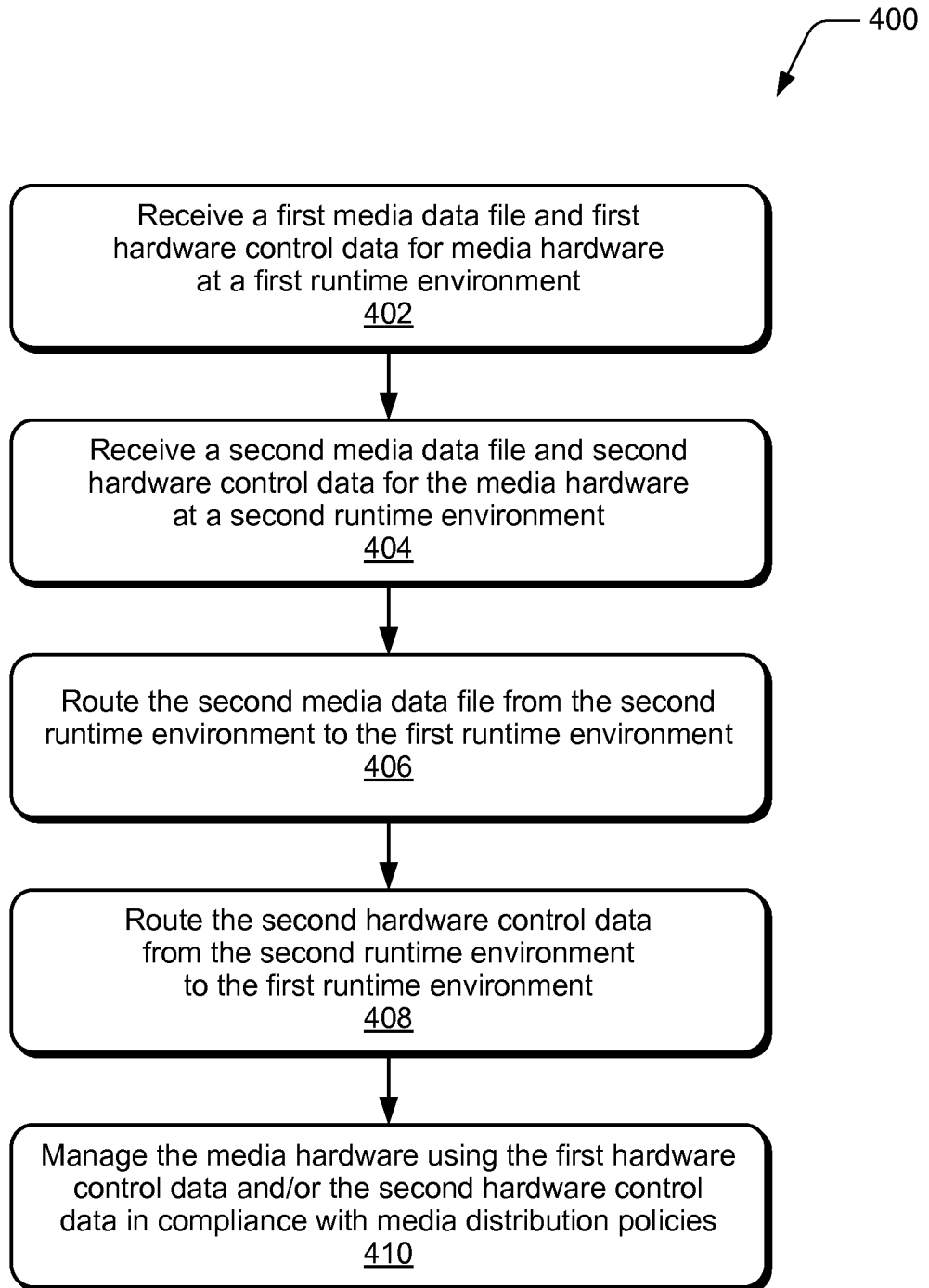


FIG. 3

**FIG. 4**

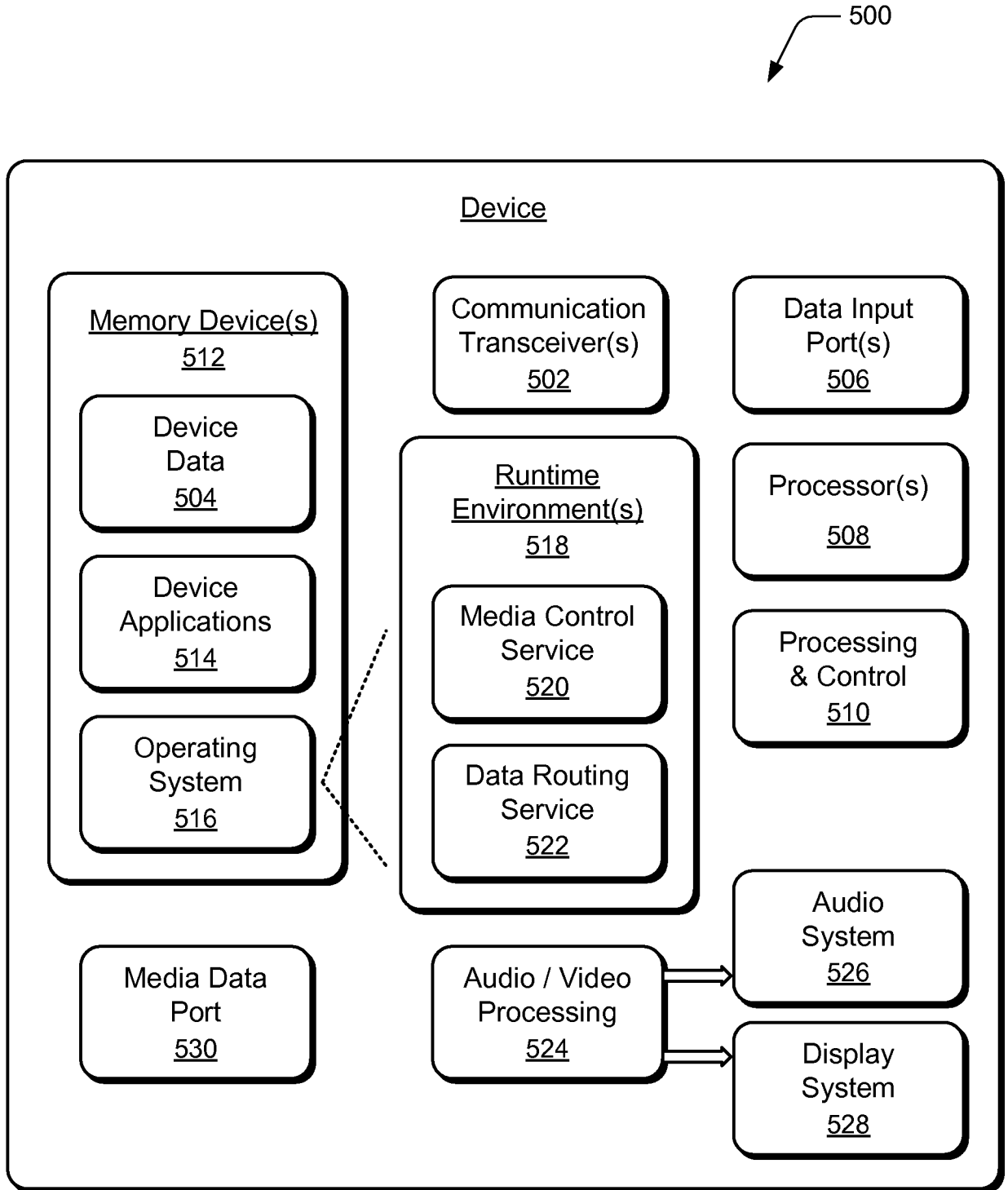


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2011/070018

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/445 (2006.01) i

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: G06F 9/-, G06F 11/-, G06F 12/-, G06F 17/-, G06F 21/-, H04L 9/-, 29/-

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)

CNPAT, WPI, EPODOC: runtime environment, os, operating system, media, multi, audio, video, shar???, hardware?, resource?
Conflict???,

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US2008077993A1 (INTEL CORP) 27 Mar. 2008 (27.03.2008) the whole document	1-20
A	US7275121B1 (NVIDIA CORP) 25 Sep. 2007 (25.09.2007) the whole document	1-20
A	JP2009104258A (INTEL CORP) 14 May 2009 (14.05.2009) the whole document	1-20
A	US2007057953A1 (MICROSOFT CORP) 15 Mar. 2007 (15.03.2007) The whole document	1-20

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search
02 Sep. 2011 (02.09.2011)Date of mailing of the international search report
20 Oct. 2011 (20.10.2011)Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088
Facsimile No. 86-10-62019451Authorized officer
MA, Yafan
Telephone No. (86-10)62411656

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2011/070018

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US2008077993A1	27.03.2008	EP1906333A3	10.11.2010
		CN101154256A	02.04.2008
		EP1906333A2	02.04.2008
		KR989977B1	26.10.2010
		KR20080028340A	31.03.2008
US7275121B1	25.09.2007	None	
JP2009-104258A	14.05.2009	None	
US2007057953A1	15.03.2007	US7463268B2	09.12.2008