

(19) **DANMARK**

(10) **DK/EP 3629169 T3**



Patent- og
Varemærkestyrelsen

(12) **Oversættelse af
europæisk patentskrift**

-
- (51) Int.Cl.: **G 06 F 9/50 (2006.01)**
- (45) Oversættelsen bekendtgjort den: **2021-03-22**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2020-12-16**
- (86) Europæisk ansøgning nr.: **19203212.6**
- (86) Europæisk indleveringsdag: **2013-02-01**
- (87) Den europæiske ansøgnings publiceringsdag: **2020-04-01**
- (62) Stamansøgningsnr: **18179638.4**
- (84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**
- (73) Patenthaver: **Amadeus S.A.S., 485 Route du Pin Montard, Sophia Antipolis, 06410 Biot, Frankrig**
- (72) Opfinder: **Dersy, Julien, Jefferson House, Hogs Back, Guildford, GU3 1AQ, Storbritannien**
Orton, David, 31 Woodside Avenue, Amersham Bucks, HP6 6BQ, Storbritannien
- (74) Fuldmægtig i Danmark: **Dennemeyer & Associates S.A, P.O. Box 700425, DE-81304 Munich, Tyskland**
- (54) Benævnelse: **KOMMUNIKATIONSSTYRING**
- (56) Fremdragne publikationer:
WO-A1-2011/022375
US-A1- 2009 043 921
US-A1- 2009 164 990

DESCRIPTION

Field of the Invention

[0001] The present invention relates to communications management and is more particularly, although not exclusively, concerned with the management of communication between peripheral devices within a virtualised communications system for airports.

Background to the Invention

[0002] In airport communication systems, a peripheral manager containing device drivers forms part of a client computer system which is geographically local to and is directly connected to one or more peripherals via a local communications system. Such a peripheral manager is constrained in establishing communications with central server-based applications due to the use of non-static internet protocol (IP) addresses and the requirement for IP address translation when the communication needs to traverse networks comprising a mix of public and private addressing schemes.

[0003] Within airport environments, workstations (including thin client devices, mobile/tablet devices) and peripheral devices may be shared for multiple airline usage. However, different airlines tend to utilise different systems and a problem arises where the same peripheral devices need to communicate with those different systems. One way of providing such communication requires the adaptation of existing client applications and print emulators to target remote hostnames in conjunction with addressing schemes based on fixed IP addresses. However, this is problematic for a number of reasons, including: the limited availability of public IP addresses; the complexity associated with security, firewalls and mobile data modes where 3G and/or 4G networks are used; the need for many routing configurations that need to be established and effectively managed; and the need to manage multiple stakeholders providing different elements of the communications, for example, airport authorities, network providers, airport information technology providers etc.

[0004] US-B-8032353 describes a bridged computer system in which a host computer executes peripheral driver software associated with a remote client computer and communications between the host computer and the remote client is processed using peripheral emulation of an audio codec. Audio signals are communicated between the host and remote computer using network interfaces provided in both the host and the remote computer. In the host computer, an audio interface is provided which connects to a communications network, and, in the remote computer, interfaces are provided for connecting to speakers and a microphone associated with the remote computer. In one embodiment, a peripheral emulator including an audio controller emulator and a codec emulator is provided in the host computer for connecting to the host computer network interface, the peripheral

emulator implementing transcoding functions, such as, compression of outbound audio data samples and/or decompression of inbound audio samples. At the remote computer, a link interface is provided which terminates audio links by providing a physical interface, bus transceiver logic and transport layer termination, the link interface transmitting output audio data and commands, receiving input audio data and responses, communicating codec control information to codec modules, distributing a sample clock, executing a power mode state machine, and providing bus reset and initialisation functions. Registers associated with the link interface maintain the configuration states of the codec modules and these registers are emulated at the host computer so that audio software on the host computer has rapid access to codec state information during driver initialisation or when the remote computer is disconnected. In another embodiment, a peripheral interface including a peripheral emulator in the host computer is connected to the network, and a peripheral controller is provided in the remote computer for controlling a peripheral. The peripheral emulator in this case interacts with the driver software to emulate the interface of the peripheral. The peripheral interface interacts through the network with the peripheral controller within the remote client computer, the peripheral controller comprising a remote driver or other software and/or hardware for coupling to the peripheral.

[0005] However, whilst the system described in US-B-8032353 uses peripheral emulation in a peripheral interface in the host computer, interfaces need to be provided on both sides of the communications network to convert the audio signals into and from appropriate data signals for transmittal over the communications network. Whilst this can readily be implemented over a known communications network, it is difficult to implement this over different types of networks where the communications network is not known, and more importantly, where several different types of networks need to be negotiated in a single communications link.

[0006] In systems where a host computer is required to be in communication with a remote client computer, multiple addressing schemes and translation across heterogeneous local and wide area networks, including mobile carrier networks, are required to enable communications between the host computer and the remote client computer. In this case, it is difficult to establish a reliable link irrespective of the multiple addressing schemes and the different types of networks required for the necessary communication.

[0007] Patent application US 2009/043921 relates to a method and system for virtualisation and redirection of input/output connections to peripheral devices in which PCI/PCIe level functions are used. Virtualisation may comprise emulating peripheral device controllers where a PCI/PCIe level function redirects the access to one or more peripheral devices to corresponding virtual devices.

[0008] Patent application WO 2011/022375 A1 relates to an input/output virtualisation system which provides a mechanism for sharing resources between a number of host computers using a managed transport fabric. A single device is shared across the numerous host systems within the system so that input/output virtualisation and host-to-host communication services can be provided to the host computers.

[0009] Patent application US 2009/164990 A1 relates to a method for deciding at run time which input/output mechanism a virtual machine should be using at any point in time, as well as a method for seamlessly switching between the different input/output mechanisms to the one most appropriate at that point in time.

Summary of the Invention

[0010] It is therefore an object of the present invention to provide a system and method for virtualising applications between a host computer and at least one remote client computer.

[0011] It is another object of the present invention to provide a method for managing the communication of peripheral devices deployed within an airport environment via a virtualised communications protocol.

[0012] It is a further object of the present invention to enable virtualised applications running on a server to establish communications with client-based peripheral managers and their associated peripheral devices.

[0013] In accordance with a first aspect of the present invention, there is provided a system for operating a peripheral device associated with a client from a server, the system comprising:-

a server being configured for running at least one application;

at least one client being configured to be connected to the server via a communications link;

at least one peripheral device associated with each client; and

a peripheral manager associated with each client and being configured for controlling each associated peripheral device;

characterised in that the server comprises an application virtualisation platform for virtualising at least one application running thereon; and each client comprises a client virtualisation platform for interacting with the application virtualisation platform;

and in that the system further comprises at least one peripheral manager simulator configured to establish a virtual channel through which communication is enabled between at least one application running on the server and at least one peripheral device associated with at least one client.

[0014] Advantageously, the provision of the virtualisation platforms on the server and each client enables applications running on the server to interact with one or more clients at the same time via virtual channels established between each client and the server. This means

that the server can be located at any suitable location with the clients located at various remote locations with respect to the server.

[0015] In addition, by the establishment of effectively dedicated virtual channels between each client and the server, different applications on the server can be accessed by each client and connected to relevant client-associated peripheral devices without having to compensate for different types of multiple addressing schemes and translation across heterogeneous local and wide area networks, including mobile carrier networks.

[0016] Each peripheral manager simulator preferably comprises at least one client peripheral manager simulator component associated with each client and a server peripheral manager simulator component associated with the server, the virtual channel being established between the server peripheral manager simulator component and the client peripheral manager simulator component.

[0017] In one embodiment, each server peripheral manager simulator component may be embedded in at least one application on the server. In another embodiment, each server peripheral manager may be connected to at least one application on the server.

[0018] Each application may further comprise a peripheral emulator and each server peripheral manager simulator component connects with each application by way of the peripheral emulator.

[0019] The system may further comprise an interface provided between the peripheral emulator and the server peripheral manager simulator component. In one embodiment, the interface may comprise a common use passenger processing system.

[0020] In one embodiment, the peripheral emulator may comprise a print emulator and at least one of the peripheral devices connected to at least one client comprises a printer.

[0021] The communication link may comprise at least a 3G or a 4G mobile telecommunications network.

[0022] In accordance with another aspect of the present invention, there is provided a method of connecting at least one application on a server to at least one peripheral device associated with at least one client connected to the server, the server having an application virtualisation platform and each client having a client virtualisation platform, the method comprising the steps of:

1. a) establishing a communications link between the server and at least one client;
2. b) virtualising communication protocols associated with each client and making the virtualised communications protocols available to the application virtualisation platform on the server; and
3. c) connecting at least one application on the server to at least one client using the client

communication protocol;

characterised in that the method further comprises the steps of:-

d) establishing a virtual channel between a server peripheral manager simulator component and at least one client peripheral manager simulator component within the communications link; and

e) operating at least one peripheral device associated with at least one client using the established virtual channel between the server peripheral manager simulator component and each client peripheral manager simulator component.

[0023] By establishing the virtual channel within the communications link, a reliable connection can be provided between the server and each client irrespective of the type(s) of communications network(s) at the locality of the client.

[0024] The method may further comprise the step of transferring data from at least one peripheral device connected to a client to the server using the established virtual channel between the client peripheral manager simulator component and the server peripheral manager simulator component.

Brief Description of the Drawings

[0025] For a better understanding of the present invention, reference will now be made, by way of example only, to the accompanying drawings in which:-

Figure 1 illustrates a block diagram of a server in accordance with the present invention;

Figure 2 is similar to Figure 1 but illustrates another configuration of the server in accordance with the present invention;

Figure 3 illustrates a block diagram of a client in accordance with the present invention;

Figure 4 illustrates a block diagram of a system in accordance with the present invention utilising the server of Figure 2 and the client of Figure 3;

Figure 5 illustrates a block diagram of a system in accordance with the present invention which comprises a server and a plurality of clients; and

Figure 6 is a flow chart illustrating the steps of operation of the system shown in Figure 4.

Description of the Invention

[0026] The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on scale for illustrative purposes.

[0027] The term "server" as used herein is intended to include any host computer that can connect to a communications network. In addition, the term "server" is also used to identify elements within the server.

[0028] The terms "remote client computer" or "client" as used herein are intended to include any computer or terminal that can connect to a server via a communications network. The term "client" is also used to identify elements within a remote client computer.

[0029] The term "emulator" as used herein refers to hardware or software that duplicates (emulates) the functions of a first computer system (for example, a client computer) in a second computer system (for example, a host computer) to provide an exact reproduction of the behaviour of the first computer system.

[0030] The term "simulator" as used herein refers to a hardware or software that provides a representation of the operation of a physical system over time using a model representing key characteristics and/or behaviours of the physical system.

[0031] The present invention is capable of exploitation by airports and any other parties that are involved in the provision of shared infrastructure to users in airport environments, including but not limited to, use in check-in and boarding areas, baggage claim desks and airline back offices. In addition, the present invention can also be utilised in non-airport environments.

[0032] This present invention provides a method for managing the communication with peripheral devices, such as printers, scanners and readers, deployed within an airport environment via a virtualised communications protocol extended over a network. Such a network may comprise of heterogeneous network types, including local area networks and wide area networks, the latter including fixed networks or 3G/4G data communications networks. (3G relates to the third generation of mobile communications for mobile telephones and comprises CDMA2000, a family of protocols that use code division multiple access systems, and Universal Mobile Telecommunications System (UMTS). CDMA is also known as International Mobile Telecommunications Multi-Carrier (IMT-MC). 4G relates to the fourth generation which provides mobile ultra-broadband internet access using, for example, Long Term Evolution (LTE).)

[0033] In particular, a method will be described which overcomes the challenges associated with communication between a remote central server and peripheral devices associated with airport-based clients where the communication may include multiple addressing schemes and

translation across heterogeneous local and wide area networks, including mobile carrier networks. As mentioned above, the present invention can be implemented in any system in which communication is to be established where multiple addressing schemes are needed between a remote central server and a local client and associated peripheral devices.

[0034] In the system of the present invention, a server is in communication with at least one remote client computer. A peripheral manager simulator simulates a peripheral manager on the server to communicate with a client peripheral manager in each remote client computer. The peripheral manager simulator preferably comprises a server peripheral manager simulator component and a client peripheral manager simulator component between which the virtual channel is established. The server peripheral manager simulator component and the client peripheral manager simulator component work together to simulate application communication from the server to each remote client computer so that the peripheral manager associated with each remote client computer can control the operation of its associated peripheral devices. Additionally, the virtual channel can also be used for data transfer between a peripheral device connected to the client and the server. In effect, the virtual channel is bi-directional.

[0035] Referring initially to Figure 1, a server 100 in accordance with the present invention is shown which has a server-based application virtualisation platform 110. A plurality of applications 120, 130, 140 are shown running on the server 100 but which are virtualised on the application virtualisation platform 110. An emulator 150, for example, a print emulator, is associated with one particular application 130, for example, a print application. The application 130 interfaces with the print emulator 150 via connection 135 to provide print instructions for a peripheral device, for example, a printer (not shown) located at the location of the client.

[0036] In accordance with the present invention, a server peripheral manager simulator component 160 is provided at the server 100, the server peripheral manager simulator component 160 interfacing with the print emulator 150 via connection 155. The operation of the server 100 within the system of the present invention will be described in more detail with reference to Figures 4 to 6 below.

[0037] The application virtualisation platform 110 includes software which virtualises the applications 120, 130, 140 or an operation system (not shown) on which each application runs. The server peripheral manager simulator component 160 simulates a peripheral manager (as will be described in more detail below) to enable local communications with the application emulator 150, in this case, the print emulator. A Common Use Passenger Processing System (CUPPS) or any other suitable interface type may be provided which enables communication between the print emulator and the server peripheral manager simulator component 160. (CUPPS is effectively the next generation of Common Use Terminal Equipment (CUTE) in accordance with IATA Recommended Practice 1797.)

[0038] In the embodiment of the server 100 shown in Figure 1, the application emulator 150 (or print emulator) is located within the application virtualisation platform 110. It may be connected to more than one application (not shown) if similar or identical applications are

running on the server 100 and which require the same type of application emulation.

[0039] In Figure 2, a modified embodiment of the server 100' is shown. Elements which have been described above with reference to Figure 1 have the same reference numerals and will not be described in detail again. In the embodiment of Figure 2, the application virtualisation platform 110' has been modified so that the print emulator 150' forms part of the application 130' itself and interfaces with the server peripheral manager simulator component 160 via connection 155'. In this case, a suitable interface is provided to enable communications between the print (or application) emulator 150' and the server peripheral manager simulator component 160.

[0040] In a further embodiment (not shown), the server peripheral manager simulator component may be embedded in the application itself.

[0041] In an airport implementation, the server 100 (or 100') is located remotely from the airport itself and is connected thereto by means of conventional communications links. A single server 100 (or 100') may be connected to more than one airport as well as to different areas, for example, check-in, boarding gates, baggage halls and other airport front and/or back offices.

[0042] Figure 3 illustrates a client 200 which is connectable to the server 100 (Figure 1) or server 100' (Figure 2) as will be described in more detail below with reference to Figures 4 to 6. The client 200 has a client virtualisation platform 210 which interfaces with a keyboard 220 and a display 230 as shown by respective links 225, 235. Although the keyboard 220 and display 230 are shown as separate elements, they may form part of a single module (not shown). The client 200 also includes a peripheral manager 240 which is connected for controlling peripheral devices 250, 260, 270, 280 via link 290. Typical peripheral devices which may be controlled by the peripheral manager 240 include, but are not limited to, printers, scanners, and readers. Whilst link 290 may be a physical link, for example, using a universal serial bus (USB) connection, the link 290 may also be a wireless link forming part of a local area network (LAN) or a wide area network (WAN). It is also possible that the link 290 operates using mobile telecommunications, for example, forming part of a 3G or 4G network or is connected via Bluetooth. (Bluetooth is a wireless technology standard managed by the Bluetooth Special Interest Group.)

[0043] In accordance with the present invention, a client peripheral manager simulator component 245 is provided which is associated with the client peripheral manager 240. The client peripheral manager simulator component 245 is used to form a virtual channel (not shown) with the server peripheral manager simulator component 160 (Figures 1 and 2) as will be described in more detail below.

[0044] In an airport environment, a plurality of clients 200 would be provided within the airport itself, for example, at check-in desks, at boarding gates, in baggage halls etc. Each client 200 connects to the remote server 100 (or 100') to access applications running on the application

virtualisation platform 110 (or 110') provided in the server 100 (or 100') as will be described in more detail below.

[0045] In Figure 4, a virtualisation system 300 in accordance with the present invention is shown. The system 300 comprises the server 100' as shown in Figure 2, but it will be appreciated that this is by way of example and the server 100 shown in Figure 1 could also be used. Elements of the server 100' as described above have the same reference numerals and will not be described in detail again here. The system 300 also includes the client 200 which is connected to the server 100' via a communications link 310. Communications link 310 may comprise a WAN or a LAN communications system which is capable of running a virtualisation protocol. As shown, the communications link 310 effectively connects the application virtualisation platform 110' to the client virtualisation platform 210. In particular, the communications link 310 comprises an internet protocol (IP) connection. In one embodiment, the virtualisation protocol utilises Citrix XenServer on the server and Citrix XenApp on each remote client computer to provide the respective virtualisation platforms. (Citrix, XenServer and XenApp are trademarks of Citrix Systems Inc. of Fort Lauderdale, Florida, USA.) However, it will be appreciated that other virtualisation protocols may be used, for example, VMWare, and Citrix applications are given by way of example. (VMWare is a trademark of VMWare Inc. of Palo Alto, California, USA.)

[0046] In addition, in accordance with the present invention, the communications link 310 effectively connects the server peripheral manager simulator component 160 with the client peripheral manager simulator component 245 forming part of the client peripheral manager 240 using a virtual channel 320 as indicated by dot-dash line as will be described in more detail below.

[0047] The client virtualisation platform 210 includes software which allows communications with the server 100 via the application virtualisation platform 110' and manages the delivery of the virtualised application or operating system to the peripheral manager 240 at the client 200. The peripheral manager 240 includes suitable device drivers (not shown) for each peripheral device 250, 260, 270, 280.

[0048] The client peripheral simulator component 245 enables communication between the client virtualisation platform 210 and the peripheral manager 240. As shown, the client peripheral manager simulator component 245 can be embedded within the peripheral manager 240. Alternatively, the client peripheral manager simulator component 245 may stand alone within the client 200 (not shown).

[0049] In accordance with the present invention, the provision of the server peripheral manager simulator component 160 and the client peripheral manager simulator component 245 enables the virtualisation of the system as will be described below with reference to Figure 6. Although not shown in Figure 4, it is to be noted that the virtual channel 320 extends through the communications link 310 via the application virtualisation platform 110' and the client virtualisation platform 210.

[0050] Figure 5 illustrates a system 400 in which a plurality of clients 200₁, 200₂, 200₃, 200₄, ..., 200_n are connected to a single server 100. Each client 200₁, 200₂, 200₃, 200₄, ..., 200_n has its own client virtualisation platform 210₁, 210₂, 210₃, 210₄, ..., 210_n and own client peripheral manager and client peripheral manager simulator component. For clarity, only the client peripheral manager 240₁ and client peripheral manager simulator component 245₁ for one client 200₁ is referenced, but it will be appreciated that the other clients also have client peripheral manager and client peripheral manager simulator components as shown.

[0051] The server 100 may be located at a central remote location and each client 200₁, 200₂, 200₃, 200₄, ..., 200_n may be located at any suitable location. For example, clients may be located at one airport or at a plurality of different airports all over the globe. As described above, the clients could also be located at locations, for example, at hotels or conference centres, where passengers can check-in prior to leaving for the airport.

[0052] As shown, a communications link 410 may be established between each of the clients 200₁, 200₂, 200₃, 200₄, ..., 200_n at the same time. A virtual channel 420 is also shown extending between the client peripheral manager simulator component 245₁ and the server peripheral manager simulator component 160. Although not shown in Figure 5, it is to be noted that the virtual channel 420 extends through the communications link 410 via the application virtualisation platform 110 and the client virtualisation platform 210.

[0053] Naturally, a virtual channel can also be established for the other client peripheral manager simulator components of each client 200₁, 200₂, 200₃, 200₄, ..., 200_n. Each client 200₁, 200₂, 200₃, 200₄, ..., 200_n may access the same application at the same time, or alternatively, more than one application may be accessed by the same or different clients. It will be appreciated that each client 200₁, 200₂, 200₃, 200₄, ..., 200_n accesses the server 100 via its own dedicated virtual channel within the communications link 410, that is, a plurality of virtual channels can be established between the server peripheral manager simulator 160 and each client 200₁, 200₂, 200₃, 200₄, ..., 200_n via the link 410.

[0054] By hosting all applications to be used by the clients on a common server, updates need only be applied once on the server without recourse to any of the clients. This reduces the complexity of keeping individual clients running the same application(s) up-to-date. In addition, as different applications can be hosted, in an airport environment, different airlines or other users of the system can use different applications but still use the same remote client computer to access their applications on the server.

[0055] In Figure 6, a flow chart 500 illustrating the basic steps of a method for operating the system is shown. Reference numerals relating to elements of the server 100' (Figure 2), the client 200 (Figure 3) and the system 300 (Figure 4) are also used in the following description. The flow chart 500 comprises four basic steps in addition to "start", step 510, and "end", step

560.

[0056] In step 510, a user logs onto the client using the keyboard 220 and the display 230 and initiates a session in which the client 200 is connected to the server 100' via communications link 310 (Figure 4). The connection between the client and the server initialises at least one graphic user interface (GUI) for one of more applications on the server 100' by calling the IP address of the server 100'. At the same time, the server peripheral simulator component 160 is also initialised.

[0057] In step 520, the connection 310 is made between the client virtualisation platform 210 and the application virtualisation platform 110' on the server 100. A connection is established for the session via the virtualisation protocol according to the platform used, for example, using Citrix as described above, provided on the two platforms 110', 210 in step 530. Support for virtual channels within this protocol enables communications with peripherals to be reliably established and which are unaffected by network address translation and blocking due to the presence of firewalls.

[0058] In step 540, the client localhost/Transmission Control Protocol (TCP) is virtualised on the client virtualisation platform 210 and made available, via the connection 310 established in step 530, to the application 130' running on the virtualised platform 110' on the server 100'. In step 550, the virtual channel 320 is established between the initialised server peripheral manager simulator component 160 and the client peripheral manager simulator component 245. The application 130' connects to the client localhost/TCP made available on the application virtualisation platform 110' in step 540, is virtualised and redirected via the virtual channel 320 back to the client peripheral manager 240 via the client peripheral manager simulator component 245.

[0059] Once the virtual channel 320 is established, all communication between the application 130' and the peripheral device with which it is to communicate is performed via the virtual channel 320, that is, all data is transferred over the virtual channel 320, either from the server 100' to the client 200 or from the client 200 to the server 100'.

[0060] The virtual channel 320 can be established regardless of the type(s) of communications network(s) operating within the communications link 310 and effectively provides a channel of direct communication between the application 130' running on the application virtualisation platform 110' and one of the peripheral devices 250, 260, 270, 280 connected to the client 200 via link 290. As described above, the application 130' may be a print application and the peripheral device connected to the client 200 may be a printer.

[0061] Naturally, other functionality may be provided by the peripheral devices, for example, scanning or reading. In this case, the scan or read is passed back from the relevant peripheral device 250,260, 270, 280 via link 290 to the server peripheral manager simulator component 160, via the associated client peripheral manager 240, the client peripheral manager simulator component 245 and the virtual channel 320. From the server peripheral manager simulator

component 160, the scan or read is passed to an associated application within the application virtualisation platform 110'. The information relating to the scan or read may then be stored in a database or memory (not shown) if required.

[0062] In step 560, the session ends when the connection between the server 100' and the client 200 is terminated when the user logs off the client 200.

[0063] As described above, the server peripheral manager simulator component is launched each time the virtualised application is launched, that is, every time a client connects to the server. However, it is also possible to have a single server peripheral manager simulator component which provides a connection port for each instance of the virtualised application being launched on the server in response to the connection of a client with the server. The latter option is not preferred as all connections between clients and the server will be lost if the server peripheral manager simulator needs to be restarted for whatever reason.

[0064] A peripheral emulator is described above which is associated with the application on the server which needs to connect to a peripheral device on the client. However, this is not essential and its functionality can be included in the server peripheral manager simulator component.

[0065] Although the present invention has been described with reference to an airport environment in which airport-related applications are run with each client corresponding to a check-in terminal, for example, connected to a single server, it will be appreciated that the present invention can also be used in non-airport environments irrespective of worldwide location, for example, cruise liner terminals, ferry terminals, hotels, resorts and conference centres, that is, any application where multiple clients need to establish a connection with a server to access the same application over a communications link having multiple addressing schemes and translation across heterogeneous local and wide area networks, including mobile carrier networks, to enable communications between applications running on the server and peripheral devices associated with the clients.

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US8032353B [0004] [0005]
- US2009043921A [0007]
- WO2011022375A1 [0008]
- US2009164990A1 [0009]

KOMMUNIKATIONSSTYRING**PATENTKRAV**

1. Klientsystem til anvendelse i en lufthavn, hvilket klientsystem omfatter:
 - en terminal (200), der omfatter en klientvirtualiseringsplatform (210), idet klientvirtualiseringsplatformen (210) kan forbindes med en server (100) via en første kommunikationsforbindelse (310), et tastatur (220) og en skærm (230);
 - en periferi-manager (240), som kan forbindes med en ekstern enhed (250; 260; 270; 280); og
 - en klient-periferi-manager-simulator (245), der kommunikativt er koblet til periferi-manageren (240); hvor:
 - klientvirtualiseringsplatformen (210) er konfigureret til at:
 - initiere en session mellem serveren (100) og terminalen (200) via den første kommunikationsforbindelse (310);
 - få adgang til en applikation på serveren (100); og
 - etablere en virtuel kanal (320) mellem klient-periferi-manager-simulatoren (245) og serveren (100) via klientvirtualiseringsplatformen (210); periferi-manageren (240) er konfigureret til at:
 - drive den eksterne enhed (250; 260; 270; 280) via en anden kommunikationsforbindelse (290); og klient-periferi-manager-simulatoren (245) er konfigureret til at:
 - kommunikere alle data mellem periferi-manageren (240) og applikationen på serveren (100) via den virtuelle kanal (320).
2. Klientsystem ifølge krav 1, hvor klientvirtualiseringsplatformen (210) er konfigureret til at initiere sessionen, når en bruger logger på terminalen (200).
3. Klientsystem ifølge krav 2, hvor klientvirtualiseringsplatformen (210) er konfigureret til at afslutte sessionen, når en bruger logger af terminalen (200).
4. Klientsystem ifølge et hvilket som helst foregående krav, hvor periferi-manageren (240) omfatter en enhedsdriver til den eksterne enhed (250; 260; 270; 280).
5. Klientsystem ifølge et hvilket som helst foregående krav, hvor klientvirtualiseringsplatformen (210) yderligere er konfigureret til at vise en grafisk

grænseflade til en applikation, der køres på serveren (100).

6. Klientsystem ifølge et hvilket som helst foregående krav, hvor den anden kommunikationsforbindelse (290) er én af: en fysisk forbindelse, en trådløs forbindelse eller en forbindelse, som fungerer via mobilkommunikation.

7. Klientsystem ifølge et hvilket som helst foregående krav, hvor den eksterne enhed (250; 260; 270; 280) er én af: en printer og en scanner.

8. Fremgangsmåde til drift et klientsystem i en lufthavn, hvilket klientsystem omfatter:
en terminal (200), der omfatter en klientvirtualiseringsplatform (210), som kan forbindes med: en server (100) via en første kommunikationsforbindelse (310), et tastatur (220) og en skærm (230); en periferi-manager (240), som kan forbindes med en ekstern enhed (250; 260; 270; 280); og en klient-periferi-manager-simulator (245), som kommunikativt er forbundet med periferi-manageren (240); hvilken fremgangsmåde omfatter:

initiering, ved hjælp af klientvirtualiseringsplatformen (210), af en session mellem serveren (100) og terminalen via en første kommunikationsforbindelse (310);

adgang, ved hjælp af klientvirtualiseringsplatformen (210), til en applikation på serveren (100);

etablering, ved hjælp af klientvirtualiseringsplatformen (210), af en virtuel kanal (320) mellem klient-periferi-manager-simulatoren (245) og serveren (100) via klientvirtualiseringsplatformen (210);

drift, ved hjælp af periferi-manageren (240), af den eksterne enhed (250; 260; 270; 280) via en anden kommunikationsforbindelse (290); og

kommunikation, ved hjælp af klient-periferi-manager-simulatoren (245), af alle data mellem periferi-manageren (240) og applikationen på serveren (100) via den virtuelle kanal (320).

9. Fremgangsmåde ifølge krav 8, der endvidere omfatter:

tilvejebringelse, via en bruger, af login-detajler for terminalen (200) inden initiering af sessionen mellem serveren (100) og terminalen (200).

10. Fremgangsmåde ifølge krav 9, der endvidere omfatter:

afslutning, ved hjælp af klientvirtualiseringsplatformen (210), af sessionen, når brugeren logger af terminalen (200).

11. Fremgangsmåde ifølge et hvilket som helst af kravene 8 til 10, hvor periferi-manageren (240) omfatter en enhedsdriver til den eksterne enhed (250; 260; 270; 280).
12. Fremgangsmåde ifølge et hvilket som helst af kravene 8 til 11, hvor initiering af en session omfatter visning af en grafisk grænseflade til en applikation, der køres på en server (100).
13. Fremgangsmåde ifølge et hvilket som helst af kravene 8 til 12, hvor den anden kommunikationsforbindelse (290) er én af: en fysisk forbindelse, en trådløs forbindelse eller en forbindelse, som fungerer via mobilkommunikation.
14. Fremgangsmåde ifølge et hvilket som helst af kravene 8 til 13, hvor den eksterne enhed (250; 260; 270; 280) er én af: en printer og en scanner.
15. Computerlæsbart medium, der lagrer computereksekverbare instruktioner, som ved afvikling ved hjælp af en processor får processoren til at udføre fremgangsmåden ifølge krav 8.

DRAWINGS

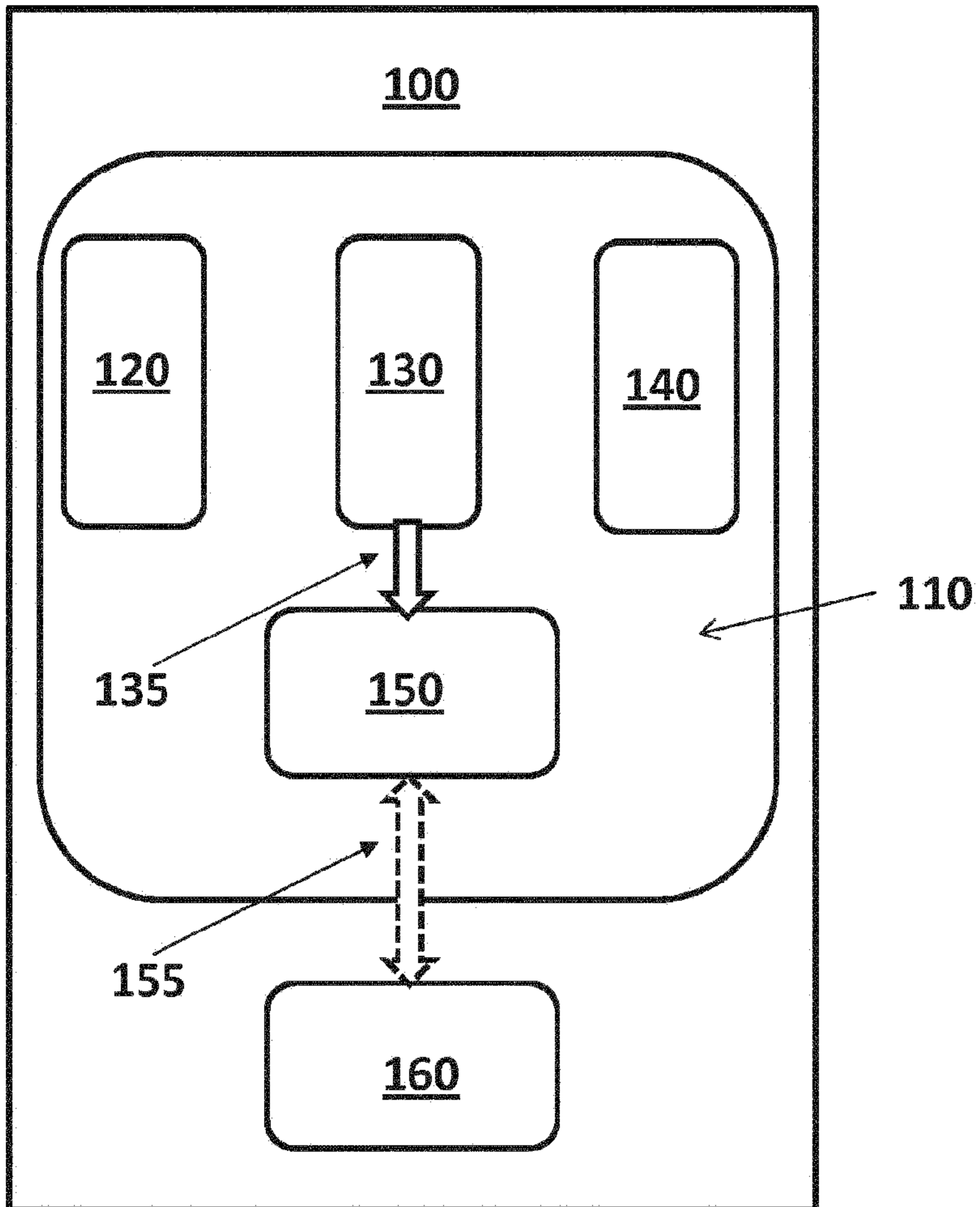


Fig. 1

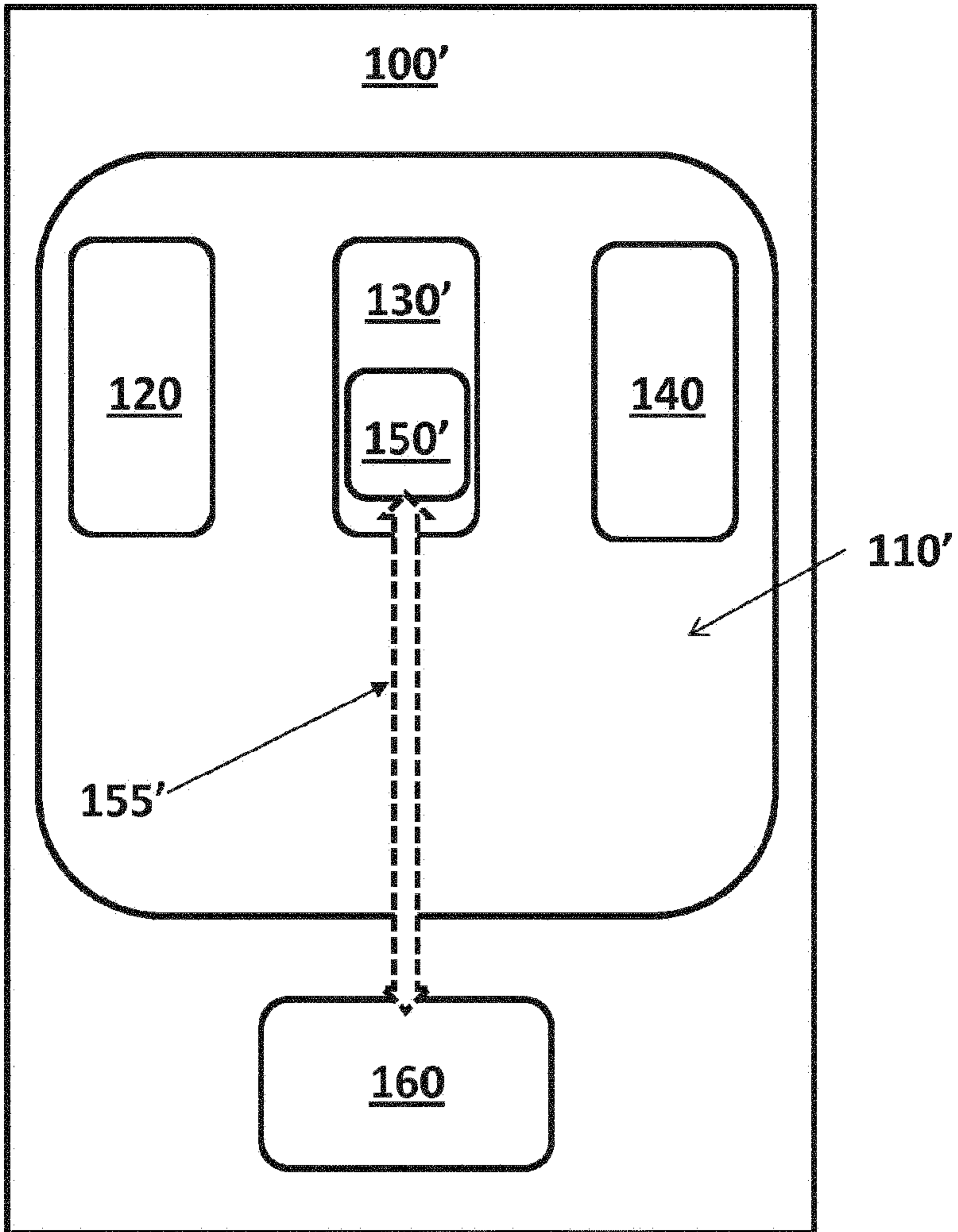


Fig. 2

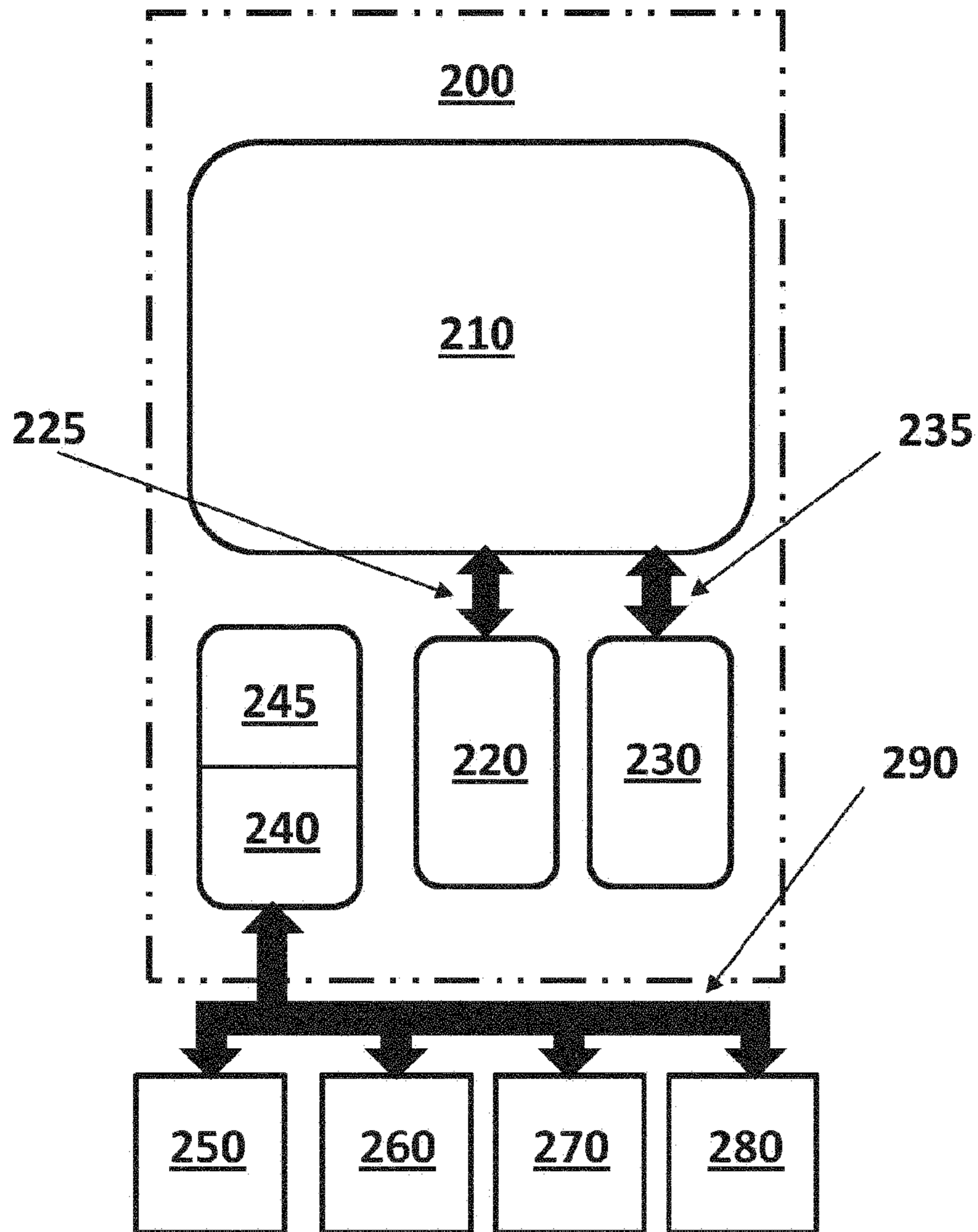


Fig. 3

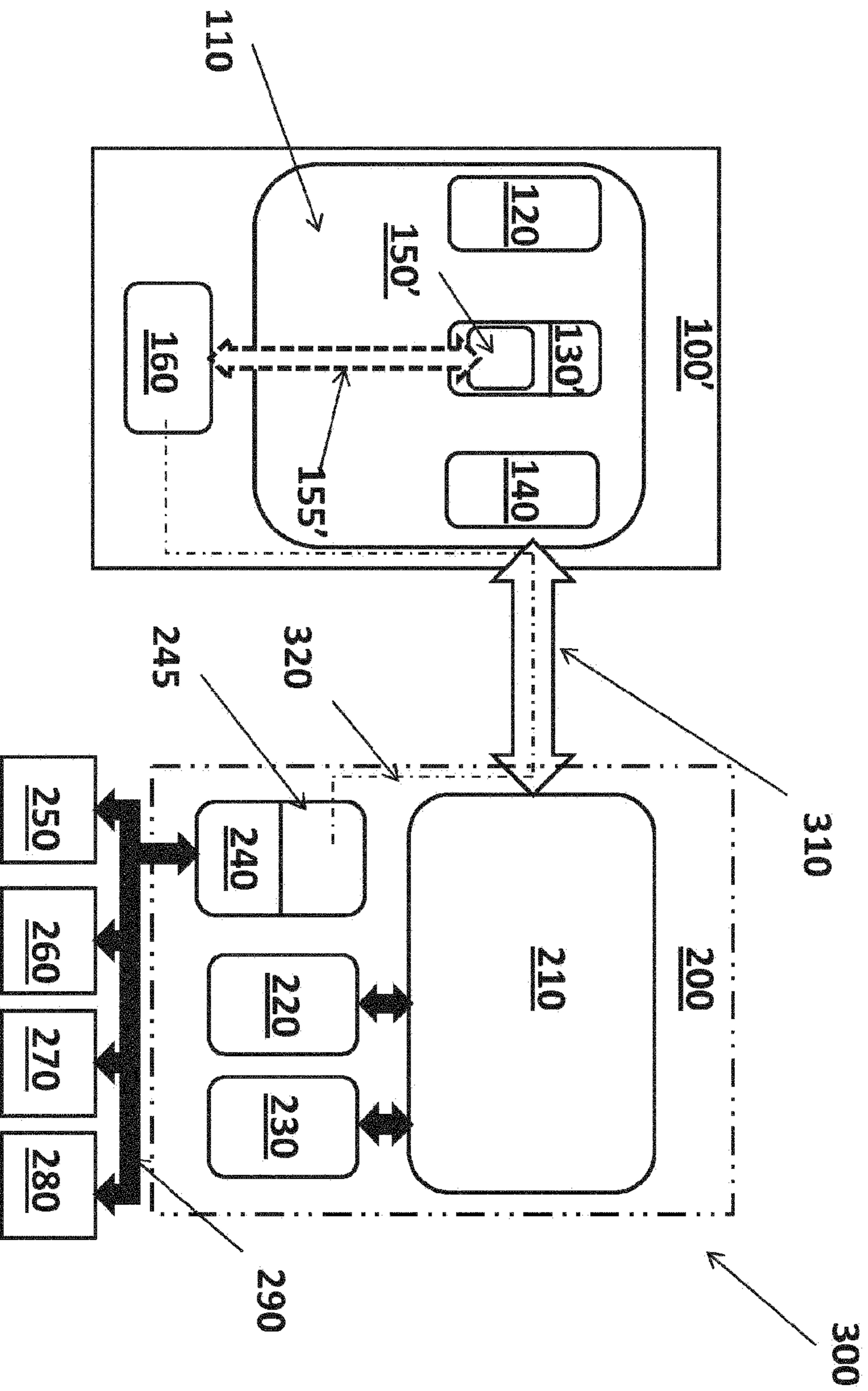


Fig. 4

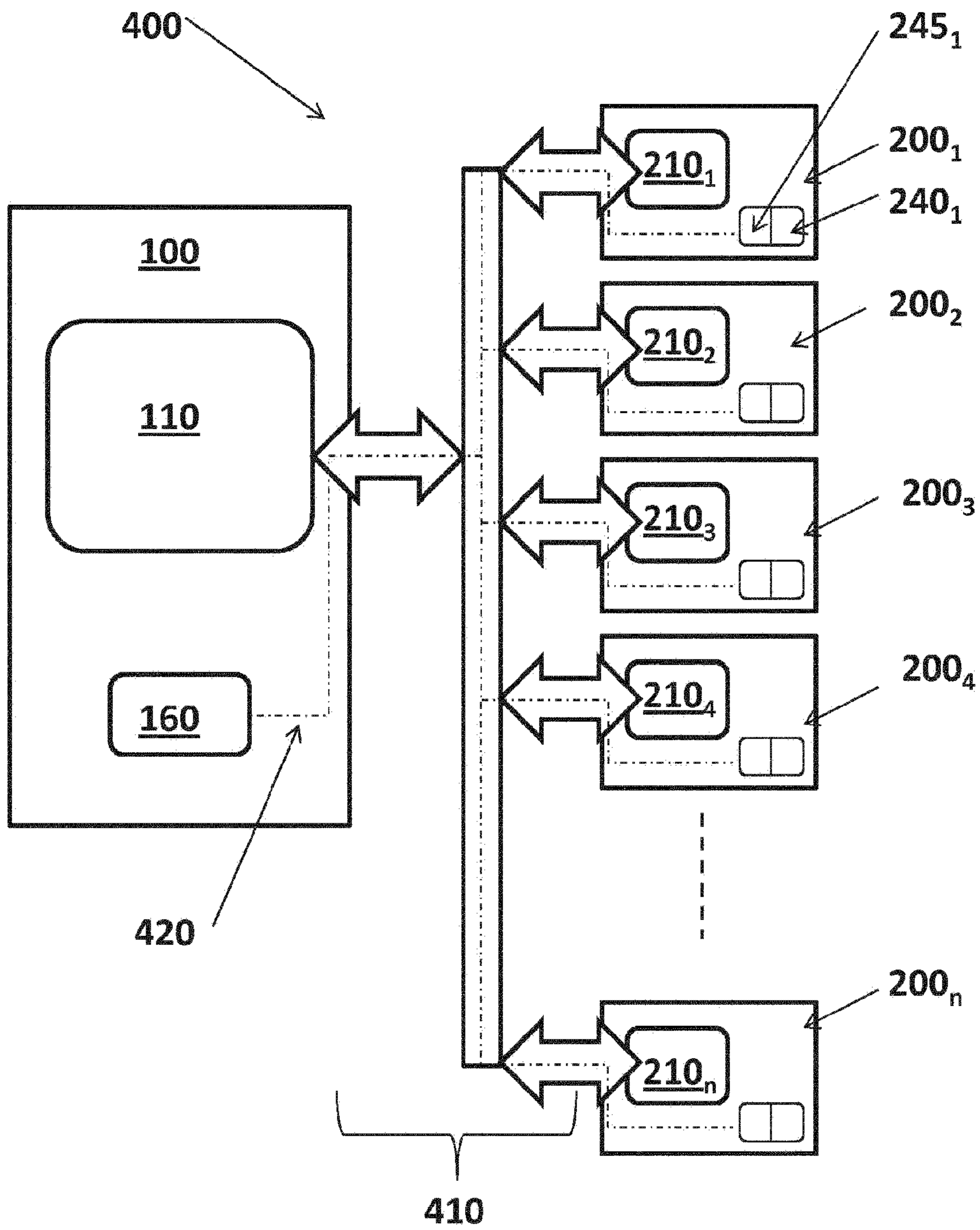


Fig. 5

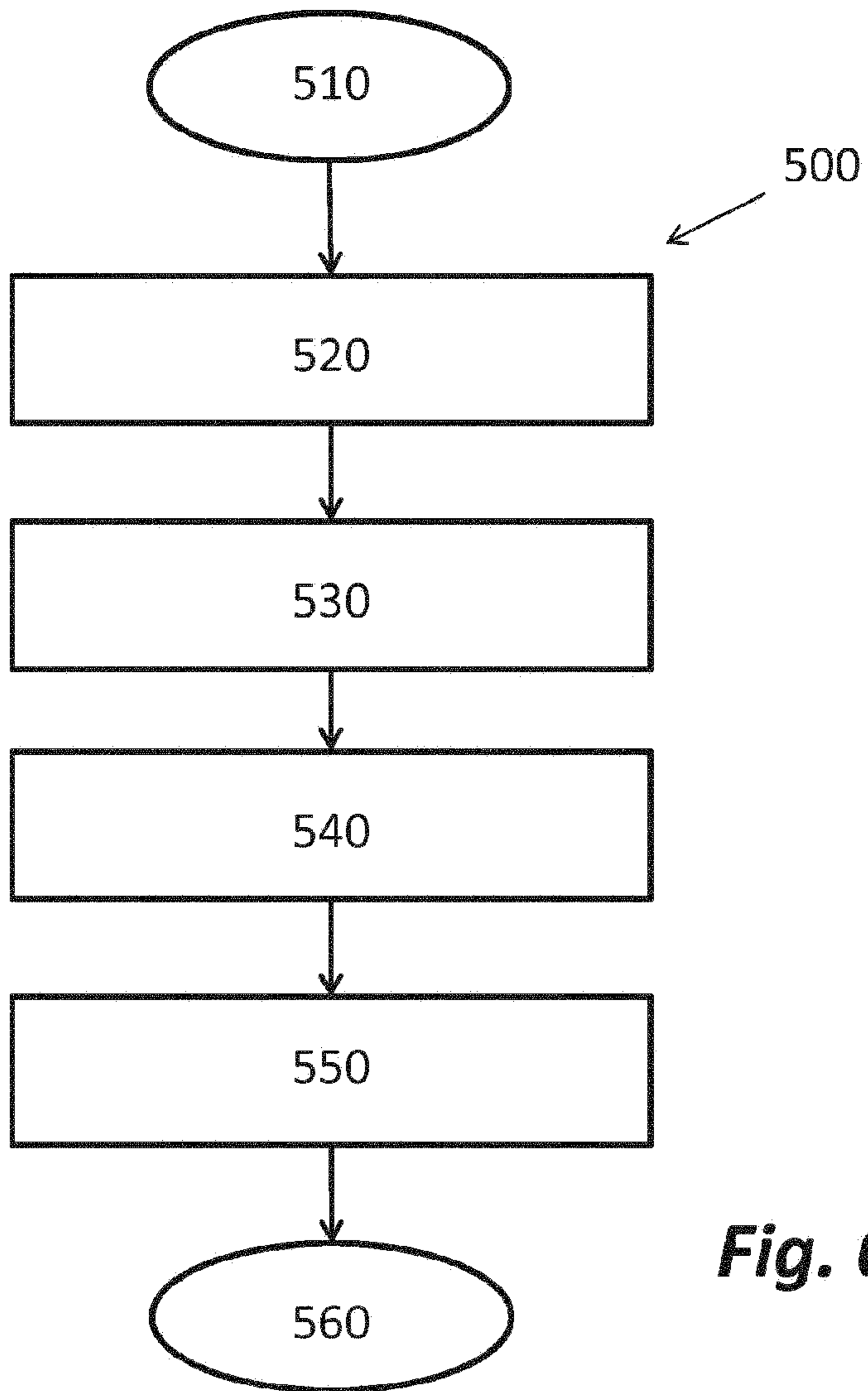


Fig. 6