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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 May 2010 (27.05.2010)

PCT

(10) International Publication Number
WO 2010/057526 A1

(51) International Patent Classification:
G06F 9/455 (2006.01) *H04L 29/08* (2006.01)
G06F 9/445 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/EP2008/065837

(22) International Filing Date:
19 November 2008 (19.11.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NEC EUROPE LTD.** [DE/DE]; Rathausallee 10, 53757 Sankt Augustin (DE).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **KOHRING, Gregory, Allen** [US/DE]; c/o NEC Laboratories Europe, IT Research Division, Rathausallee 10, 53757 Sankt Augustin (DE).

(74) Agent: **VOSSIUS & PARTNER**; Siebertstrasse 4, 81675 München (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI

[Continued on next page]

(54) Title: VIRTUALIZATION MEDIATED SYSTEM AND METHOD

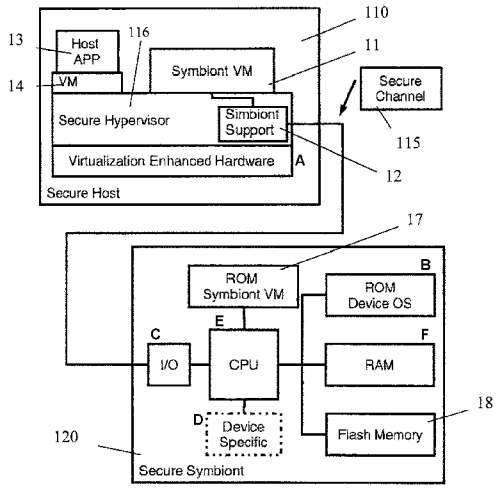

Fig. 3

13 Application hôte
14 Machine virtuelle
11 Machine virtuelle symbiote
116 Hyperviseur sécurisé
12 Support symbiote
A Matériel amélioré par virtualisation
110 Hôte sécurisé
115 Canal sécurisé
17 Mémoire morte ; machine virtuelle symbiote
B Mémoire morte ; Système d'exploitation de dispositif
C E/S
D Spécifique de dispositif
18 Mémoire flash
120 Symbiote sécurisé
E Unité centrale
F Mémoire vive

(57) Abstract: The present invention relates an electronic device (20, 120) comprising a virtual appliance stored on a memory (17) thereof, in particular, device software for driving components of the electronic device (20, 120) are encapsulated in the virtual machine being initiated from the virtual appliance. The present invention also relates to a method for providing connectivity between the electronic device (20, 120) with a virtual appliance and a host (10, 110) being capable to manage a virtual machine by using a hypervisor (16), wherein the method comprises steps of a) establishing a connection between the electronic device (20) and the host (10); b) providing an access for the host (10) to the electronic device (20); c) transferring the virtual appliance from the electronic device to the host and instantiating the virtual machine from the virtual appliance; d) launching an application (13) on the host (10) and initiating a communication with the virtual machine by the application (13) via a connection interface provided by the hypervisor of the host; e) shutting down the virtual machine and terminating the connection between electronic device and the host when the communication is no longer required.

WO 2010/057526 A1

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

VIRTUALIZATION MEDIATED SYSTEM AND METHOD

The present invention relates to an electronic device comprising a virtual appliance stored in a memory thereof. In particular, a device software for driving components of the electronic device is provided on the electronic device, preferably encapsulated in the virtual appliance. The virtual appliance being adapted to initiate a virtual machine on a host, such that the device software is preferably encapsulated in said virtual machine. The present invention also relates to a method and a system for providing connectivity between the electronic device and a host, in particular, for providing connectivity between an electronic device and a host by using virtualization technology, more particular, for providing secure connectivity between an electronic device and a host by using virtualization technology.

BACKGROUND OF THE INVENTION

When dynamically attaching a device to a computer problems can arise if the computer does not have the necessary device software to operate the device. Usually this problem is solved by installing the necessary device software (also called device driver or software driver) on the computer using a CD, DVD or some other medium shipped with the device, or downloading the device software from the internet. This conventional method of device software handling, that is, the decoupling of the software from the device provides a limited number of supported platforms, i.e., operating systems and hardware combinations. Thus, the choice of a platform to those supported by the device manufacturer is restricted. If the device manufacture does not support the device any longer, the device may become inoperable when the platform is updated to a newer major version. Furthermore, since the device software can stored separated from the device, in the event the device software is lost, it is impossible to use the device.

For example, ISDN PBX (Private Branch Exchange) devices are often used by consumers to manage multiple telephone numbers on their home lines. The PBX device itself is self contained and requires no interaction with a computer, except for configuration purposes. To configure the PBX, special configuration software on the host computer needs to be installed at first, then a connection, e.g., through a USB Interface, is made to a computer, and the previously installed software is used to set the various parameters. The problem with this scenario is that the PBX may run fine for years without needing any user intervention. In the meantime the user's computer has been upgraded and the old configuration software no longer runs on the new platform. If the PBX vendor is not willing to supply software for the new platform, then the user has few options other than retaining a copy of the old platform, i.e., hardware and software, simply for the purpose of configuring the PBX, or buying a new PBX. If the PBX vendor is willing to supply software for the new platform, they may send the device software, i.e., device drivers on a CD to end user, or provide the device software on their web page for downloading. The problem with this approach is that the PBX vendor need to support many different operating systems, or the end users is limited on the operating systems supported by the vendor.

The device software handling according to the conventional approach may be summarized as following steps: plugging a device into a computer; searching in a database of device drivers on the computer and attempting to locate an appropriate driver for the device; activating and communicating the device if the appropriate driver is found; or prompting a request for manual intervention such as inserting a CD containing the appropriate driver or downloading the appropriate driver if no appropriate driver is found.

A previous attempt to solve this problem is disclosed in US 6,754,725, which discloses to store device drivers with the device and to transfer them to a host machine when the device is connected. While this does resolve the problem of providing a tight coupling between the drivers and the device, it is a method, which is dependent on the operating system for which the device drivers are provided.

Using Java for the device drivers gains hardware independence, but there is still a residual coupling between the Java VM on the host machine and the byte code version used to compile the device drivers.

It is therefore an object of the present invention to provide an improved electronic device and an improved method and system for a simplified and preferably secure connectivity between the electronic device and a host. In particular, it is a further object of the present invention to provide a method and an electronic device by which a device driver is preferably provided within the device and independent from the platform and/or operating system of the host.

These and other objects are achieved by the features of the independent claims. Further preferred embodiments are characterized in the dependent claims.

SUMMARY OF THE INVENTION

The present invention overcomes the above discussed problems of the prior art solutions and preferably provides a fully platform and/or operating system independent solution. In particular, the electronic device according to a first aspect of the present invention is preferably provided with a small virtual machine (also virtual appliance in its uninitialized state) which can be dynamically loaded by the host when the device and host are dynamically bound, and can be dynamically shutdown when the connection is broken. The device driver may be provided within the virtual machine, i.e. it is sufficient to run an application platform independent from the host of the virtual machine. That is, not only the hardware platform is independent but also the software platform, e.g., the favorite operating system without regard to the demands of the device manufacture, may be also freely selected, changed or upgraded as long as the operating system supports the virtual machine provided by the manufacture of the electronic device. That is, the present invention maintains a tight coupling between the device and the device software required to service the device from a host. As using virtualization technology the dependency between the host operating system and the device software are decoupled, the device may be used in multiple environments. Furthermore, this decoupling may enhance the security

of both the electronic device and the host, hence the present invention may also be used in security critical situations.

In other words, according to the present invention, an application for virtualization technology in combination with high speed interconnections and electronic devices is explored. The basic idea here is to equip electronic devices with their own virtual machine which can be read and instantiated by any virtualization enhanced device.

When two such devices come together, the virtual machine image, i.e., the virtual appliance of the first device (electronic device) is dynamically transferred to the second device (the host), giving rise to a symbiotic system with mutually beneficial results for both devices. In view of the relationship of the symbiotic system, the second device is the host capable for hosting a virtual machine, whereas the first device (electronic device) is the guest or symbiont whose operating system runs on top of operating system of the host. The hardware environment for running the operating system of the electronic device may be virtualized by the virtualizations software of the host.

Virtualization technologies are changing the landscape of the enterprise server environment and have also made inroads into the arena of desktop computing. In a virtualization enhanced environment, hardware and software technologies enables creating a virtual machine (VM), within which an application can run without affecting other applications running on the same server or running in other virtual machine on the same server. There are many reasons for wanting such a separation of applications particularly in a server environment where running applications in their own virtual machine allows the operators to make security and performance guarantees even when other applications running in other virtual machines on the same host. In a traditional time sharing environment with all applications running on a single physical machine, such guarantees are difficult to keep. In a desktop environment virtualization also allows simultaneously running applications written for different operating systems, thus allowing choosing applications based upon their quality and suitability for the purpose and not upon their compatibility with a particular operating system. Furthermore, with the introduction of hardware support for the context switching required in a virtualized environment, many of

the performance barriers which previously prevented why spread acceptance of this technology are disappearing.

The method for providing device software according to the present invention can have large consequences as a major part of an operating system kernel is taken up by device drivers. If these can all be moved out to their own virtual machine, then the kernel sizes can shrink by several million lines of code, leading to improved stability and performance of the operating system kernel.

In order to overcome the drawbacks in the prior art, assuming a scenario in which the PBX had a virtual machine containing its device drivers and this VM was automatically instantiated on a virtualization enhanced computer when the PBX is connected. The system upgrades on the host may be carried out without consideration of the requirements of the PBX's configuration software. If all the devices in the home or office were similarly enabled, a favorite platform may be selected without regard to extraneous compatibility issues.

According to an aspect, the present invention provides a method for providing connectivity between an electronic device with a virtual appliance and a host being capable to manage a virtual machine by using a hypervisor. The method comprises steps of: a) establishing a connection between the electronic device and the host; b) providing an access for the host to the electronic device; c) transferring the virtual appliance from the electronic device to the host and instantiating the virtual machine from the virtual appliance; d) launching an application on the host and initiating a communication with the virtual machine by the application via a connection interface provided by the hypervisor of the host; e) shutting down the virtual machine and terminating the connection between electronic device and the host when the communication is no longer required.

According to another aspect, the subject of the present invention is an electronic device comprising a virtual appliance stored on a memory of the electronic device. The stored appliance being capable to initiate a virtual machine on the host, wherein a device

software (device driver) for driving components of the electronic device is encapsulated in the virtual machine.

The virtual appliance may be provided in an image format as a file on the electronic device. After the virtual appliance is transferred into the host, the virtualization software, e.g. the hyper visor on the host may lunch the appliance and initiate a virtual machine.

According to another aspect of the present invention, the electronic device may comprise components such as a CPU, an I/O interface, a memory, wherein said memory is preferably a ROM. In a preferred embodiment, said memory may also comprise a RAM and/or a flash memory.

According to another aspect of the present invention, the electronic device may further comprise a first network interface and the host further comprises a second network interface.

According to another aspect of the present invention, the virtual appliance may be provided on the memory of electronic device.

According to another aspect of the present invention, a device software for driving the components of the electronic device may be encapsulated in the virtual appliance. The device software may also stored in a separate memory as the virtual appliance located in.

According to another aspect of the present invention, the device software may be provided for a particular operating system which may be freely predetermined for hosting the virtual machine. If the device manufacture supports a plurality of operating systems, multiple device software each for driving a particular operating system may also provided.

According to another aspect of the present invention, the access for the host to the electronic device may be an exclusive access.

According to another aspect of the present invention, the host may be capable to manage multiple virtual machines and the host may comprise at least one preconfigured virtual machine with at least one application.

According to another aspect of the present invention, the connection between the electronic device and the host may be established over a wired interface such as USB, the first and the second network interface, or a wireless interface such as WLAN, Bluetooth.

According to another aspect of the present invention, the host may be operated with a virtual environment software comprising the hypervisor.

According to another aspect of the present invention, the hypervisor may provide a virtual machine support which is connected with the I/O interface of the electronic device.

According to another aspect of the present invention, after the step a) the method may further comprise a step of: a1) determining if the electronic device includes a virtual appliance.

According to another aspect of the present invention, the device software may be provided by a device driver database of the host if the electronic device does not include a virtual appliance.

According to another aspect of the present invention, after the step a1) the method may further comprise a step of: a2) terminating the connection between the electronic device and the host if or when the host is not capable to manage the virtual machine.

According to another aspect of the present invention, after the step c) the method may further comprise a step of: c1) launching and interacting the virtual machine by using the standard input and output device.

According to another aspect of the present invention, the host may be a secure host and the hypervisor may be secure hypervisor, and the electronic device may be a secure electronic

device and the connection between the electronic device and the host may be established through a secure channel.

According to another aspect of the present invention, the I/O interface of the electronic device and the virtual machine support may be connected through a secure connection channel, or the first and the second network interface and the virtual machine support may be connected through a secure connection channel.

According to another aspect of the present invention, an authentication between the electronic device and the secure host may be carried out before the establishment of the connection therebetween.

According to another aspect, the subject of the present invention is a system for providing connectivity between the electronic device according to any above mentioned aspects and a host being capable to manage a virtual machine by using a hypervisor.

The present invention allows a simplified procedure for providing the device software as follows: plugging the device into the host; a hypervisor running on the host loads the virtual appliance including the device software from the device; the hypervisor instantiates the virtual machine; the hypervisor maps the virtual machine to its list of devices and notifies any other virtual machines that a new device is available. In this sequence there is no need for the operating system to keep a large database of device drivers, most of which are never needed. There is also no need for the device manufacturer to create and maintain CDs containing the device drivers. The conventional approach, that is, the conventional plug-and-play sequence may be replaced by a virtual machine based plug-and-play sequence.

Furthermore, other devices may also benefit from the present invention, e.g., household appliances, furnaces, smart cards, monitors, sensors, etc. Basically, any device which does not require a tight coupling to other applications on the host could benefit from the present invention.

Moreover, device manufactures may also gain as they do not have support multiple combinations of hardware and operating systems, but can rely on the virtualization technology to overcome any compatibility problems. This frees up resources for the software developers to concentrate on producing high quality software and not on getting bogged down in implementation details. Furthermore, as the device software servicing the devices runs in its own virtual machine, there is a reduced risk of a manufacture's software having undesired interactions with other software running in the same real machine.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in detail with respect to preferred embodiments with reference to accompanying drawings, wherein:

Fig. 1 shows a schematic view of the connectivity between the electronic device and the host according to a preferred embodiment of the present invention;

Fig. 2 shows a flow chart of the connectivity established between the electronic device and the host according to a preferred embodiment of the present invention;

Fig. 3 shows a schematic view of the connectivity between the electronic device and the host according to a preferred embodiment of the present invention;

Fig. 4 shows a flow chart of the connectivity established between the electronic device and the host according to a preferred embodiment of the present invention; and

Fig. 5 shows a further schematic view of the connectivity between the electronic device and the host according to a preferred embodiment of the present invention, similar to the embodiment of Fig. 1.

DETAILED DESCRIPTION OF THE INVENTION

Reference is now made in detail to the preferred embodiment of the present invention, examples of which are illustrated in the accompanying drawings.

Fig. 1 shows an exemplary architecture for a dynamic symbiotic system according to a preferred embodiment of the present invention. The new components shown here are the symbiont virtual machine 11, i.e., symbiont VM 11 and the symbiont support software 12 in the hypervisor 16. The symbiont support software 12 is a generic device driver which is not dependent upon the details of the symbiont itself and can therefore be re-used for interacting with other symbionts. A simplified description of the sequence of events leading up to such a symbiosis is as follows:

- The symbiont, i.e. the electronic device 20 and the host 10 establish a mutual contact. This might be over a wired or wireless interface 15, e.g., USB, Ethernet, Bluetooth, WLAN, etc.
- The host 10 retrieves the symbiont virtual machine image, i.e., virtual appliance from the symbiont 20 and instantiates it. For instance, the virtual appliance may be stored in ROM 17. At this point the hypervisor 16 may lock the device node providing access to the symbiont 20 in order to ensure no other virtual machines are able to access the symbiont device 20.
- Users interact with the symbiont VM 11 using standard input devices.
- Other host applications 13 running an other virtual machines 14 may interact with the symbiont VM 11 using an interface, e.g. a network interface provided by the hypervisor 16 as illustrated in the Fig. 1. Alternatively, the application may also directly run on the host and communicate with the virtual machine 11 as shown in Fig. 5.
- When the symbiont 20 is no longer needed, its virtual machine 11 is shutdown and the connection between host 10 and symbiont 20 is closed.

Therefore, according to the present invention, the approach of providing device driver for component is improved significantly, in terms of platform independence and simplicity, in view of the conventional plug and play approach.

Fig. 2 shows a simplified workflow diagram delineating the exemplary steps of the method according to the present invention for providing connectivity between the electronic device and the host. The step of terminating the connection when the connection is no longer required, is not shown in the workflow. The steps of the method may summarized as follows:

- S1: The symbiont (device 20) and host establish a mutual contact;
- S2: If the device is not a symbiont the normal device activation procedure may be carried out in step S3, or if the device 20 is a symbiont according to the present invention the host 10 queries symbiont for VM requirements in step S4;
- S5: If the host is not capable for hosting symbiont, e.g. wrong version and/or lack of hypervisor provided within the virtualization software, the connection between the host and the device 20 may be terminated in step S6, or if host is capable for hosting symbiont the transfer of symbiont VM, e.g. virtual appliance may be transferred from the device 20 to the host 10 in step S7;
- S8: After the transfer of the virtual appliance is completed, the virtual machine (symbiont VM) may be initiated.

The symbiotic system mentioned above is fine for non-critical environments. However, when security is an issue, special precautions need to be taken.

Fig. 3 shows an exemplary model for a secure symbiotic system according to a preferred embodiment of the present invention. The major difference here, is that the symbiont 120 and host 110 first may establish a trusted relationship and a secure communication channel 115 using established technologies, e.g., developed by the Trusted Computing Group. In particular, for a virtualization enhanced host 110, a secure hypervisor 116 may be required and all other applications may run within one or more virtual machines started by the secure hypervisor, i.e., the hypervisor should not allow applications to simply run a top of the hypervisor 116. The hypervisor 116 should only allow the different virtual machines to communicate through the virtualized network interface. Drag and drop, or other such types of GUI operations have to be prohibited in order to prevent other virtual machines from infecting the symbiont 20 virtual machine 11.

Fig. 4 shows a simplified workflow illustrating how a secure symbiont 120 interacts with a secure host 110. It shows the exemplary steps of the method according to the present invention for providing a secure connectivity between the electronic device 120 (secure symbiont) and the host 110 (secure host). The additional steps for authentication of the connection may be described as follows:

- S5: If the host 110 is capable for hosting the symbiont VM 11 the transfer of symbiont VM, the mutual authentication of the symbiont 120 and the host 110 may be started in step S5a;
- S5b: The symbiont 120 and the host 110 verify the software of the other party;
- S5c After the verification, a secure communication channel 115 may be established.

Applications that could benefit from a secure dynamic symbiont system include smart cards. For example, if a smart card is used for the purposes of digitally signing documents, it may be desired to ensure, that what is being signed is actually what is should be signed. The smart card acting as the secure symbiont may contain a virtual machine capable of displaying the document to be signed. When plugged into a host, they may mutually authenticate and verify the software of the other party, and the secure symbiont's virtual machine may be loaded and run. Such a scenario would not require the host system to install any software other than that necessary to support a secure virtual environment.

A secure symbiont used in the area of electronic health records may allow authorized health professionals to examine a patient's electronic health card without any special software. As further advantage over the standard smart cards, the secure symbiont according to the present invention may prevent the health professionals from making local copies of the information, thus a spread of sensitive patient data may be controlled.

The present invention has now been described with reference to several embodiments thereof. The foregoing detailed description and examples have been given for clarity of understanding only. No unnecessary limitations are to be understood therefore. It will be apparent to those skilled in the art that many changes can be made in the embodiments

described without departing from scope of the present invention. In particular, although features and elements of the present invention are described in the preferred embodiments in particular combinations, each feature or element can be used alone without the other features and elements of the preferred embodiments or in various combinations with or without other features and elements of the invention. Therefore, the scope of the present invention should not be limited to the apparatuses, methods and systems described herein.

Claims

1. A method for providing connectivity between an electronic device (20, 120) with a virtual appliance and a host (10, 110), said host being capable to manage a virtual machine by using a hypervisor (16, 116), wherein the method comprises steps of:

a) establishing a connection between the electronic device (20, 120) and the host (10, 110);

b) providing an access for the host (10, 110) to the electronic device (20, 120);

c) transferring the virtual appliance from the electronic device to the host and instantiating the virtual machine from the virtual appliance;

d) launching an application (13) on the host (10) and initiating a communication with the virtual machine (11) by the application (13) via a connection interface provided by the hypervisor (16, 116) of the host;

e) shutting down the virtual machine (11) and terminating the connection between electronic device (20, 120) and the host when the communication is no longer required.

2. The method according to claim 1, wherein the electronic device comprises a memory (17, 18), wherein said memory is preferably a ROM, and wherein the virtual appliance is provided on said memory (17, 18).

3. The method according to claim 2, wherein a device software for driving the electronic device is stored in the memory (17, 18) and preferably encapsulated in the virtual appliance.

4. The method according to claim3, wherein the device software is provided for a particular operating system which is predetermined for the virtual machine.

5. The method according to any one of the preceding claims, wherein the access for the host (10, 110) to the electronic device (20, 120) is an exclusive access.

6. The method according to any one of the preceding claims, wherein the host (10, 110) is capable to manage multiple virtual machines (11, 14) and the host comprises at least one preconfigured virtual machine (14) with at least one application (13).

7. The method according to any one of the preceding claims, wherein the connection between the electronic device (20, 120) and the host (10, 110) is being established over a wired interface such as USB, a network interface, such as an Ethernet Network Interface, or a wireless interface such as WLAN or Bluetooth.

8. The method according to any one of the preceding claims, wherein the host (10, 110) being operated with a virtual environment software comprising the hypervisor (16).

9. The method according to any one of the preceding claims, wherein the hypervisor (16) provides a virtual machine support (12) which is connected with the electronic device (20, 120).

10. The method according to any one of the preceding claims, wherein after the step a) the method further comprises a step of:

a1) determining if the electronic device (20, 120) includes a virtual appliance.

11. The method according to claim 10, wherein the device software is provided by a device driver database of the host if the electronic device (20) does not includes a virtual appliance.

12. The method according to claim 10, wherein after the step a1) the method further comprises a step of:

a2) terminating the connection between the electronic device (20) and the host if the host is not capable to manage the virtual machine.

13. The method according to any one of the preceding claims, wherein after the step c) the method further comprises a step of:

c1) launching and interacting the virtual machine by using the standard input and output device.

14. The method according to any one of the preceding claims, wherein the host is a secure host (110) and the hypervisor is a secure hypervisor (116), and wherein the electronic device is a secure electronic device (120) and the connection between the electronic device and the host is being established through a secure channel (115).

15. The method according to any one of the preceding claims, wherein the electronic device (120) and the virtual machine support (12) is connected through a secure connection channel (115), or the first and the second network interface and the virtual machine support (12) is connected through a secure connection channel (115).

16. The method according to any of the preceding claims, wherein an authentication between the electronic device (120) and the secure host (110) is carried out before the establishment of the connection therebetween.

17. An electronic device (20, 120) comprising a virtual appliance stored on a memory (17, 18) of the electronic device (20, 120), said virtual appliance and being capable to initiate a virtual machine on a host, wherein a device software for driving components of the electronic device is further stored on said memory (17, 18) and preferably encapsulated in the virtual appliance.

18. The electronic device (20, 120) according to claim 17, wherein the device software for driving the electronic device is provided for a particular operating system.

19. A system for providing connectivity between an electronic device (20, 120) according to claims 17 or 18 and a host (10, 110) being capable to manage a virtual machine by using a hypervisor (16, 116).

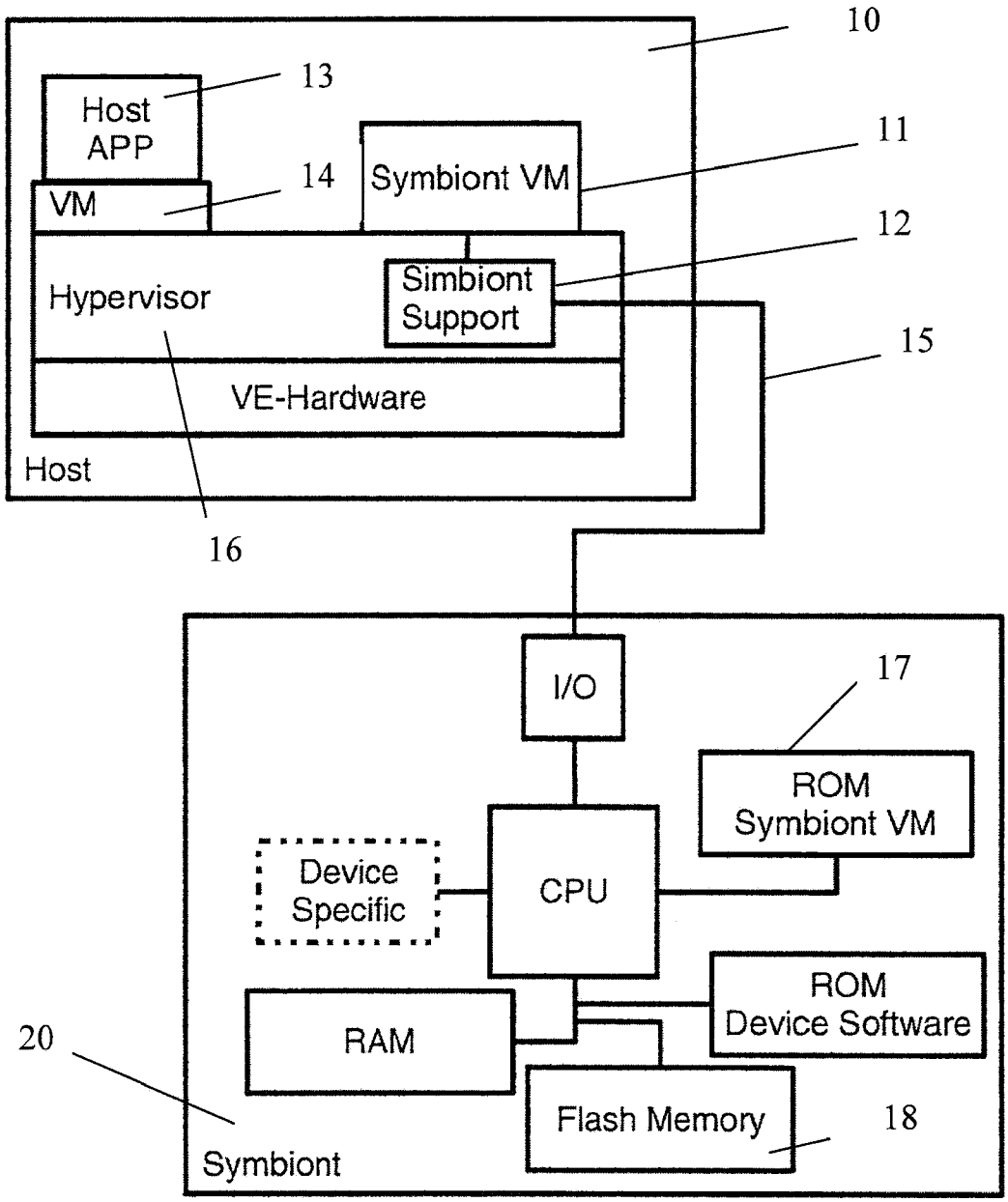
10
13
Host APP
14
VM
Symbiont VM
11
12
Hypervisor
Simbiont Support
15
VE-Hardware
Host
16
I/O
17
ROM Symbiont VM
Device Specific
CPU
ROM Device Software
RAM
20
Flash Memory
18
Symbiont

Fig. 1

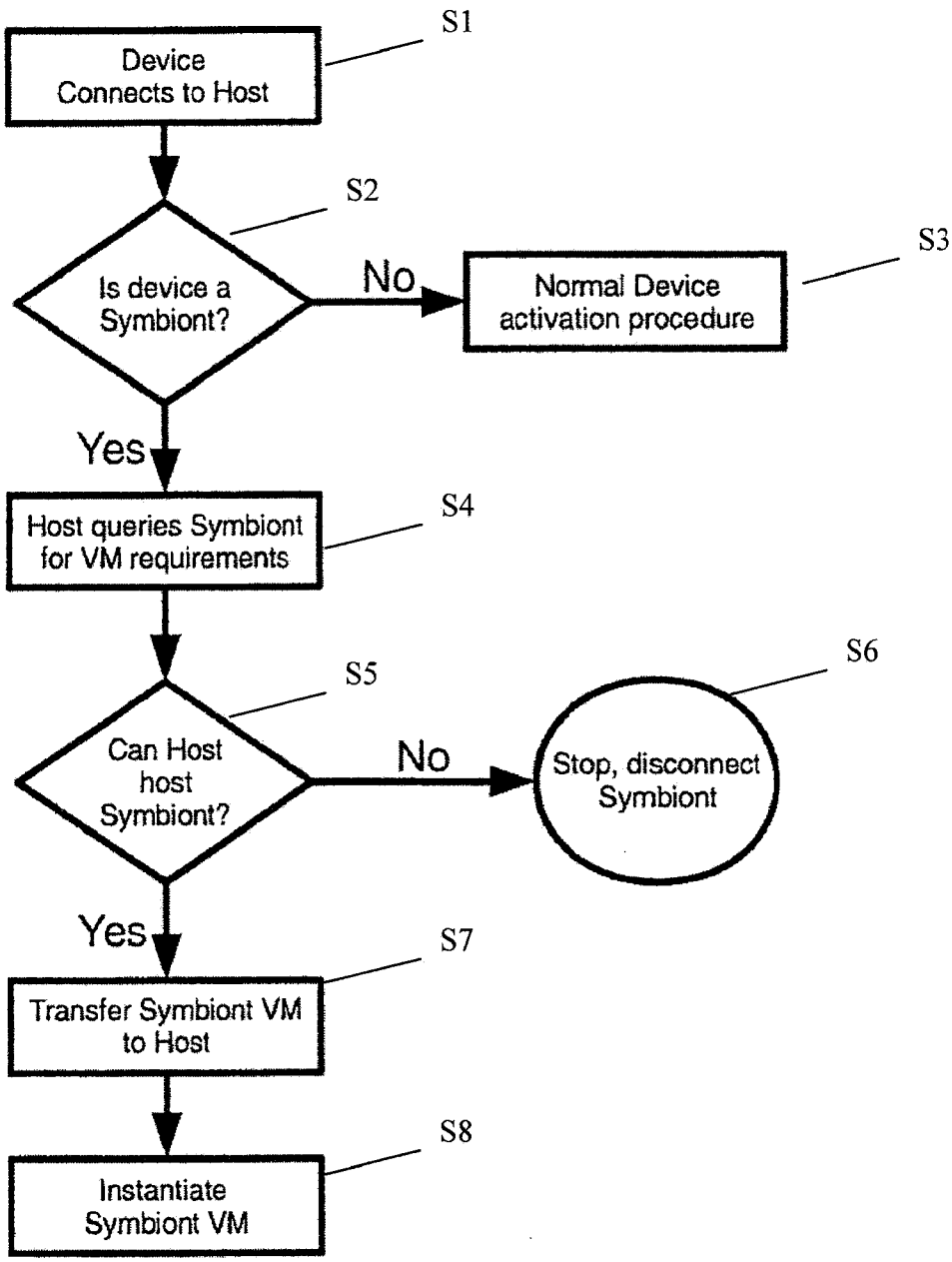
S1
Device Connects to Host
S2
Is device a Symbiont?
No
S3
Normal Device activation procedure
Yes
S4
Host queries Symbiont for VM requirements
S5
Can Host host Symbiont?
No
S6
Stop, disconnect Symbiont
Yes
S7
Transfer Symbiont VM to Host
S8
Instantiate Symbiont VM

Fig. 2

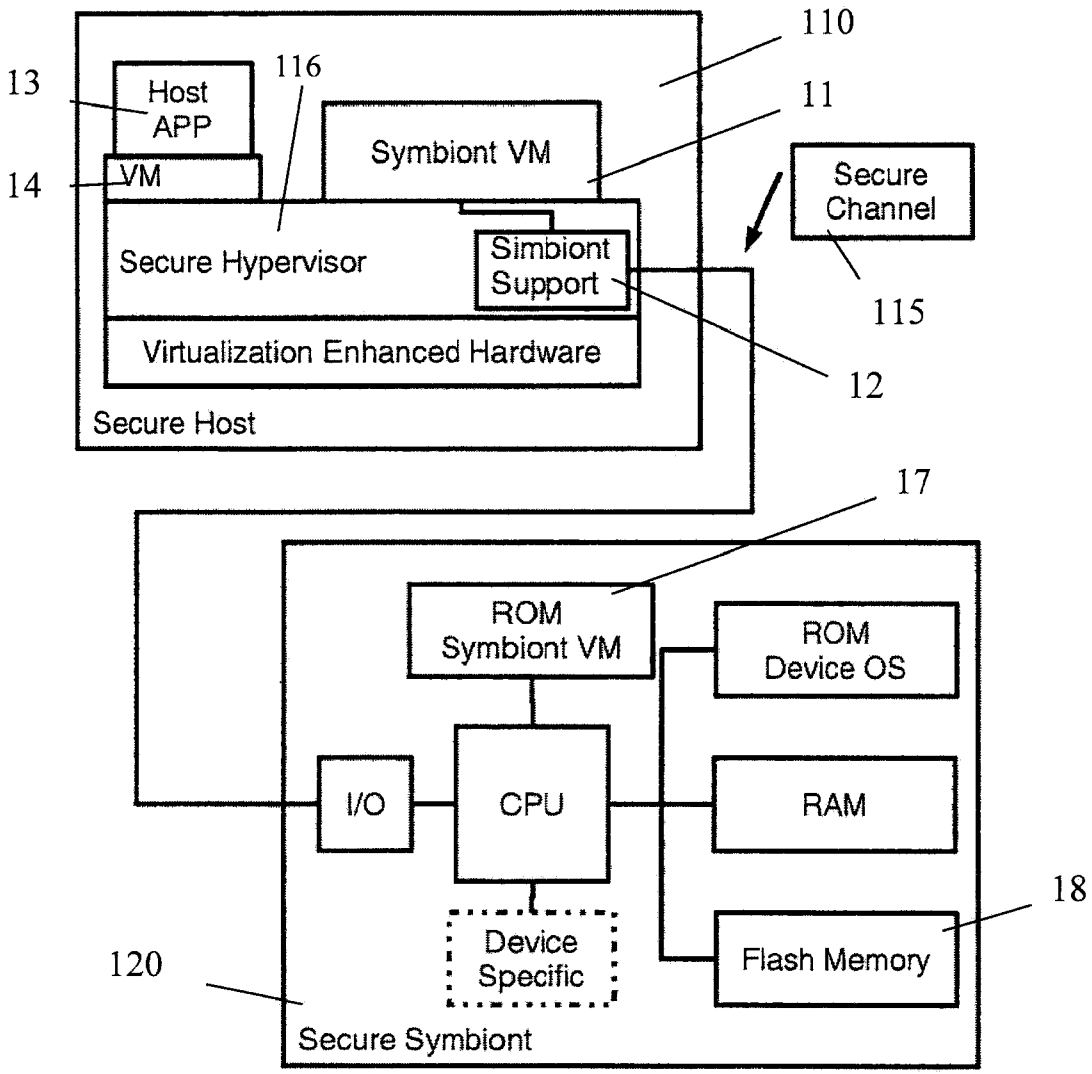
110
13
Host APP
116
11
Symbiont VM
14
VM
Secure Channel
Secure Hypervisor
Simbiont Support
115
Virtualization Enhanced Hardware
12
Secure Host
17
ROM Symbiont VM
ROM Device OS
I/O
CPU
RAM
18
Device Specific
Flash Memory
120
Secure Symbiont

Fig. 3

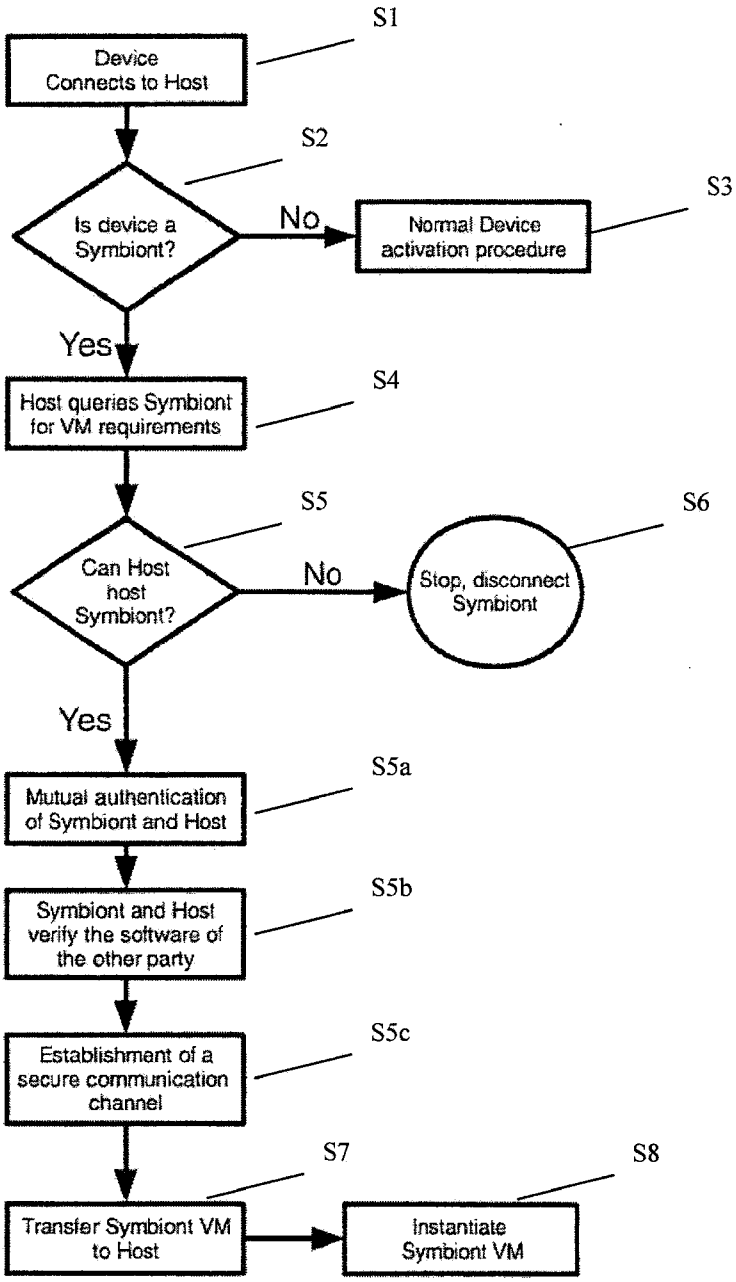
S1
Device Connects to Host
S2
Is device a Symbiont?
No
S3
Normal Device activation procedure
Yes
S4
Host queries Symbiont for VM requirements
S5
Can Host host Symbiont?
No
S6
Stop, disconnect Symbiont
Yes
S5a
Mutual authentication of Symbiont and Host
S5b
Symbiont and Host verify the software of the other party
S5c
Establishment of a secure communication channel
S7
Transfer Symbiont VM to Host
S8
Instantiate Symbiont VM

Fig. 4

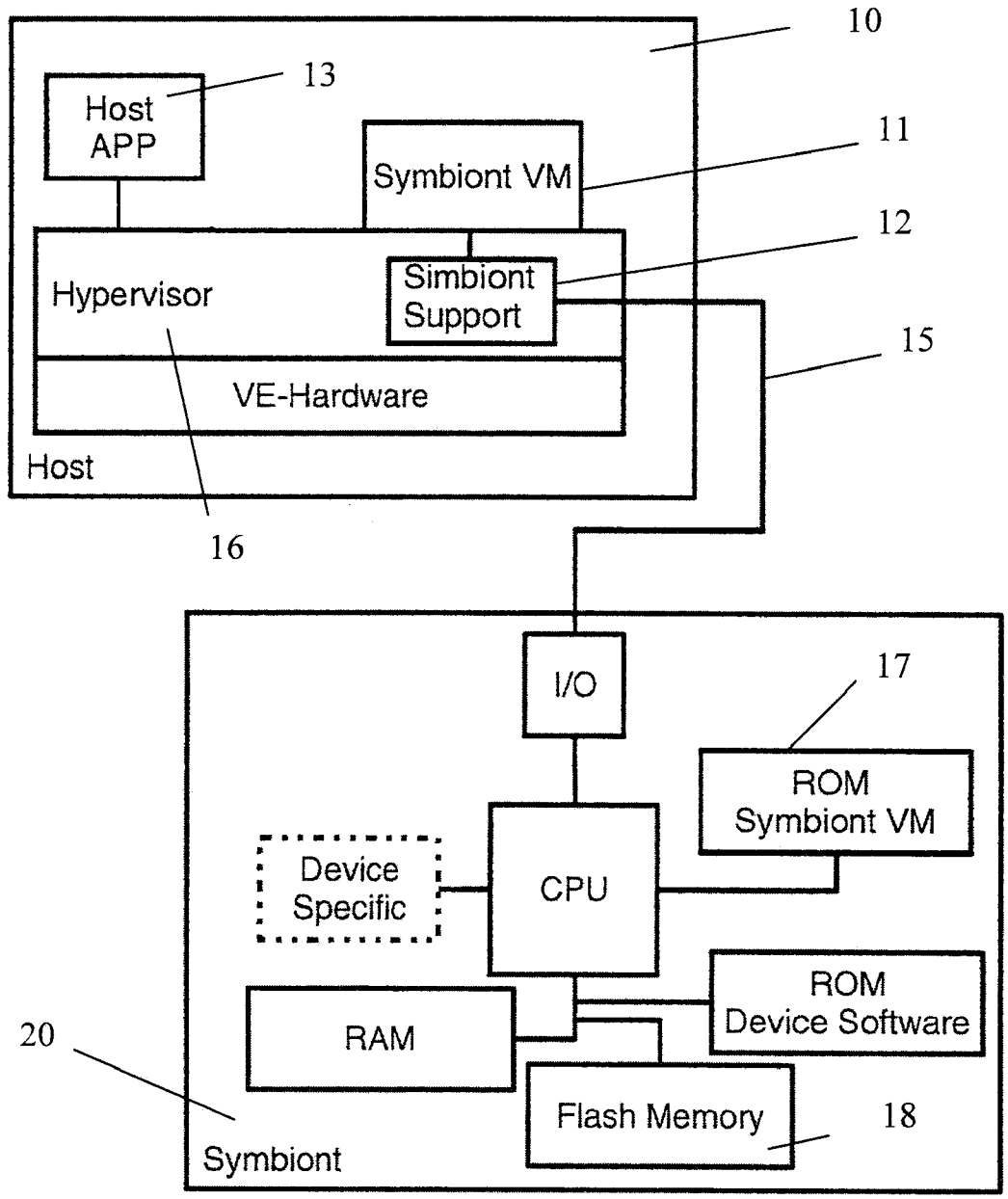
10
13
Host APP
11
Symbiont VM
12
Hypervisor
Simbiont Support
15
VE-Hardware
Host
16
I/O
17
ROM Symbiont VM
Device Specific
CPU
ROM Device Software
RAM
20
Flash Memory
18
Symbiont

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/065837

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/455 G06F9/445 H04L29/08 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 813 669 B1 (CATO ROBERT THOMAS [US] ET AL) 2 November 2004 (2004-11-02) abstract paragraph [0016] - paragraph [0017]	1-19
X	EP 1 351 138 A (MATSUSHITA ELECTRIC IND CO LTD [JP]) 8 October 2003 (2003-10-08) abstract paragraphs [0071], [0078], [0094]	1-19
X	EP 1 056 001 A (XEROX CORP [US]) 29 November 2000 (2000-11-29) abstract	1-19

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

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

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 March 2009	16/03/2009

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Fax: (+31–70) 340–3016	Beyer, Steffen

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/065837

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2008/066574 A (SONY ERICSSON MOBILE COMM AB [SE]; MULLIS II SAMUEL L [US]; LENNON WIL) 5 June 2008 (2008-06-05) the whole document	1-19
X	figure 5	11
A	WO 97/49023 A (HANSON GORDON L [US]) 24 December 1997 (1997-12-24) the whole document	1-19
A	WO 97/24656 A (INTEL CORP [US]; THURLO CLARK S [US]) 10 July 1997 (1997-07-10) abstract	1-19
A	WO 2007/149671 A (SENTILLION INC [US]; SELIGER ROB [US]; HARTZ GEORGE [US]; FONTANA ERIC) 27 December 2007 (2007-12-27) the whole document	1-19
X	claims 6,17	14-16
A	US 6 754 725 B1 (WRIGHT DAVID G [US] ET AL) 22 June 2004 (2004-06-22) cited in the application abstract	1-19

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2008/065837

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6813669 B1	02-11-2004	JP 3636080 B2	06-04-2005
		JP 2001265705 A	28-09-2001
		KR 20010085354 A	07-09-2001
		TW 498211 B	11-08-2002
EP 1351138 A	08-10-2003	US 2005259473 A1	24-11-2005
		US 2003192042 A1	09-10-2003
EP 1056001 A	29-11-2000	JP 2001005622 A	12-01-2001
WO 2008066574 A	05-06-2008	US 2008127225 A1	29-05-2008
WO 9749023 A	24-12-1997	CA 2258596 A1	24-12-1997
		EP 0974090 A2	26-01-2000
		JP 2001511920 T	14-08-2001
		US 6148346 A	14-11-2000
WO 9724656 A	10-07-1997	AU 1568497 A	28-07-1997
		US 5835772 A	10-11-1998
WO 2007149671 A	27-12-2007	US 2007300220 A1	27-12-2007
US 6754725 B1	22-06-2004	NONE	