

US009270530B1

(12) **United States Patent**
Ford

(10) **Patent No.:** **US 9,270,530 B1**
(45) **Date of Patent:** **Feb. 23, 2016**

(54) **MANAGING IMAGING OF MULTIPLE COMPUTING DEVICES**

(75) Inventor: **Daniel M. Ford**, Seattle, WA (US)

(73) Assignee: **Amazon Technologies, Inc.**, Seattle, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1086 days.

(21) Appl. No.: **13/118,157**

(22) Filed: **May 27, 2011**

(51) **Int. Cl.**
G06F 15/173 (2006.01)
H04L 12/24 (2006.01)
H04L 29/08 (2006.01)

(52) **U.S. Cl.**
CPC **H04L 41/0893** (2013.01); **H04L 67/303** (2013.01)

(58) **Field of Classification Search**
CPC . H04L 41/0893; H04L 41/5003; H04L 41/50; H04L 41/5025; H04L 41/08; H04L 41/0803; H04L 41/0806; H04L 41/0672; H04L 41/12; H04L 41/28; H04L 41/5041; H04L 67/306; H04L 67/34; H04L 67/125; H04L 67/30303; H04L 63/10; H04L 47/2408; H04L 41/0886; H04L 41/5045
USPC 709/224, 223, 222, 225, 226, 203, 204, 709/213, 217, 220, 221, 228; 370/252, 254, 370/400; 715/736, 734, 735; 717/177, 101, 717/176, 178
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,421,777 B1 * 7/2002 Pierre-Louis et al. 713/2
6,567,774 B1 * 5/2003 Lee et al. 703/23
7,082,527 B2 * 7/2006 Zimmer et al. 713/2

7,103,762 B2 * 9/2006 Harper et al. 713/1
7,487,343 B1 * 2/2009 Insley et al. 713/1
7,664,834 B2 * 2/2010 Keith, Jr. 709/220
7,945,751 B2 * 5/2011 Garapati et al. 711/162
8,112,505 B1 * 2/2012 Ben-Shaul et al. 709/219
8,244,836 B2 * 8/2012 DeHaan 709/220
8,302,091 B2 * 10/2012 Aridor et al. 717/174
8,326,972 B2 * 12/2012 DeHaan et al. 709/224
8,332,689 B2 * 12/2012 Timashev et al. 714/15
8,386,757 B1 * 2/2013 Midgley et al. 713/1
8,639,917 B1 * 1/2014 Ben-Shaul et al. 713/2
2001/0047406 A1 * 11/2001 Araujo et al. 709/223
2003/0051128 A1 * 3/2003 Rodriguez et al. 713/100
2003/0126242 A1 * 7/2003 Chang 709/222
2005/0273588 A1 * 12/2005 Ong et al. 713/2
2006/0047946 A1 * 3/2006 Keith, Jr. 713/2
2008/0027950 A1 * 1/2008 Fukumi 707/10
2009/0049160 A1 * 2/2009 Cherian et al. 709/222
2009/0193245 A1 * 7/2009 Isaacson 713/2
2010/0125770 A1 * 5/2010 Keith, Jr. 714/752
2010/0223610 A1 * 9/2010 DeHaan et al. 717/178
2010/0306337 A1 * 12/2010 DeHaan 709/213

* cited by examiner

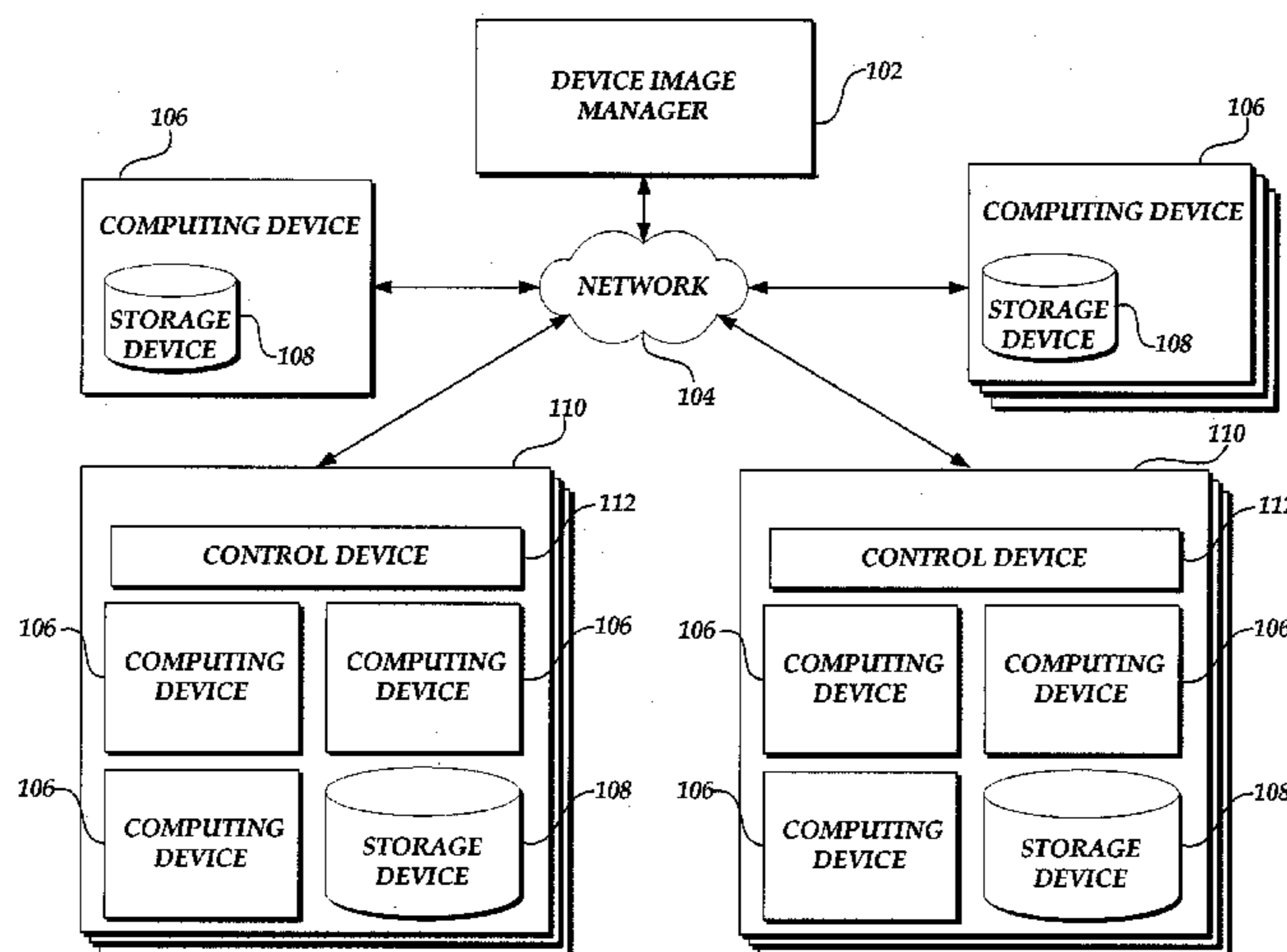
Primary Examiner — Saket K Daftuar

(74) *Attorney, Agent, or Firm* — Knobbe, Martens Olson & Bear, LLP

(57) **ABSTRACT**

Systems and methods for managing imaging of multiple computing devices are provided. A device image manager determines a number of device images which may be loaded by the computing devices. The device image manager transmits a set of device images to the computing devices, which then store the device images. When provisioning of one or more of the computing devices is desired, the device image manager selects a device image or receives selection of a device image. The device image manager then sends a provisioning command device to the appropriate computing devices identifying the selected device image. If the computing devices have stored the selected device image, the computing devices may load the selected device image without requiring retransmission of the device image.

27 Claims, 9 Drawing Sheets



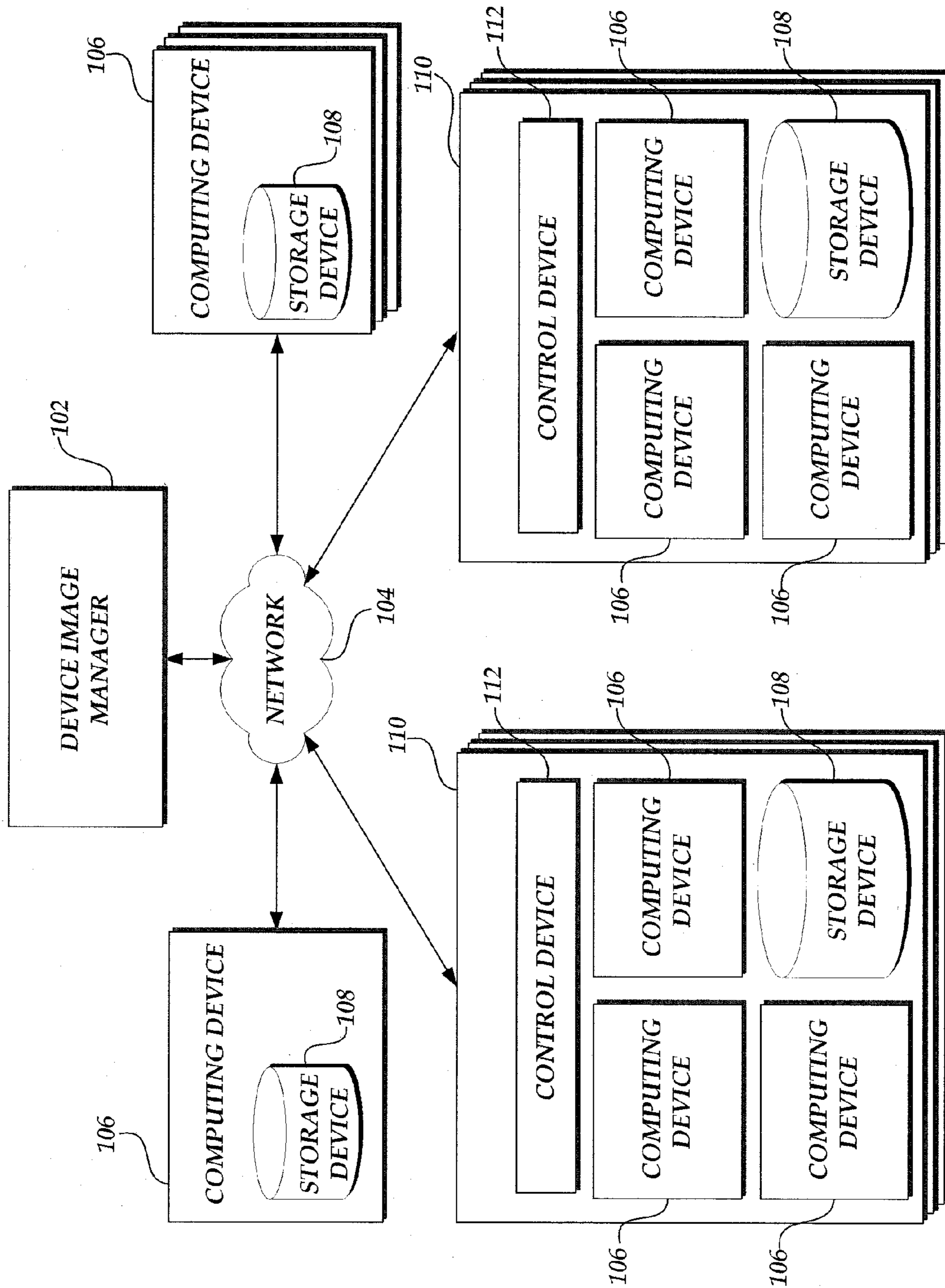


Fig. 1.

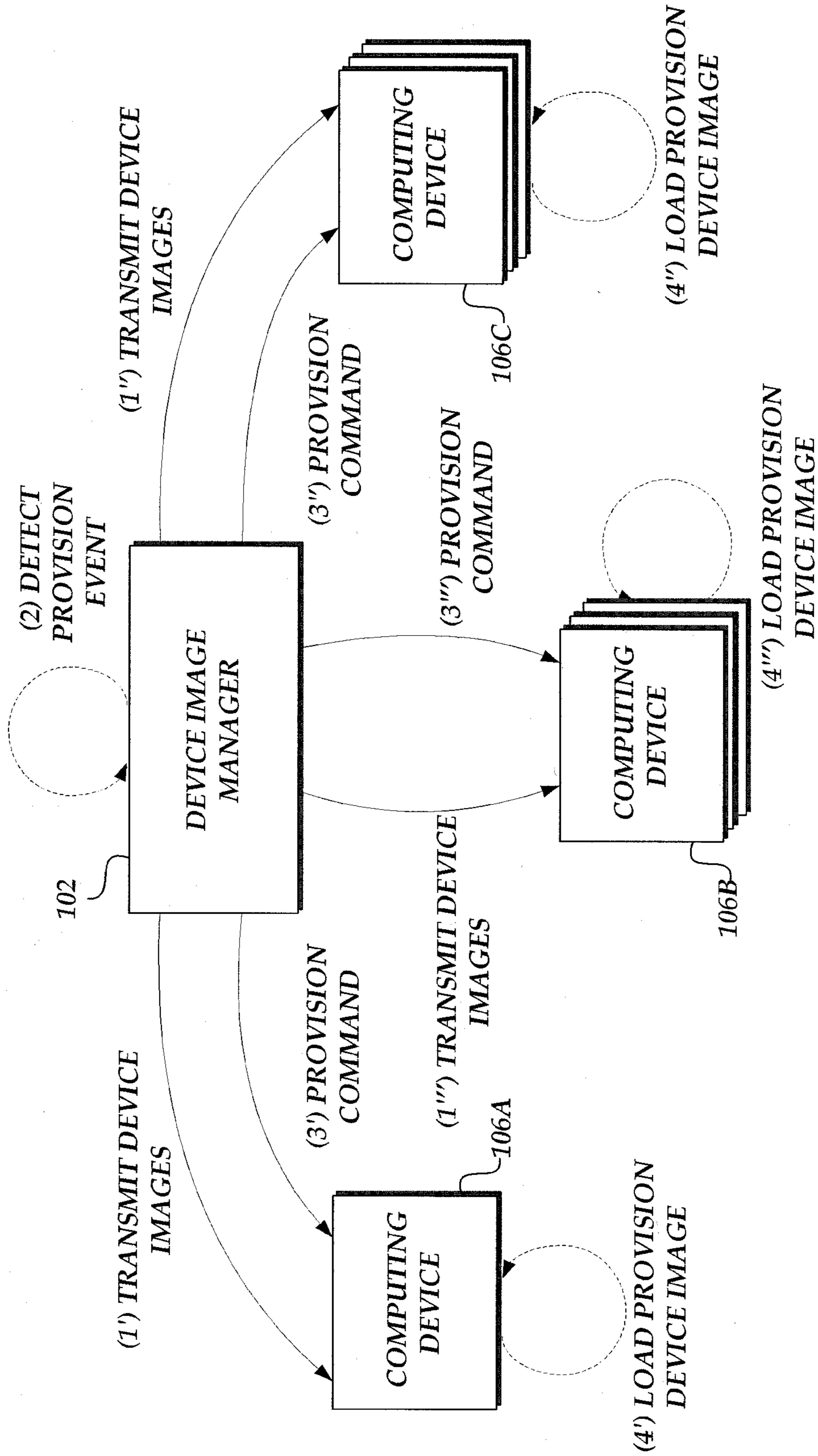


Fig. 2A.

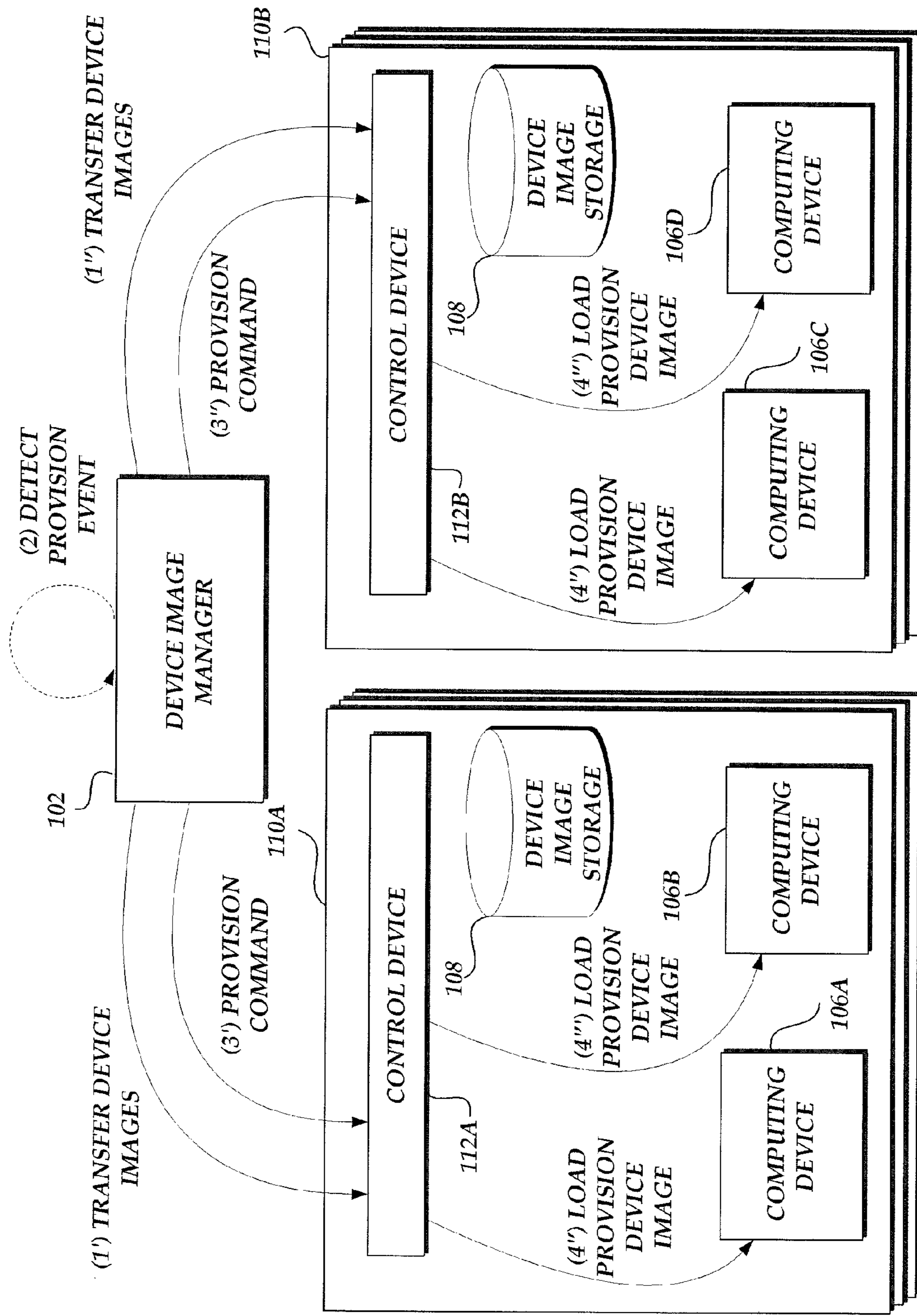


Fig. 2B.

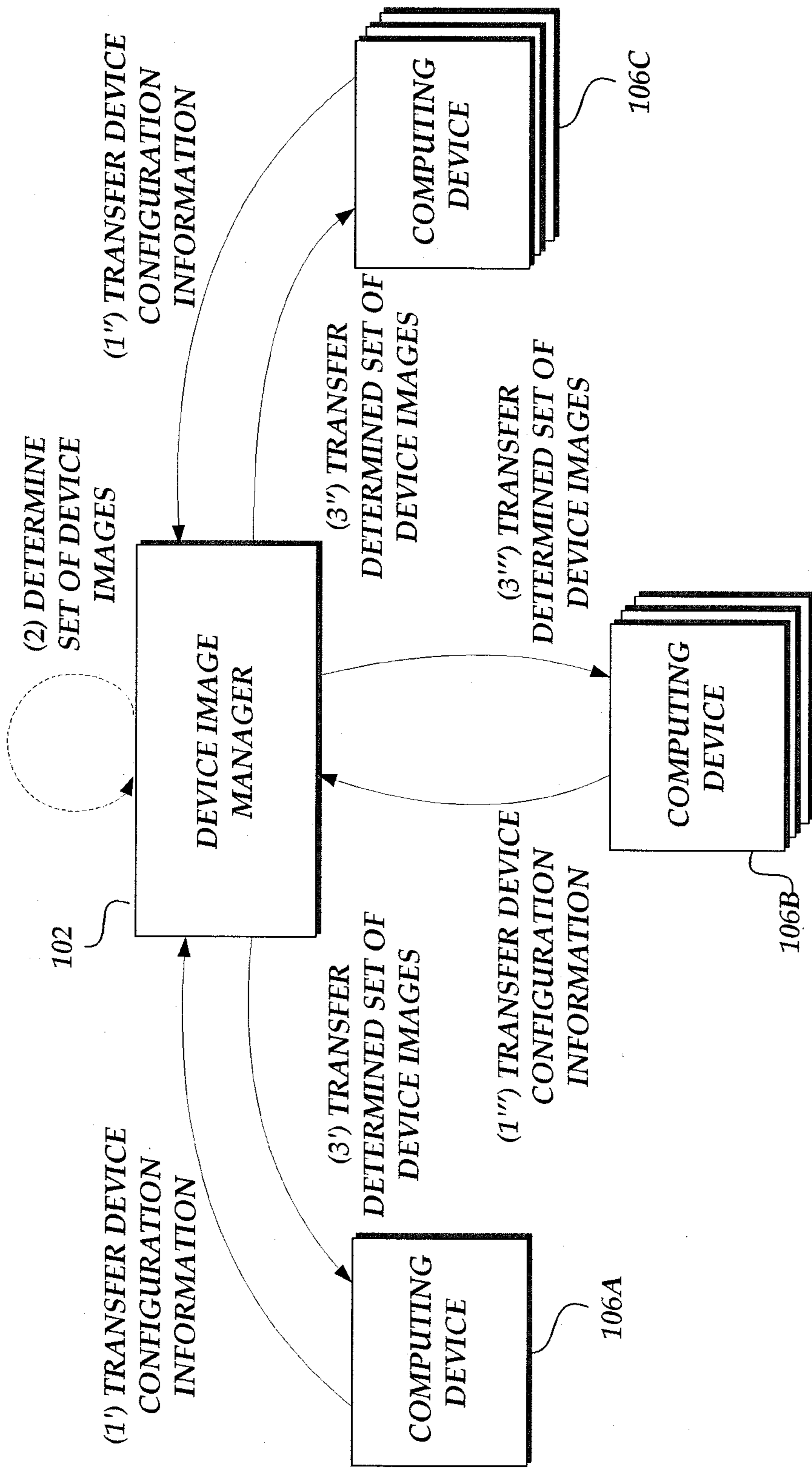


FIG.3A.

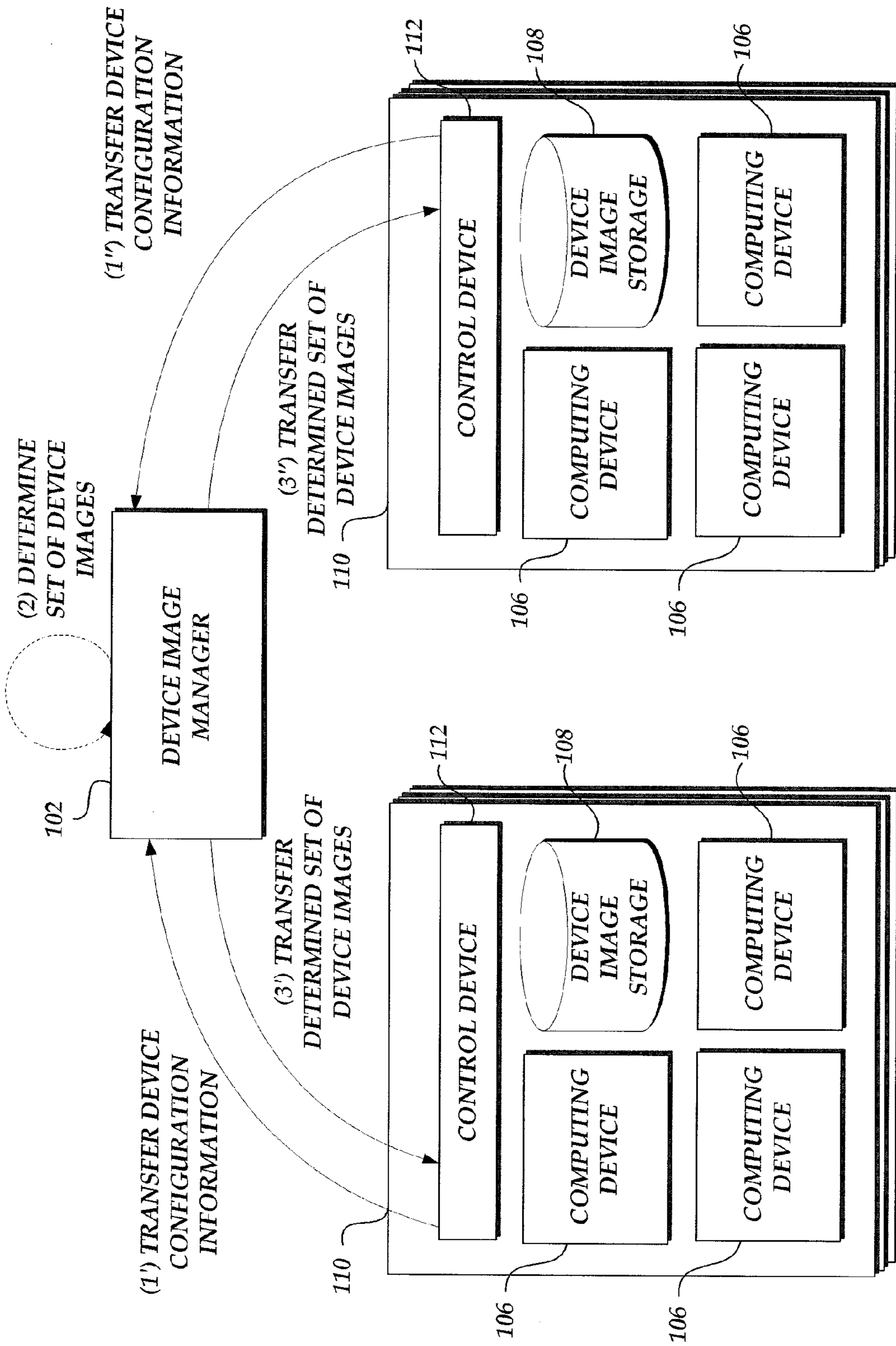


FIG.3B.

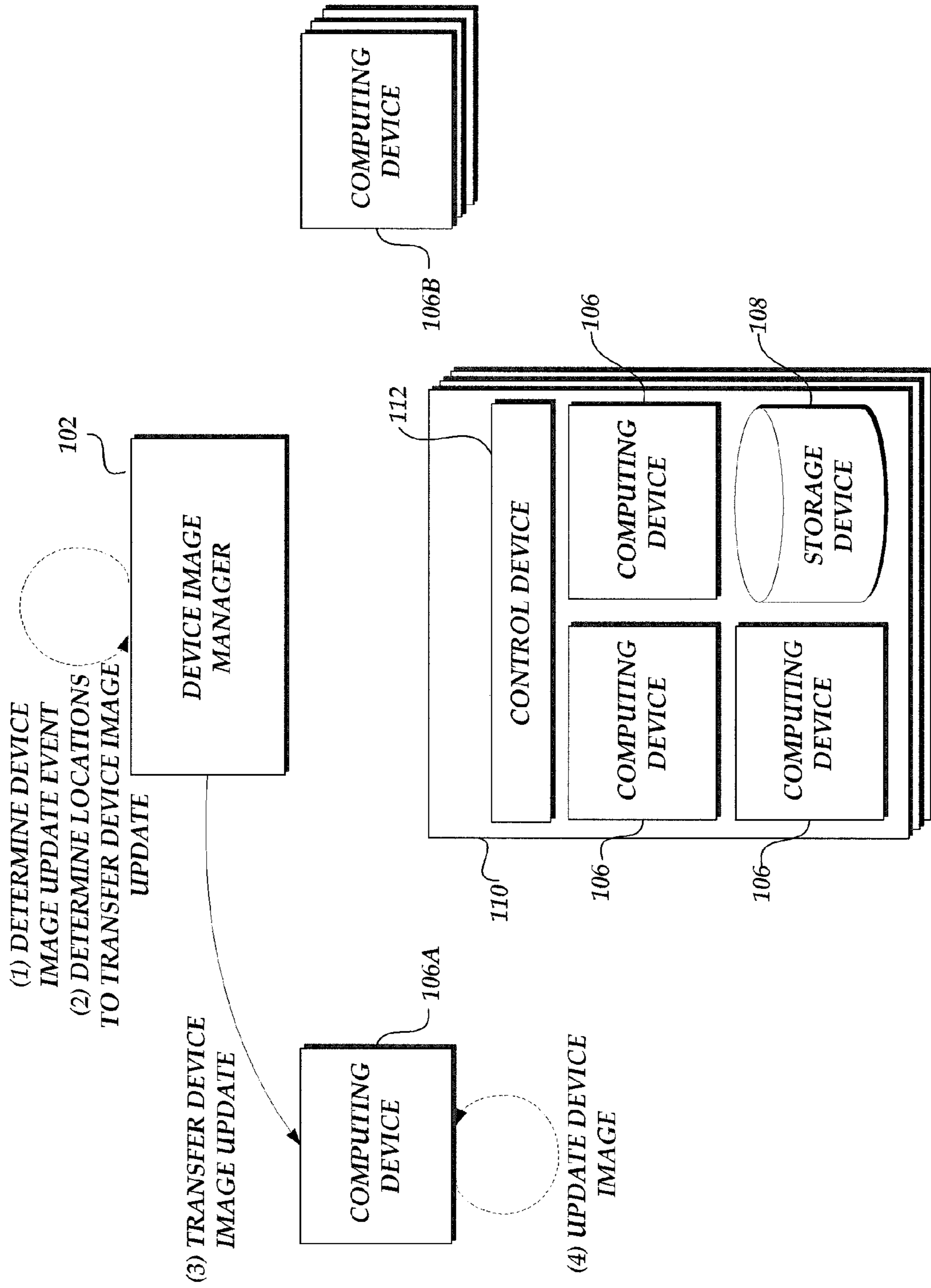


Fig. 4.

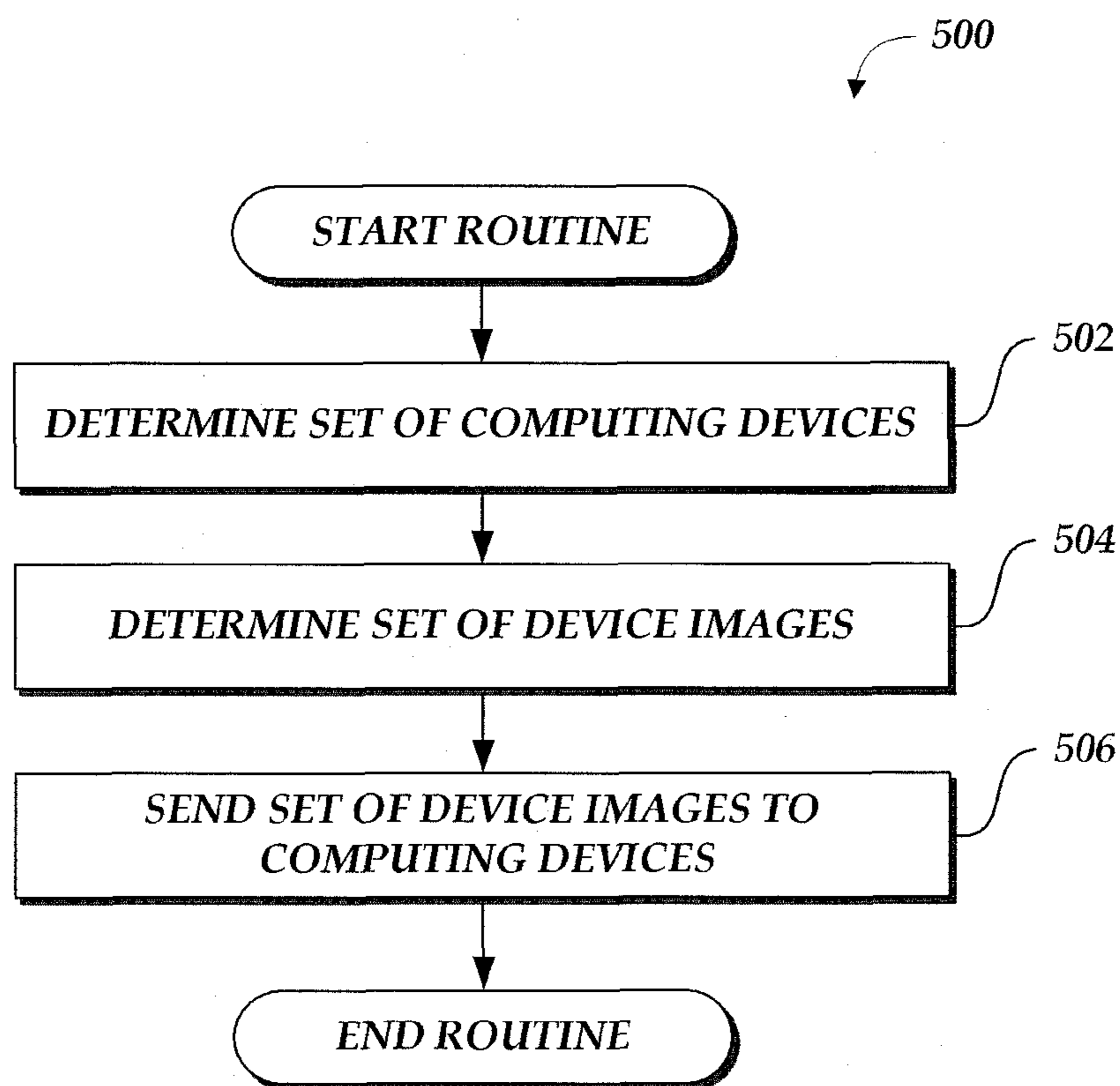


Fig. 5A.

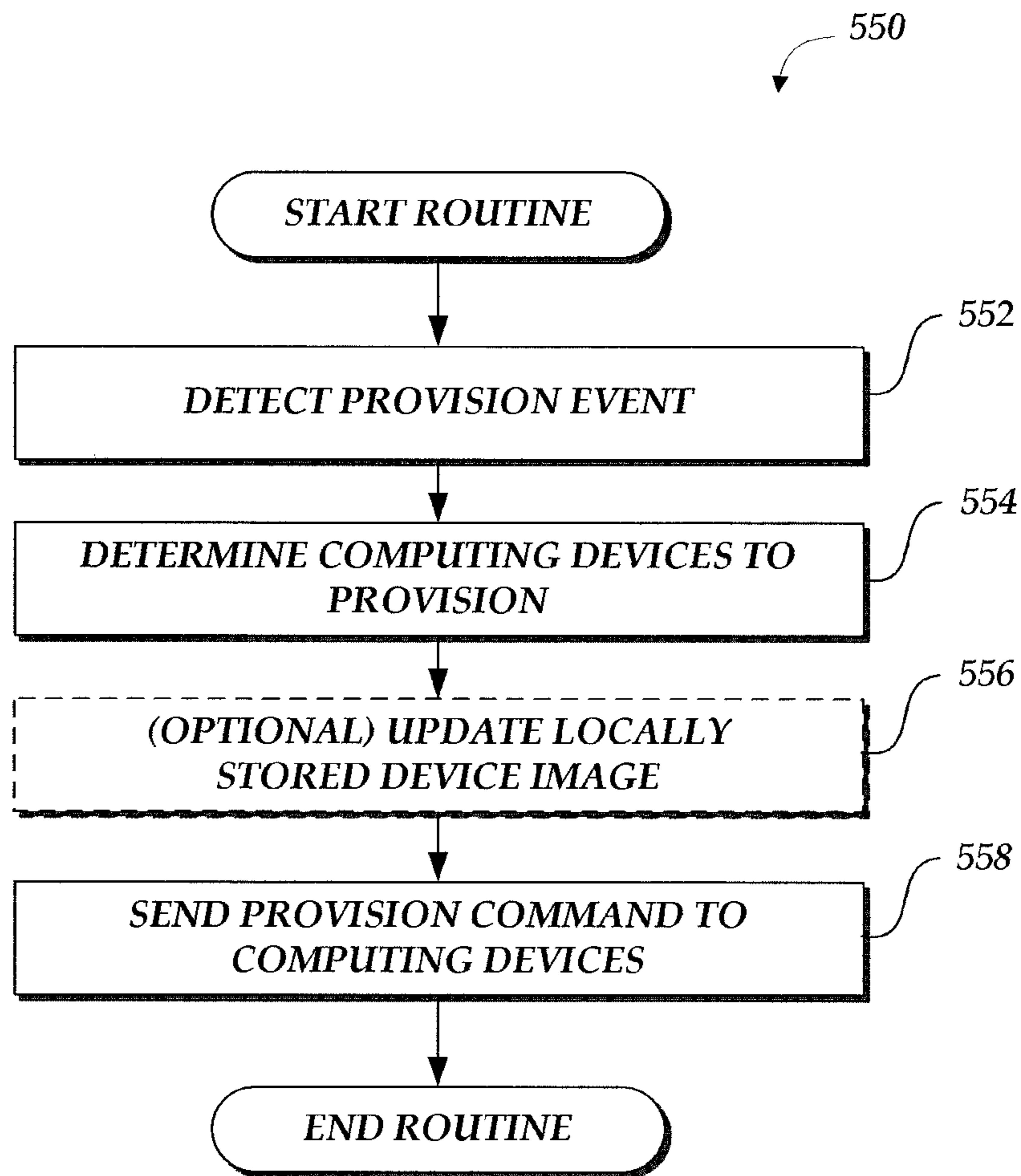


Fig. 5B.

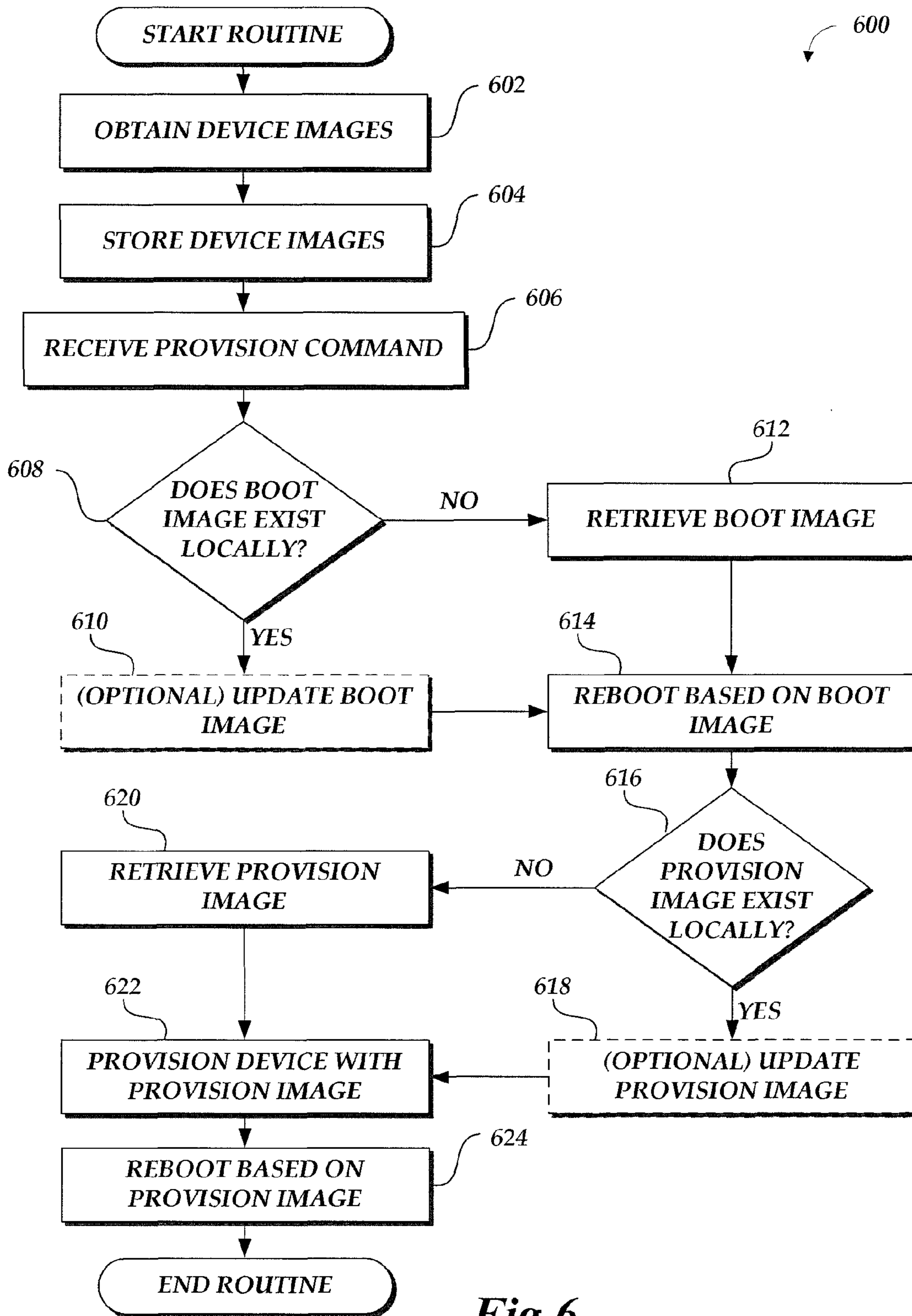


Fig. 6.

MANAGING IMAGING OF MULTIPLE COMPUTING DEVICES

BACKGROUND

Generally described, computing devices utilize a communication network, or a series of communication networks, to exchange data. Companies and organizations operate computer networks that interconnect a number of computing devices to support operations or provide services to third parties. The computing systems can be located in a single geographic location or located in multiple, distinct geographic locations (e.g., interconnected via private or public communication networks). Specifically, data centers or data processing centers, herein generally referred to as a “data center,” may include a number of interconnected computing systems to provide computing resources to users of the data center. The data centers may be private data centers operated on behalf of an organization or public data centers operated on behalf, or for the benefit of the general public.

To facilitate increased utilization of data center resources, individual computing devices within a data center may be configured to provide specific functionality according to the requirements of the data center. For example, various computing devices may be associated with different operating systems or operating system configurations to enable a computing device to provide different desired functionalities, or to provide similar functionalities more efficiently. These operating systems or operating system configurations are often contained within a device image, which a computing device may process in order implement the desired software configuration. This implementation of a desired software configuration is generally referred to as provisioning.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this disclosure will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIG. 1 is a block diagram depicting an illustrative environment for managing provisioning of multiple computing devices, the environment including a number of computing devices, a number of collections of computing devices with a control device, and a device image manager;

FIGS. 2A and 2B are block diagrams of the provisioning management environment of FIG. 1 illustrating the transfer of devices images, detection of provision event, and loading of device images on multiple computing devices;

FIGS. 3A and 3B are block diagrams of the provisioning management environment of FIG. 1 illustrating the determination of device images based on information relating to multiple computing devices;

FIG. 4 is a block diagram of the provisioning management environment of FIG. 1 illustrating the updating of a device image;

FIG. 5A is a flow diagram depicting an illustrative routine for transmitting a set of device images implemented by the device image manager shown in FIG. 1;

FIG. 5B is a flow diagram depicting an illustrative provision management routine implemented by the device image manager shown in FIG. 1; and

FIG. 6 is a flow diagram depicting an illustrative routine for provisioning implemented by a computing device shown in FIG. 1.

DETAILED DESCRIPTION

Generally described, aspects of the present disclosure relate to the management of device imaging processes occurring on multiple computing devices. Specifically, systems and methods are disclosed which facilitate imaging of multiple computing devices. In one aspect, device images are transmitted to a number of computing devices by a device image manager. Illustratively, each device image can correspond to a software configuration of a computing device, such as an operating system or operating system kernel. The computing devices then store the devices images such that they may be accessed at a later time. Illustratively, the computing devices may store the device images on a local hard disk drive of each of the computing devices. The device image manager may then detect that some or all of the computing devices should load at least one of the images. Once a determination is made that some or all of the computing devices under the control of the device image manager should load an image, the device image manager may send a command to the relevant computing devices to load a specified device image. The computing device may then determine whether the specified device image exists within locally accessible storage. If so, the computing device may process the device image without requiring transmission of the specified image from the device image manager in response to the command from the device image manager. In this way, network resources required to load one or more device images onto multiple computing devices may be reduced.

In some embodiments, the device image manager may further be capable of selecting one or more device images to be placed on a computing device. This determination may be made, at least in part, according to the characteristics of the computing device, such as the computing device’s hardware, network configuration, performance, or location. Additionally, the determination may be made, at least in part, according to the intended function or functions of the computing device. In still more embodiments, the device image manager may be further capable of updating device images stored by computing devices.

Specific embodiments and example applications of the present disclosure will now be described with reference to the drawings. These embodiments and example applications are intended to illustrate, and not limit, the present disclosure.

FIG. 1 is a block diagram depicting an illustrative environment 100 for managing provisioning of multiple computing devices. The illustrated environment includes multiple computing devices 106. In the illustrated environment, some computing devices 106 are arranged into collections of computing devices 110. Illustratively, a collection of computing devices may correspond to multiple physical computing devices arranged within a shared housing. In other embodiments, a collection of computing devices may correspond to multiple virtual computing devices, such as virtual machine instances, which share a single physical computing device. As will be appreciated by one skilled in the art, a collection of computing devices may correspond to collections of physical computing devices, virtual computing devices, or combinations of the two. Collections of computing devices 110 may further include a control device 112. This control device 112 can correspond to a computing device which functions to control some or all activities of computing devices 106 within the collection of computing devices 110. Each computing device 106 has access to a storage device 108. As depicted in FIG. 1, these storage devices 108 may be individual to the computing devices 106 or shared between a number of computing devices 106, such as computing devices 106 within a collec-

tion of computing devices **110**. These storage devices **108** may correspond to a hard disk drive, either internal or external to a computing device, to network accessible storage, or to other storage devices accessible by the computing devices **106** or control devices **112**.

The illustrated environment further includes a device image manager **102** in communication with the computing devices **106** via a network **104**. The network **104** may be, for instance, a wide area network (WAN), a local area network (LAN), or a global communications network.

Illustratively, the device image manager may manage provisioning of computing devices **106** by transmitting device images to the computing devices **106** or the collection of computing devices **110**. As used herein, provisioning of a computing device refers to the processes of accessing and loading a device image onto the computing device. In some embodiments, such provisioning may alter the software of a computing device such that the device may carry out additional or different functionality, generally referred to as re-provisioning. The computing devices **106** or collection of computing devices **110** may store the transmitted set of device images in storage **108**. In this manner, the device images may later be accessed by the computing devices **106** or collection of computing devices **110** independent of the device image manager **102**. The device image manager may further manage provisioning by detecting provision events and sending a provision command to the computing devices **106** or collection of computing devices **110**. The provision events may correspond to an automated detection that one or more computing devices **106** or collection of computing devices **110** should be provisioned, or to a received input specifying devices to be provisioned. The device image manager may then cause the computing devices **106** or collection of computing devices **110** to load a specified device image by transmitting a provision command. An example of the interaction between components in a provisioning process will be described in more detail with respect to FIGS. 2A and 2B, below.

In some embodiments, the device image manager **102** may be operable to determine which device images to transmit or cause to be transmitted based on identified configuration information of the computing devices **106** or the collection of computing devices **110**. The configuration information may correspond to hardware profile information, network profile information, benchmarking information, or computing device location information. Illustratively, configuration information may correspond to hardware components of the computing device **106**, such as information corresponding to a central processing unit of the computing device **106**. Configuration information may further correspond to an IP address of the computing device **106**. Additional examples of configuration information will be discussed with respect to FIG. 3A, below. The device image manager **102** may use the configuration information to determine a set of device images which should be transmitted to the computing devices **106** or the collections of computing devices **110**. One example illustration of an interaction by which the device image manager **102** may determine a set of device images based on configuration information will be discussed in more detail with respect to FIGS. 3A and 3B, below. In other embodiments, the set of device images may be determined in advance or otherwise preset, such that a common set of device images is transferred to each computing device **106** or group of computing devices. In still more embodiments, the devices images to be transferred may be specified manually in response to input received from an operator or client of the device image manager **102**.

In some embodiments, the device image manager **102** may be operable to update device images stored by computing devices **106** or collection of computing devices **110**. The device image manager **102** may determine that an update exists for one or more of the device images which have been transmitted to computing devices **106** or collection of computing devices **110**. The device image manager may determine which devices are currently storing the non-updated device image and transmit a device image update to the relevant devices. One example illustration of such an update process will be discussed with respect to FIG. 4, below.

With reference to FIG. 2A, an illustrative interaction for provisioning multiple computing devices will be described. As illustrated in FIG. 2A, the device image manager **102** may transfer a set of device images to computing devices **106**. As previously described, the set of device images may be selected by an operator of the device image manager **102**, may be a predetermined set of device images, or may be determined based on configuration information of a computing devices **106** as will be described in FIGS. 3A and 3B, below. Additionally, the device image manager **102** may transmit a set of device images to a single computing device, such as computing device **106A**, or to multiple computing devices, such as computing devices **106B** and **106C**. Transmission of the device images to computing devices **106A-C** may occur simultaneously, sequentially, or in any order. Additionally, transmission of device images may occur intermittently, such that portions of the set of device images are transmitted separately. For example, the device image manager **102** may transmit device images or portions of device images during periods of lowered activity across the network **104** or portions of the network. In some embodiments, the device image manager **102** may transmit device images or portions of device images during periods of lowered activity by the computing devices **106**. One skilled in the art will appreciate that transmission of the device images to computing devices **106** may be accomplished by additional or alternative timings or mechanisms while not departing from the scope of the current disclosure. Subsequent to receiving the device images, the computing devices may store the device images in a device image storage, such as storage **108** of FIG. 1.

With continued reference to FIG. 2A, the device image manager **102** may detect that a provision event has occurred. In some embodiments, this detection will correspond to a provision request by an operator of the device image manager, a client, or another entity. This provision request may correspond to a consumer demand for resources at a data center housing computing devices **106**. In other embodiments, this detection may be automated, such that the device image manager **102** or another computing device detects that a provision should occur. One skilled in the art will appreciate that provisioning may occur in response to a variety of events. By way of non-limiting example, a provision event may correspond to an addition of one or more computing devices, such that these computing devices must be provisioned in order to be utilized. A provision event may further correspond to a reallocation of computing device resources, such as when demand is increased or decreased for certain functionality provided by provisioning. By way of further example, a provision event may correspond to a detection that utilization of computing devices providing a specific functionality has reached a predetermined threshold. When such a threshold has been reached, a provision event may indicate desirability to reconfigure additional computing devices with the specific functionality in order to reduce overall utilization levels and provide additional computing resources. A provision event may still further correspond to a detection of malfunction on

5

one or more computing devices. Additionally, a detected provision event may correspond to a detection of a change in configuration of a computing device.

After detection of a provision event, the device image manager 102 may transmit a provision command to computing devices 106A-C. This provision command may specify the type of provisioning which is to occur, the specific device image which is to be loaded, or other information sufficient to allow the computing device 106 to load a device image corresponding to the provision command. As will be appreciated by one skilled in the art, transmission of provision commands to computing devices 106A-C may occur simultaneously or in any order. In some embodiments, provision commands may be transferred only to a single computing device 106, such as computing device 106A, or group of computing devices 106, such as computing devices 106B. As will be appreciated by one skilled in the art, the provision command may be transmitted through a variety of processes.

By way of non-limiting example, a provision command may correspond to a command sent from the device image manager 102 via an application programming interface (API). A provision command may further correspond with boot instructions to a computing device 106. Such boot instructions may direct the computing device 106 to take specified actions after a reboot of the computing device. For example, a computing device 106 may be configured to receive boot instructions from a device image manager 102 on each boot of the computing device 106. On detection of a provision event, the boot image manager 102 may modify the boot instructions such that the computing device 106 loads a specified device image when it next boots.

In response to receiving a provision command, the computing devices 106 then process the corresponding device image. If the corresponding device image was previously transmitted to the computing device 106 and stored in device image storage, the computing device 106 may process the device image without requiring retransmission of the device image from the device image manager 102 or another location. If the device image was not previously stored by the computing device 106, transmission of the device image may be required from another source.

In some embodiments, processing the device image may correspond to copying all or part of a specified device image to an image destination storage of the computing device 106. The image destination storage corresponds to a storage of the computing device 106 on which the device image is to be loaded. In some embodiments, this image destination storage may be distinct from the device image storage 108, which stores the received device images. Image destination storage may correspond to random access memory (RAM) of the computing device 106, to a hard disk drive of the computing device 106, or to other storage accessible by the computing device 106. After copying all or part of the specified device image to the image destination storage, the computing device 106 may run the software now located on the image destination storage. This software corresponds with the software contained within the specified device image. In some embodiments; the device image may contain an operating system, operating system kernel, or operating system configuration. As will be appreciated by one skilled in the art, device images may contain a variety of software or software configurations.

In still more embodiments, processing of a specified device image may require loading of more than one device image by a computing device 106. By way of illustrative example, a computing device 106 may utilize a temporary device image containing a temporary operating system. The computing device may load this temporary operating system into RAM,

6

and reboot to operate via the temporary operating system. The temporary operating system may then load a final device image into a non-temporary storage of the computing device 106. The temporary operating system may then cause the computing device 106 to reboot and load the software contained within the final device image from the non-temporary storage.

In some embodiments, loading of multiple device images may be required gain unrestricted write access to an image destination storage location. Such access may be restricted, for example, by a current operating system which is operating from the image destination storage. By loading an intermediate device image, such as one containing a temporary operating system, unrestricted access to the image destination storage may be gained. The computing device 106 may then load the desired device image containing the new operating system onto the image destination storage.

As will be appreciated by one skilled in the art, the provision command may correspond to a single command, or to series of commands which cause the processing of a device image or images. For example, where intermediary device images must be processed in order to load a final desired device image, the initial provision command may specify each device image that must be loaded, only a portion of the device images which must be loaded, or only the first device image to be loaded. The device image manager 102 may subsequently communicate with the software loaded by intermediary device images to specify additional intermediary device images or the final desired device image. One of ordinary skill in the art will appreciate that a variety of provision commands and image loading techniques may be utilized without departing from the scope of the present disclosure.

With reference now to FIG. 2B, an illustrative interaction for provisioning multiple computing devices 106 contained within collections of computing devices 110 will be described. As illustrated in FIG. 2A, the device image manager 102 may transfer a set of device images to a collection of computing devices 110 which contain a control device 112. As previously described, the set of device images which are transferred may be selected by an operator of the device image manager 102, may be a predetermined set of device images, or may be determined based on configuration information of a collection of computing devices 110 or computing devices 106 contained within the collection of computing devices 110. The number of device images transferred to a collection of computing devices 110 may increase where the computing devices 106 within the collection have many potential functionalities. An illustrative example for determining device images based on configuration information will be described in FIGS. 3A and 3B, below. Transmission of the device images to the collection of computing devices 110 may occur simultaneously or non-simultaneously. Additionally, as described above, transmission of device images may occur intermittently, such that portions of the set of device images are transmitted separately. For example, the device image manager 102 may transmit device images or portions of device images during periods of lowered activity across the network 104 or portions of the network. The device image manager 102 may further transmit device images during periods of lowered activity by a collection of computing devices 110. One skilled in the art will appreciate that transmission of the device images to a collection of computing devices 110 may be accomplished by additional or alternative timings or mechanisms while not departing from the scope of the current disclosure. Subsequent to receiving the device images, the collection of computing devices 110 may store the device images in device image storage 108. In some embodiments,

storing of the device images may be accomplished by the control device 112. In still more embodiments, the control device 112 may transfer some or all of the device images to an individual storage of one or more of the computing devices 106.

With continued reference to FIG. 2B, the device image manager 102 may detect that a provision event has occurred. As described with reference to FIG. 2A, detection may correspond to a provision request by an operator, a client, or another entity, or may be automated such that the device image manager 102 or another computing device detects that a provision should occur without requiring human input. Similarly to as described above with respect to FIG. 2A, a detection of a provision event may correspond to an addition of one or more collections of computing devices 110 or computing devices 106 with a collection of computing devices 110, to a reallocation of computing device resources, a detection of a change in configuration of collections of computing devices 110, or to a detection of a malfunction of one or more collections of computing devices 110.

After detection of a provision event, the device image manager 102 may transmit a provision command to control devices 112 within the collections of computing devices 110. This provision command may specify parameters of the provisioning, such as which computing devices 106 should be provisioned, what image should be loaded into each computing device 106, or other information sufficient to allow the control device 112 to provision the computing devices 106. As will be appreciated by one skilled in the art, transmission of provision commands to collections of computing devices 110 may occur simultaneously or in any order.

In response to receiving a provision command, in one embodiment, the control devices 112 may then cause the loading of corresponding device image onto appropriate computing devices 106. If the corresponding device image was previously transmitted to the collection of computing devices 110 and stored in device image storage 108, the collection of computing devices 110 may process the device image without requiring retransmission of the device image from the device image manager 102 or another location. If the device image was not previously stored by the collection of computing devices 110, such as in storage 108, transmission of the device image may be required from an alternative source outside the collection of computing devices 110.

In other embodiments, the control device 112 may forward an appropriate provision command to each individual computing device 106. The control device 112 may determine the appropriate provision command based on the provision command received from the device image manager 102. For example, the device image manager 102 may transmit a provision command to control device 112A specifying that computing device 106A is to be provisioned with a first device image. The provision command may further specify that computing device 106B is to be provisioned with second device image. In this illustrative example, control device 112A may determine two separate commands for the computing devices 106A and 106B, each command specifying the appropriate device image to be loaded by the individual computing device 106. One illustration of an individual computing device 106 provisioning itself based on a provision command is discussed with respect to FIG. 2A, above.

In other embodiments, the control device 112 may directly control provisioning of one or more computing devices 106. In these embodiments, the control device 112 may copy all or part of the device image to an image destination storage of the computing device 106. The image destination storage may correspond to random access memory (RAM) of the comput-

ing device 106, to a hard disk drive of the computing device 106, or to other storage accessible by the computing device 106 in order to load the software included within the device image. For example, the control device 112 may directly load a device image onto an image destination storage accessible by the computing device 106. This direct loading may not require any processing by the computing device 106. In some embodiments, the computing device 106 may not be aware that the control device 112 is loading a device image onto an image destination storage. The control device 112 may further reboot the computing device 106 in order to cause the device to load the device image as copied to the image destination storage. In some embodiments, causing the computing device 106 to load the device image may correspond to a modification of a boot instruction of the computing device 106. In some embodiments, utilizing a control device 112 in order to provision a computing device 106 may eliminate or reduce the need to use temporary device images in order to load a desired device image.

With reference now to FIG. 3A, an illustrative interaction for determining a set of device images based on configuration information of computing devices 106 will be described. As illustrated in FIG. 3A, computing devices 106A-C may transfer device configuration information to device image manager 102. In one embodiment, the device configuration information may be based on or include one or more device profiles, such as a hardware profile, network profile, benchmark profile, or location profile, or other configuration information associated with the computing devices 106. These profiles will be discussed in greater detail below. Illustratively, computing devices 106 may determine device configuration information by querying or identifying hardware components of the computing devices 106, performing benchmarking or testing on the computing devices 106, obtaining data previously generated or associated with the computing devices 106, or inferring information based on any of the above or any other information associated with the computing devices 106. The process of determining device configuration information may be guided by or based upon device provisioning preferences, which may comprise information identifying or interpretable to identify one or more profiles, types, or specific pieces of device configuration information for the computing devices 106 to obtain or determine.

Specifically, and for purposes of example, device configuration information may include hardware profile information associated with a hardware component of a computing device including, but not limited to, a network interface controller (NIC), a central processing unit (CPU), a computing device basic input/output system (BIOS), computing device firmware, a motherboard, a daughterboard, an expansion board, a bus, a computer readable media drive, a memory, a storage device, a computer display, an input or output interface, or a computing device enclosure. Device configuration information may further include network profile information or location information corresponding to or associated with a or logical location in a network or data center, including, but not limited to, information associated or determined from with a MAC address, a static IP address, a logical device group identifier, a control device identifier, a serial number or ID corresponding to one or more components of the computing device 106, a data storage location, information identifying one or more specific instances of a device image manager 102, or other information that may indicate or be used to infer a logical or location. Device configuration information may still further include benchmark profile information associated with one or more performance benchmarks of a computing

device **106** or computing device component, including but not limited to, processor performance; energy usage; bus speed; drive or data storage component throughput or seek time; network bandwidth, throughput, or latency to one or more remote or local network locations; or other benchmark data associated with the computing device.

Illustrative, the computing devices transfer device configuration information to the device image manager **102**. Transferring device configuration information may include transmitting data comprising the device configuration information across a direct connection or network **104** of FIG. **1**, transmitting or identifying a pointer or reference to a location of device configuration information stored on a storage device or computer memory, or any combination thereof. Upon receipt, the device image manager **102** may utilize the device configuration information to select a set of device images based on hardware qualifications associated with the device images. Illustratively, hardware qualifications may be minimum hardware requirements needed by a computing device **106** in order to run software contained within a device image. In one embodiment, the device image manager **102** may compare device configuration information corresponding to a computing device **106** to a set of hardware qualifications associated with a number of device images in order to identify a set of device images capable of be loaded on the computing device **106**. In a further embodiment, the device image manager **102** may determine a set of device images by comparing network profiles, location profiles, benchmark profiles, or additional non-essential hardware profiles included in the device configuration information to the set of hardware qualifications.

For example, the device image manager **102** may compare the device configuration information corresponding to the computing device **106A** to a set of hardware qualifications associated with a group of fifty device images to identify a set of ten different device images that may be loaded on computing device **106A**. In an alternate embodiment, the device image manager **102** may identify all device images as potentially able to be loaded on the computing device **106A**, or may skip the step entirely. To continue this example, the device image manager **102** may further compare the device configuration information to the set of hardware qualifications to narrow this initial set of device images down to a set of device images appropriate for computing device **106A**. For example, if device configuration information such as a network profile associated with computing device **106A** indicates that the computing device **106A** is located in a network subnet that is dedicated to servicing client DNS queries, a comparison of the device configuration information with the set of hardware qualifications may cause the device image manager **102** to identify three device images out of the initial set of ten that correspond to operating systems, software, or operating system configurations for responding to DNS queries.

As another example, if device configuration information such as a location profile associated with the computing device **106A** indicates that the computing device **106A** is located in a data center in an area with high energy costs, a comparison of the device configuration information with the hardware qualification information may cause the device image manager **102** to identify two device images out of the initial set of ten that correspond to operating systems, software, or operating system configurations configured for low power usage. As still another example, if device configuration information such as a benchmark profile associated with the computing device **106A** indicates that the computing device **106A** is associated with a high network latency to a particular group of network clients, a comparison of the device configuration

information with the hardware qualification information may cause the device image manager **102** to identify one device image out of the initial set of ten that corresponds to an operating systems, software, or operating system configuration configured for long term backup or storage of content, rather than rapid service of content to the group of network clients. As still another example, a comparison of the device configuration information with hardware qualifications may cause the device image manager **102** to identify all ten of the initial set of device images as potential images for the computing device **106A**.

As yet another example, each device image may be associated with one or more specific categories. Illustratively, these categories may correspond to functionalities of a computing device **106**, such as a database server or web services host server. Device images may be manually associated with specific categories. In some embodiments, device images may be automatically associated with a category based on software contained within the device image or hardware requirements associated with the device image. The device image manager **102** may utilize these categories in order to determine a set of device images which may be loaded onto a computing device **106**. Illustratively, the device image manager **102** may use configuration information of the computing device **106A** to determine that the computing device **106A** is suitable for loading device images within a database server category. This determination may be based on configuration information indicating computing device **106A** has a large amount of storage space. The device image manager **102** may further determine that the computing device **106A** is also suitable for loading device images contained within a web services host category. This determination may be based on configuration information indicating that computing device **106A** has a large amount of network resources. The device image manager **102** may then determine that the set of device images to be transmitted to computing device **106A** is the set of device images contained within the categories "database server" and "web services host."

Still further, a set of device images may be determined based in part on past, present, or estimated future demand for specific functionality. For example, where demand for a specific functionality has historically increased, the device image manager **102** may send to a computing device **106A** a device image which allows the device to implement that specific functionality. As a further example, where estimations of future user demand indicate that demand is falling for a specific functionality, the device image manager **102** may be less likely to send to a computing device **106A** a device image which allows the device to implement that specific functionality. In some embodiments, the device image manager **102** may further gather information as to the number of computing devices **106** which are currently storing a device image implementing a specific functionality. For example, where future demand is expected to decrease for that specific functionality, and the number of computing devices **106** storing the device image is high, the device image manager **106** would be less likely to transmit the device image to an additional computing device **106**. Conversely, where demand is expected to rise for a functionality, and the number of computing devices **106** storing a device image or images implementing that functionality is low, a device image manager may be more likely to transmit that device image or images.

Although the selection of a first set of device images that may be loaded on the computing device **106A** and the selection of a second subset of device images is described here for the purpose of example as two logical steps, it should be understood that the process of identify one or more appropri-

11

ate device images may be implemented or executed in one, two, or any other number of logical steps by the device image manager 102.

After the device image manager 102 has selected a set of device images for the computing devices 106, the device image manager 102 may cause the selected device image to be transferred to the computing devices 106.

With reference now to FIG. 3B, an illustrative interaction for determining a set of device images based on configuration information of collections of computing devices 110 will be described. As illustrated in FIG. 3B, control devices 112 may transfer device configuration information to the device image manager 102. As described above with reference to FIG. 3A, the device configuration information may be based on or include one or more device profiles, such as a hardware profile, network profile, benchmark profile, or location profile, or other configuration information associated with collections of computing devices 110, control devices 112, or computing devices 106 which exist within collections of computing devices 110. Illustratively, the control devices 112 may determine device configuration of the computing devices 106, the control devices 112, or the collections of computing devices 110 by directly querying or identifying hardware information of the relevant devices. In some embodiments, the control devices 112 may query the computing devices 106, which may provide hardware information to the control devices 112. As described above with reference to FIG. 3A, the process of determining device configuration information may be guided by or based upon device provisioning preferences, which may comprise information identifying or interpretable to identify one or more profiles, types, or specific pieces of device configuration information for the collections of computing devices 110 to obtain or determine.

After the control devices 112 have determined their associated device configuration information, the device configuration information is transferred to the device image manager 102. Transferring device configuration information may include transmitting data comprising the device configuration information across a direct connection or network 104 of FIG. 1, transmitting or identifying a pointer or reference to a location of device configuration information stored on a storage device or computer memory, or any combination thereof.

After the device image manager 102 has access to device configuration information corresponding to the collections of computing devices 110, the device image manager may utilize the device configuration information to select a set of device images based on hardware qualifications associated with the device images. In one embodiment, the device image manager 102 may compare device configuration information corresponding to a computing device 106 to a set of hardware qualifications associated with a number of device images in order to identify a set of device images capable of be loaded on the computing device 106 of the collection of computing devices 110. In a further embodiment, device image manager 102 may determine a set of device images by comparing network profiles, location profiles, benchmark profiles, or additional non-essential hardware profiles included in the device configuration information to the set of hardware qualifications.

For example, the device image manager 102 may compare device configuration information corresponding to a collection of computing devices 110 to a set of hardware qualifications associated with a group of fifty device images to identify a set of ten different device images that may be loaded on the collection of computing devices 110. In an alternate embodiment, the device image manager 102 may identify all device images as potentially able to be loaded on a collection of

12

computing devices 110, or may skip the step entirely. To continue this example, the device image manager 102 may further compare the device configuration information to the set of hardware qualifications to narrow this initial set of device images down to a set of device images appropriate for a collection of computing devices 110. For example, if device configuration information such as a network profile associated with a collection of computing devices 110 indicates that the collection of computing devices 110 is located in a network subnet that is dedicated to servicing client DNS queries, a comparison of the device configuration information with the set of hardware qualifications may cause the device image manager 102 to identify three device images out of the initial set of ten that correspond to operating systems, software, or operating system configurations for responding to DNS queries.

Similar to as described above with respect to FIG. 3A, each device image may be associated with one or more specific categories. The device image manager 102 may utilize these categories in order to determine a set of device images which may be loaded onto a collection of computing devices 110. Illustratively, the device image manager 102 may use configuration information of the collection of computing devices 110 to determine that the collection of computing devices 110 is suitable for loading device images within a database server category. The device image manager 102 may further determine that the collection of computing devices 110 is also suitable for loading device images contained within a web services host category. The device image manager 102 may then determine that the set of device images to be transmitted to the collection of computing devices 110 is the set of device images contained within the categories “database server” and “web services host.” Additionally, the device image manager 112 may associate individual computing devices within the collection of computing device 110 with specific categories, as described with reference to FIG. 3A. In some embodiments, the device image manager 112 may determine an appropriate set of device images based on all categories of device images which may be loaded on any computing device 106 within a collection of computing devices 110.

As another example, if device configuration information such as a location profile associated with a collection of computing devices 110 indicates that the collection of computing devices 110 is located in a data center in an area with high energy costs, a comparison of the device configuration information with the hardware qualification information may cause the device image manager 102 to identify two device images out of the initial set of ten that correspond to operating systems, software, or operating system configurations configured for low power usage. As still another example, if device configuration information such as a benchmark profile associated with a collection of computing devices 110 indicates that the collection of computing devices 110 is associated with a high network latency to a particular group of network clients, a comparison of the device configuration information with the hardware qualification information may cause device image manager 102 to identify one device image out of the initial set of ten that corresponds to an operating systems, software, or operating system configuration configured for long term backup or storage of content, rather than rapid service of content to the group of network clients. As still another example, a comparison of the device configuration information with hardware qualifications may cause the device image manager 102 to identify all ten of the initial set of device images as potential images for a collection of computing devices 110. Although the selection of a first set of device images that may be loaded on collection of computing

devices **110** and the selection of a second subset of device images is described here for the purpose of example as two logical steps, it should be understood that the process of identifying one or more appropriate device images may be implemented or executed in one, two, or any other number of logical steps by the device image manager **102**.

Once the device image manager **102** has selected a set of device images for the collection of computing devices **106**, device image manager **102** may cause the selected device image to be transferred to the collection of computing devices **110**.

With reference now to FIG. 4, an illustrative interaction for updating a device image will be described. As illustrated in FIG. 4, the device image manager **102** may determine a device image update event. This event may correspond to any event which notifies the device image manager **102** that a device image update exists for a device image which is managed by the device image manager **102**. By way of non-limiting example, such an event may correspond to an update received from a manufacturer or developer of software contained within the device image, to a modification of software or software configuration by a client or operator of the device image manager, or to other modification of device images. In order to update a device image or images stored by computing devices **106**, the device image manager **102** may first determine which computing devices **106** have stored the non-updated image. One skilled in the art will appreciate that such a determination may be made through a variety of processes without departing from the scope of the present disclosure. For example, the device image manager **102** may keep a record of all computing devices **106** which have previously received a non-updated device image. Further, the device image manager **102** may query some or all of the computing devices **106** in order to determine the status of device images stored by the computing devices **106**. In some embodiments, a request from a computing device **106** may signal that the computing device **106** requires a device image update. As illustrated in FIG. 4, the device image manager **102** may determine that computing device **106A** requires a device image update.

The device image manager **102** may next transfer the device image update to the determined computing device **106A**. The device image update may correspond to a full copy of the updated device image, to a portion of the updated device image, or to other information sufficient to allow the computing device **106A** to update a device image. By way of non-limiting example, a device image update may correspond to a patch file for a device image, or to an executable which updates the device image when processed by the computing device **106A**. In some embodiments, device image manager may transmit a location of a device image update file to the computing device **106A**. The location of the image file may, in some instances, correspond to other computing devices **106** or to other storage devices. Where a location of a file is specified, the computing device **106** may further retrieve the image update file from the specified location.

After reception of the device image update, the computing device **106A** may then update the device image using the device image update. One skilled in the art will appreciate that the method used to update the device image may vary according to the format of the device image update. For example, where the device image update is an executable file or a patch, the computing device **106A** may execute the file. Where the device image update is a new device image, the computing device **106A** may replace the non-updated device image in device image storage with the updated device image **106A**.

Other variations will be recognized by one skilled in the art without departing from the scope of this disclosure.

With reference now to FIG. 5A, one embodiment of a routine **500** implemented by a device image manager **102** for sending a set of device images to a set of computing devices will be described. At block **502**, the device image manager **102** determines or identifies a set of computing devices **106** to which device images will be transmitted. In some embodiments, the device image manager **102** may determine that all computing devices **106** should receive device images. In other embodiments, the device image manager **102** may determine that only computing devices **106** meeting certain criteria should receive device images. By way of non-limiting example, such criteria may correspond to network, hardware, benchmarking, location, or current provisioning information. Such criteria may further correspond to manual criteria specified by an operator of the device image manager **102**. Other criteria for determining applicable computing devices **106** will be appreciated by one skilled in the art. In still more embodiments, a set of computing devices **106** may be determined by receiving input from an operator or client of the device image manager **102**.

At block **504**, the device image manager **102** determines the set of device images which should be sent to the determined set of computing devices **106**. As discussed above with respect to FIGS. 3A and 3B, the device image manager **102** may use configuration information to determine a set of device images which should be transmitted to computing devices **106** or the collections of computing devices **110**. In some embodiments, the device image manager **110** may utilize configuration information to associate a computing device **106** or collection of computing devices **110** with categories of device images, such as categories corresponding to specific functionality. In other embodiments, the set of device images may be determined in advance or otherwise preset, such that a common set of device images is transferred to each computing device **106** or collection of computing devices **110**. In still more embodiments, the device images to be transferred may be specified manually in response to input received from an operator or client of the device image manager **102**. Though the determination of the set of device images is discussed subsequent to a determination of the set of computing devices, one skilled in the art will appreciate that the order of these blocks may be interchanged, such that a set of device images is determined prior to determining a set of computing devices **106**.

At block **506**, once both the set of computing devices and set of device images has been determined, the device image manager **102** may transmit or cause the transmission of the set of device images to the set of computing devices. As discussed previously with respect to FIG. 1, the device image manager **102** may send all device images in the set of device images during the same period, or may transmit portions of the set of device images over discreet periods of time. For example, the device image manager **102** may transmit images during periods of low network utilization, encouraging efficient use of network resources. Further, the device image manager **102** may send the set of device images to individual computing devices **106** simultaneously or at different times. These different times may correspond to periods of low activity for the individual computing devices **106**. In some embodiments, the device image manager **102** may utilize a peer to peer network model to distribute device images. In such a peer to peer model, computing devices **106** may, after reception of a device image or images, act to retransmit the device image or images to other computing devices **106** which have not yet received them. One skilled in the art will

15

appreciate that a variety of processes may be used to send the set of device images without departing from the scope of the current disclosure. The routine **500** may then end.

With reference now to FIG. **5B**, one embodiment of a routine **550** implemented by a device image manager **102** for provisioning multiple computing devices will be described. Illustratively, this routing may be implemented subsequent to a routine which causes the multiple computing devices to store a set of device images, such as routine **500** of FIG. **5A**. As will be appreciated by one skilled in the art, routine **500** of FIG. **5A** may be implemented multiple times before implementation of routine **550** of FIG. **5B**. Further, routines **500** of FIG. **5A** and **550** of FIG. **5B** may be implemented simultaneously on the same or different computing systems. Still further, routines **500** of FIG. **5A** and **550** of FIG. **5B** may be implanted in any order. For example, routine **550** of FIG. **5B** may occur before routine **500** of FIG. **5A**, as long as the computing device which is to be provisioned has access to the relevant stored device image.

With continued reference to FIG. **5B**, at block **552**, the device image manager **102** detects that a provision event has occurred. Similarly to as discussed above with respect to FIG. **2A**, a provision event may correspond to a number of events. For example, a provision could be requested by an operator of the device image manager **102**, a client, or another entity. In some embodiments, the device image manager **102** or another computer may detect a change in configuration or a malfunction of one or more computing devices **106**. In other embodiments, the device image manager **102** or another computer may detect the addition of one or more computing devices **106** or collections of computing devices **110**. In still more embodiments, a provision event may be detected by the device image manager **102** based on a determination that computing resources may be better allocated by provisioning one or more computing devices **106**. Such a determination may correspond to a detection that utilization of computing devices **106** providing a specific functionality has reached a predetermined threshold, such that provisioning may provide additional computing resources providing the specific functionality. One skilled in the art will appreciate that provisioning may occur in response to a variety of events.

At block **554**, the device image manager **102** determines the set of computing devices which are to be provisioned. In some embodiments, the computing devices which are to be provisioned may be based on the provision event which was detected. Provisioning may occur on a single computing device **106** or a group of computing devices **106**.

At block **556**, the device image manager **102** may optionally update a device image which has been stored by the computing devices **106** which are to be provisioned. This may correspond to requesting a device image update from the device image manager **102**. This device image update may then be used to update a device image prior to loading the device image.

At block **558**, a provision command is sent to the determined computing devices **106** to be provisioned. As discussed above with respect to FIG. **2A**, a provision command may be transmitted through a variety of processes. By way of non-limiting example, a provision command may correspond to a command sent from the device image manager **102** via an application programming interface (API). In some embodiments, a provision command may be transmitted with boot instructions to the determined set computing device **106** to be provisioned. As discussed above, a provision command may correspond with a command to load multiple device images in order to provision to a final desired device image. In some embodiments, a provision command may specify all device

16

images to be loaded. In other embodiments, the device image manager **102** may further communicate with the computing devices **106** in order to specify additional required device images. In still more embodiments, a computing device **106** may be preprogrammed to use specified intermediary device images, such as temporary operating systems, in order to process a final device image. The routine **550** may then end.

With reference to FIG. **6**, one example of a routine to provision based on multiple device images will be described. In FIG. **6**, a boot image must first be processed in order to provision the machine with the provision image. Such a boot image may be required, for example, where the provision image is to be placed on an image destination storage already in use by operating system of the computing device **106**. In some embodiments, a boot image may correspond to a temporary operating system which is loaded into RAM or other storage separate from the image destination storage. The temporary operating system may then write to previously inaccessible portions of the image destination storage.

With continued reference to FIG. **6**, at block **602** the computing device **106** obtains a number of device images from the device image manager **102**. These device images contain a number of software configurations which may be loaded by the computing device **106**. In some embodiments, these device images may correspond to various operating systems or operating system configurations usable by the computing device **106**. By way of non-limiting example, these operating systems or configurations may allow the computing device **106** to carry out functionalities such as a web services host, a file storage host, networking component, or database server. Other potential functionalities will be apparent to one skilled in the art.

After obtaining the device images, at block **604** the computing device **106** may store those device images on device image storage **108** such that they will be accessible in the future. As discussed above, storage **108** may correspond to a hard disk drive of the computing device **106**, to network attached storage, or to any storage medium sufficient to store the device images such that they are available during provisioning. As will be appreciated by one skilled in the art, blocks **602** and **604** may, in some embodiments, be repeated a number of times before the routine continues to block **606**.

At block **606**, the computing device **106** receives a provision command from the device image manager **102**. In this illustrative example, the provision command specifies multiple device images to be loaded by the computing device **106**, as described above in FIGS. **2A** and **2B**. These device images correspond to both a boot image and a provision image to be used during provisioning. The provision image contains an operating system able to be loaded by the computing device **106**. The boot image corresponds to a temporary operating system to be used by the computing device **106** in order to gain unrestricted write access to a image destination storage device. In this illustrative example, the image destination storage device is distinct from the device image storage **108**. Either the boot image or the provision image may correspond with images previously received from a device image manager **102**, as described above in FIGS. **2A** and **2B**. Further, the boot image may be provided by a source other than the device image manager **102**, such as by an operator of the computing device **106**. Illustratively, the operator may provide a boot image corresponding to a temporary operating system from a universal serial bus hard drive. The computing device **106** may be further operable to determine whether device images are authorized to be loaded onto the computing device **106A**. For example, the computing device **106** may only load images that are signed with specified digital encryption keys. In

another example, the computing device **106** may only load images when authentication is received from a trusted source. Such a trusted source could be another network connected entity, such as another computing device **106**, or an operator of a computing device **106**. Illustratively, the operator of the computing device could be authenticated by use of a password, by use of biometric hardware, or by other authentication mechanisms. One skilled in the art will appreciate that different images could be associated with different security schemas or mechanisms, and that various authentication methods could be used without departing from the scope of the current disclosure. Further, as will be appreciated by one skilled in the art, the destination to which a device image is to be written may be required to be different than the source of the device image. This requirement may occur, for example, where loading the device image requires writing to the entirety of the destination storage device. In some embodiments, loading the device image may not require writing to the entirety of the destination storage device. In such embodiments, the image destination storage and the device image storage **108** may be the same storage device.

At block **608**, the computing device **106** determines whether the boot image specified in the provision command exists within local storage **108**. If the boot image does not exist within local storage, computing device **106** proceeds to block **612** and requests the necessary boot image from the device image manager **102**. Optionally, if the boot image does exist within local storage, the computing device **106** may update the boot image before proceeding. This may correspond to requesting a device image update from the device image manager **102**. This device image update may then be used to update a device image prior to loading the device image.

At block **614**, after gaining access to the boot image, the computing device **106** reboots itself in order to load the boot image. In this illustrative embodiment, the computing device **106** loads the boot image into RAM during the boot process. As described above, the boot image contains a temporary operating system which enables the computing device to gain unrestricted write access to an image destination storage device on which the provision image will be loaded. The boot image further contains code which allows the computing device **106** to load the provision image.

At block **616**, the computing device **106** determines whether the provision image specified within the provision command exists within local storage **108**. If the provision image does not exist within local storage **108**, the computing device **106** may request the provision image from the device image manager **102**. Optionally, if the provision image does exist within local storage the computing device **106** may update the provision image before proceeding. As will be appreciated by one skilled in the art, though blocks **616-620** are described subsequently to blocks **608-14**, blocks **616-620** may be performed at any point between receiving a provision command at block **606** and loading the provision image.

At block **622**, the computing device **106** may provision itself with the provision image. In this illustrative embodiment, provisioning corresponds to writing the operating system and information contained within the provision image to a destination storage device associated with the computing device **106**. After writing the information within the provision image, the computing device **106** may reboot in order to load the operating system within the provision image. In this manner, the computing device **106** may be provisioned to carry out a desired functionality associated with the provision image.

At block **624**, after provisioning the computing device **106** with the provision image, the computing device **106** reboots itself in order to load the provision image. In this illustrative embodiment, the computing device **106** loads the software contained within the provision image from a storage. As such, the computing device **106** is able to implement the functionality associated with the provision image. The routine **600** may then end.

It will be appreciated by those skilled in the art and others that all of the functions described in this disclosure may be embodied in software executed by one or more processors of the disclosed components and mobile communication devices. The software may be persistently stored in any type of non-volatile storage.

Conditional language, such as, among others, “can,” “could,” “might,” or “may,” unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art. It will further be appreciated that the data and/or components described above may be stored on a computer-readable medium and loaded into memory of the computing device using a drive mechanism associated with a computer readable storing the computer executable components such as a CD-ROM, DVD-ROM, or network interface further, the component and/or data can be included in a single device or distributed in any manner. Accordingly, general purpose computing devices may be configured to implement the processes, algorithms, and methodology of the present disclosure with the processing and/or execution of the various data and/or components described above.

It should be emphasized that many variations and modifications may be made to the above-described embodiments, the elements of which are to be understood as being among other acceptable examples. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.

What is claimed is:

1. A system for management of provisioning of devices, comprising:
 - a computing system comprising a processor coupled to memory, the memory containing computer-executable instructions for execution by the computing system to implement a device image manager, wherein execution of the computer-executable instructions causes the computing system to:
 - transmit a set of device images to multiple computing devices, wherein the set of device images includes a

19

first device image which, when processed by a computing device, enables the computing device to carry out a first desired functionality and a second device image which, when processed by the computing device, enables the computing device to carry out a second desired functionality different from the first desired functionality, wherein transmission of the set of device images causes the multiple computing devices to store the set of device images locally to individual computing devices of the multiple computing devices, and wherein transmission of the set of device images to the multiple computing devices occurs independently from implementation, at the multiple computing devices, of instructions to provision the multiple computing devices with a device image of the set of device images;

subsequent to transmitting the set of device images, receive a request to provision the multiple computing devices according to the first desired functionality corresponding to the first device image; and

in response to the received request, transmit instructions to the multiple computing devices to process, at individual computing devices of the multiple computing device, the first device image of the set of device images, as previously received independently from implementation of instructions to provision the multiple computing devices with a device image of the set of device images and as stored locally to individual computing devices of the multiple computing devices, wherein the instructions specify the first device image, and wherein the processing enables each of the multiple computing devices to carry out the first desired functionality.

2. The system of claim 1, wherein at least one of the set of device images includes at least one of an operating system or an operating system kernel.

3. The system of claim 1, wherein the set of device images is determined at least in part based on a characteristic of the multiple computing devices.

4. The system of claim 3, wherein the characteristic of the multiple computing devices is determined at least in part based on at least one of a device hardware profile, a device network profile, a device benchmark profile, and device location information.

5. The system of claim 1, wherein causing the multiple computing devices to store the set of device images locally to individual computing devices of the multiple computing devices includes causing each of the multiple computing devices to store the set of device images at a data storage device internal to each computing devices.

6. The system of claim 5, wherein causing each of the multiple computing devices to store the set of device images locally includes causing each the multiple computing devices to store the set of device images on a hard disk drive.

7. The system of claim 1, wherein causing the multiple computing devices to store the set of device images locally to individual computing devices of the multiple computing devices includes causing at least one of the multiple computing devices to store the set of device images on a rack control device of a computing rack including the at least one computing device.

8. The system of claim 1, wherein causing the multiple computing devices to store the set of device images locally to individual computing devices of the multiple computing devices includes causing at least one of the multiple computing devices to store the set of device images on network accessible storage, the network accessible storage being dis-

20

tinct from the device image manager and sharing a local area network with the at least one computing device.

9. The system of claim 1, wherein execution of the computer-executable instructions further causes the computing system to:

receive an updated device image, the updated device image corresponding to an image of the set of device images; and

transmit the updated device image to the multiple computing devices, wherein transmission of the updated device image causes the multiple computing devices to store the updated device image.

10. The system of claim 9, wherein the instructions to process, at individual computing devices of the multiple computing device, the first device image of the set of device images, include instructions to request transmission of the updated device image.

11. The system of claim 1, wherein the instructions to process, at individual computing devices of the multiple computing device, the first device image of the set of device images, include instructions to verify the at least one device image.

12. A method for provisioning a set of computing devices, comprising:

determining, at an image management device, a set of device images including a first device image corresponding to a first set of instructions executable by a computing device to enable the computing device to carry out a first desired functionality and a second device image that corresponds to a second set of instructions executable by the computing device to enable the computing device to carry out a second desired functionality;

transmitting, from data storage associated with the image management device, the set of device images to a determined set of computing devices independent of implementation, at the set of computing devices, of instructions to provision the set of computing devices with a device image of the set of device images, wherein transmission of the set of device images causes the set of computing devices to store the set of device images locally to individual computing devices within the set of computing devices;

detecting at the image management device, a provision event, the provision event corresponding to a detection that at least one computing device of the set of computing devices should be enabled to implement the first desired functionality; and

transmitting, by the image management device, instructions to the at least one computing device to process, at the at least one computing device, the first device image of the previously stored set of device images, wherein the instructions specify the first device image, and wherein the processing enables the at least one computing device to carry out the first desired functionality.

13. The method of claim 12, wherein the set of device images is determined at least in part based on a characteristic of the at least one computing device.

14. The method of claim 12, wherein the characteristic of the at least one computing device is determined at least in part based on at least one of a device hardware profile, a device network profile, a device benchmark profile, and device location information.

15. The method of claim 12, wherein causing the set of computing devices to store the set of device images includes causing each of the set of computing devices to store the set of device images on at least one of a hard disk drive, a rack

21

control device, and a network accessible storage distinct from the source of the transmitted set of device images.

16. The method of claim **12** further comprising:

receiving an updated device image, the updated device image corresponding to an image of the set of device images; and

transmitting the updated device image to the set of computing devices, wherein transmission of the updated device image causes set of computing devices to store the updated device image.

17. The method of claim **16**, wherein transmitting instructions to the at least one computing device to process at least one locally stored device image of the stored set of device images includes transmitting instructions to the at least one computing device to request transmission of the updated device image.

18. The method of claim **12**, wherein transmitting instructions to the at least one computing device to process at least one locally stored device image of the stored set of device images includes transmitting instructions to the at least one computing device to verify the at least one device image.

19. The method of claim **12** further comprising:

determining a second set of device images, the second set of device images distinct from the previously determined set of device images;

transmitting the second set of device images to the determined set of computing devices;

detecting a second provision event; and

in response to the detected second provision event, causing at least one computing device of the set of computing device to process at least one device image of the second set of device images.

20. A system for provisioning a set of computing devices, comprising:

a computing system comprising a processor coupled to memory, the memory containing computer-executable instructions for executable by the computing system to implement a device image manager, wherein execution of the computer-executable instructions causes the computing system to:

determine a set of computing devices;

determine a set of device images including a first device image that, when processed by a computing device of the set of computing devices, enables the computing device to carry out a first desired functionality and a second device image that, when processed by the computing device of the set of computing devices, enables the computing device to carry out a second desired functionality;

independent of implementation, at the set of computing devices, of instructions to provision the set of computing devices with a device image of the set of device images, cause the set of computing devices to store

22

the set of device images locally to individual computing devices of the set of computing devices; and transmit instructions to the set of computing devices to process, at individual computing devices of the set of computing devices, the first device image of the previously stored set of device images, wherein the instructions specify the first device image, and wherein the processing enables each computing device of the set of computing device to carry out the first desired functionality.

21. The system of claim **20**, wherein the set of device images is determined at least in part based on a characteristic of the set of computing devices.

22. The system of claim **21**, wherein the characteristic of the set of computing devices is determined at least in part based on at least one of a device hardware profile, a device network profile, a device benchmark profile, and device location information.

23. The system of claim **20**, wherein causing the set of computing devices to store the set of device images includes causing each of the set of computing devices to store the set of device images on at least one of a hard disk drive, a rack control device, and a network accessible storage distinct from the device image manager.

24. The system of claim **20**, wherein execution of the computer-executable instructions further causes the computing system to:

receive an updated device image, the updated device image corresponding to an image of the set of device images; and

transmit the updated device image to the set of computing devices, wherein transmission of the updated device image causes the multiple computing devices to store the updated device image.

25. The system of claim **24**, wherein transmitting instructions to the set of computing devices to process the first device image of the stored set of device images includes transmitting instructions to the set of computing devices to request transmission of the updated device image.

26. The system of claim **20**, wherein transmitting instructions to the set of computing devices to process the first device image of the stored set of device images includes transmitting instructions to the set of computing devices to verify the first device image.

27. The system of claim **20**, wherein the execution of the computer-executable instructions further causes the computing system to determine a second set of device images, the second set of device images comprising device images distinct from the device images contained within the previously determined set of device images, wherein causing the set of computing devices to store the set of device images includes causing the set of computing devices to store the second set of device images.

* * * * *