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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 July 2008 (10.07.2008)

PCT

(10) International Publication Number
WO 2008/083168 A1

(51) International Patent Classification:
G06F 9/445 (2006.01)

(21) International Application Number:
PCT/US2007/088839

(22) International Filing Date:
26 December 2007 (26.12.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

11/618,519	29 December 2006 (29.12.2006)	US
11/618,526	29 December 2006 (29.12.2006)	US

(71) Applicant *(for all designated States except US)*: **SANDISK CORPORATION** [US/US]; 601 McCarthy Boulevard, Milpitas, California 95035-7932 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LAPEDIS, Ron** [US/US]; 2115 Sea Cliff Way, San Bruno, California 94066 (US). **TRIULZI, Arrigo** [IT/CH]; 67, Rue des Lattes, CH-1217 Meyrin (CH).

(74) Agents: **SINGH, Tejinder** et al.; Klein, O'neill & Singh, LLP, 43 Corporate Park, Suite 204, Irvine, CA 92606 (US).

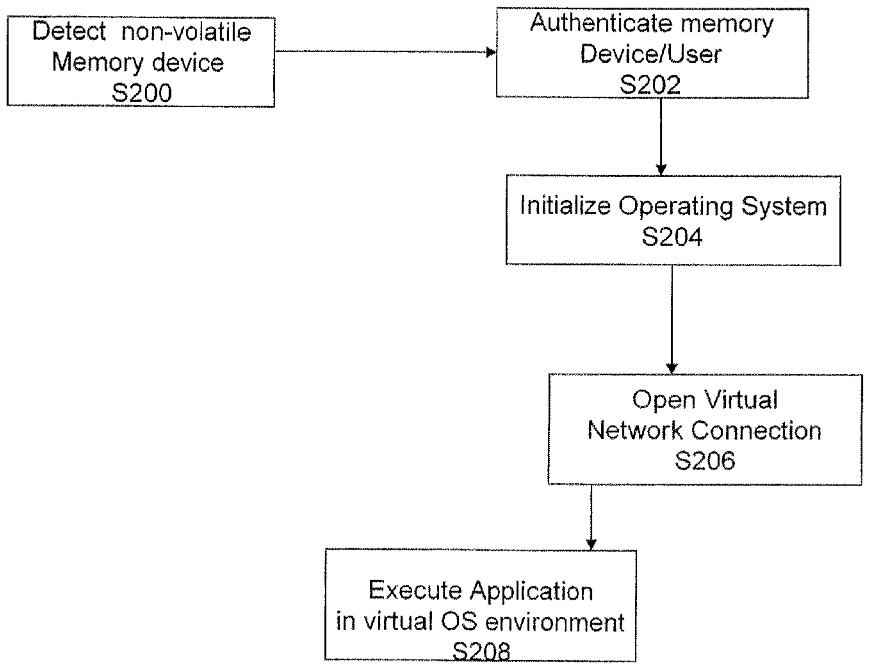
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, 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, 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, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

[Continued on next page]

(54) Title: METHOD FOR CODE EXECUTION


Detect non-volatile Memory device S200
Authenticate memory Device/User S202
Initialize Operating System S204
Open Virtual Network Connection S206
Execute Application in virtual OS environment S208

(57) **Abstract:** Method for executing a software application is provided. The method includes detecting a host operating system; executing a virtual operating system in a virtual environment, wherein the virtual operating system is stored in a non-volatile memory device; and executing a software application in the virtual environment, wherein the software application is stored in the non-volatile memory device; and the virtual operating system and the software application are executed independently of the host operating system execution.

WO 2008/083168 A1

Published:
— *with international search report*

METHOD FOR CODE EXECUTION

Inventors:

Ron LaPedis

Arrigo Triulzi

BACKGROUND

1. Technical Field

[0001] The present disclosure relates to computing systems, and more particularly, to code execution.

2. Related Art

[0002] Computing systems (stand-alone and networked) are commonplace. The Internet has increased the popularity of electronic commerce, where users of computing systems conduct millions of electronic transactions. This increase in popularity has also made computing systems and user information vulnerable to pirates (sometimes known as "hackers").

[0003] Operating systems are computer programs used to perform certain computing tasks, such as, for example, managing input/output tasks, peripheral devices (for example, storage devices) and file systems. Operating systems provide a software platform on top of which other software applications are written. Software applications are used for various tasks, including, for example, word-processing, electronic mail (email) and Internet browsing. Some common operating systems include Windows®, Linux®, IBM® OS/2, MacOS, UNIX, and MS-DOS.

[0004] Many operating systems are pirated (or "hacked," *i.e.*, they experience unauthorized use or interruption) through use of disruptive software programs, such as those known as computer viruses, worms, key-loggers, and root-kits. Securing operating systems and overall application code execution is a challenge.

[0005] Furthermore, software applications will often run on multiple operating systems or hardware platforms. Typically, separate code for a software application must be created for each different environment/platform. This is expensive and undesirable. An efficient method and system are desirable for managing code for software applications to be used on different hardware and software platforms.

SUMMARY

[0006] In one embodiment, a non-volatile memory device is provided. The non-volatile memory comprises a plurality of memory cells, wherein a read only segment of a plurality of memory cells stores (a) code for a micro-operating system for running a virtual engine; (b) code for the virtual engine that provides a virtual environment, independent of a host operating system; (c) code for a virtual operating system that is executed in the virtual environment; and (d) code for a software application, wherein the code for the software application can be executed in different host system platforms in the virtual environment.

[0007] In another embodiment, a system for code execution is provided.. The system comprises a host computing system; and a non-volatile memory device operationally coupled to the host computing system, the non-volatile memory device comprising a plurality of memory cells, wherein a read only segment of a plurality of memory cells stores: (a) code for a micro-operating system for running a virtual engine; (b) code for the virtual engine that provides a virtual environment, independent of a host operating system; (c) code for a virtual operating system that is executed in the virtual environment;

and (d) code for a software application, wherein the code for the software application can be executed in different host system platforms in the virtual environment.

[0008] In yet another embodiment of the present disclosure, a method for executing a software application is provided. The method includes detecting a host operating system; executing a virtual operating system under a virtual engine, wherein the virtual operating system and the virtual engine are stored in a non-volatile memory device; and executing a software application in a virtual environment, wherein the software application is stored in the non-volatile memory device. The virtual operating system and the software application are executed independently of the host operating system execution.

[0009] In another embodiment of the present disclosure, a method for executing a software application is provided. The method includes authenticating a non-volatile memory device coupled to a host system; detecting a host operating system by the non-volatile memory device; executing a virtual operating system under a virtual engine, wherein the virtual operating system and the virtual engine are stored in a read only segment of the non-volatile memory device; and executing the software application in a virtual environment, wherein the software application is stored in a read-only segment of the non-volatile memory device. The virtual operating system and the software application are executed independently of the host operating system execution.

[0010] This brief summary is not intended to limit the disclosure to any particular embodiment. Rather, the disclosure is intended to cover the subject matter defined by the claims appended hereto, and all equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The foregoing features and other features will now be described with reference to the drawings of a preferred embodiment. In the drawings, the same components have the same reference numerals. The illustrated embodiments are intended to illustrate, but not to limit the disclosure. The drawings include the following Figures:

[0012] Figure 1A shows a block diagram of a computing system of an embodiment;

[0013] Figure 1B shows a block diagram of a memory controller in Figure 1A,

[0014] Figure 1C shows a top-level block diagram of a system of an embodiment for authenticating a non-volatile memory device;

[0015] Figure 1D shows a block diagram of a software architecture of an embodiment; and

[0016] Figure 2 shows a process flow diagram of an embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] To facilitate an understanding of the preferred embodiment, the general architecture and operation of a computing system/non-volatile memory storage device will first be described. The specific architecture and operation of the preferred embodiment will then be described with reference to the general architecture.

[0018] Figure 1A shows a block diagram of a typical computing system (may also be referred to as "host system" or "host") 100 that includes a central processing unit ("CPU") (may also be referred to as microprocessor/processor) 101 operationally coupled to a system bus 101B. Random access memory ("RAM") 103 provides CPU 101 with access to memory storage. When executing program instructions, CPU 101 stores those process steps (code) in RAM 103 and executes the stored process steps out of RAM 103.

[0019] Read only memory ("ROM") 102 is provided to store invariant instruction sequences such as start-up instruction sequences or Basic Input/Output Operating System (BIOS) sequences.

[0020] Input/Output ("I/O") devices 102A, such as, for example, a keyboard, a pointing device ("mouse"), a monitor, a modem and the like, are also provided for receiving input/output instructions.

[0021] Host system 100 optionally connects to a computer network (not shown) via network interface 101A. One such network is the Internet that allows host system 100 to download applications, code, documents and others electronic information.

[0022] Host system 100 is coupled to a non-volatile memory device (for example, a flash memory device (or card)) 105 that includes a controller module 106 (may also be referred to as "memory controller" or "controller"), and solid-state memory modules (may also be referred to as cells/cell arrays) 107-108 (shown as Memory Module #1 and Memory Module #n). Controller module 106 interfaces with host system 100 via a bus interface 104, directly via system bus 101B or any other peripheral bus (not shown).

[0023] Non-volatile memory device 105 includes a processor (shown as "crypto-engine") 106A that performs various cryptographic functions, for example, encrypting and/or decrypting stored content. Crypto-engine 106A may also be used to authenticate a non-volatile memory device, as described below.

[0024] In some embodiments, non-volatile memory devices are flash memory devices. There are currently many different flash memory cards that are commercially available, examples being the CompactFlash (CF), the MultiMediaCard (MMC), Secure Digital (SD), miniSD, Memory Stick, SmartMedia and TransFlash cards. Although each of these cards has a unique mechanical and/or electrical interface according to its standardized specifications (for example, the Universal Serial Bus (USB) specification based interface, incorporated herein by reference in its entirety), the flash memory included in each card is very similar. These cards are all available from SanDisk Corporation, assignee of the present application.

[0025] SanDisk Corporation also provides a line of flash drives under its Cruzer trademark, which are hand held memory systems in small packages that have a Universal Serial Bus (USB) plug for connecting with a host by plugging into the host's USB receptacle. Each of these memory cards and flash drives includes controllers that interface with the host and control operation of the flash memory on the card or drive.

[0026] Host systems that use such memory cards and flash drives are many and varied. They include personal computers (PCs), laptop and other portable computers, cellular telephones, personal digital assistants (PDAs), digital still cameras, digital movie cameras and portable audio players. Host systems typically include a built-in receptacle for one or more types of memory cards or flash drives but some require adapters into which a memory card is plugged.

[0027] A NAND architecture of the memory cell arrays 107-108 is currently preferred, although other architectures, such as NOR, can also be used instead. Examples of NAND flash memories and their operation as part of a memory system may be had by reference to United States Patents numbers 5,570,315; 5,774,397; 6,046,935; 6,373,746;

6,456,528; 6,522,580; 6,771,536 and 6,781,877 and United States Patent Application Publication number 2003/0147278.

[0028] The various embodiments described herein are not limited to the foregoing structures. Various other structures and memory types may be used, for example, non-flash memory devices can be used with this disclosure, such as one time programmable memory devices or 3D memory devices, which may include a monolithic three-dimensional memory array. In a three dimensional memory array, multiple memory levels are formed above a single substrate, such as a wafer, with no intervening substrates. The layers forming one memory level are deposited or grown directly over the layers of an existing level or levels. In contrast, stacked memories have been constructed by forming memory levels on separate substrates and adhering the memory levels atop each other, as in Leedy, US Patent No. 5,915,167, "Three dimensional structure memory." The substrates may be thinned or removed from the memory levels before bonding, but as the memory levels are initially formed over separate substrates, such memories are not true monolithic three dimensional memory arrays.

[0029] Figure 1B shows a block diagram of the internal architecture of controller module 106. Controller module 106 includes a microcontroller 109 that interfaces with various other components via interface logic 111. Memory 110 stores firmware and/or software instructions that are used by microcontroller 109 to control the operation of non-volatile memory device 105. Memory 110 may be volatile re-programmable random access memory ("RAM"), a non-volatile memory that is not re-programmable ("ROM"), a one-time programmable memory or a re-programmable flash electrically-erasable and programmable read-only memory ("EEPROM").

[0030] A host interface 113 interfaces with host system 100, while a flash interface 112 interfaces with memory modules 107-108.

[0031] Figure 1C shows a block diagram of system 100A where non-volatile memory device 105 interfaces with host system 100 via a USB interface 100B. A remote server 114 authenticates non-volatile memory device 105 before a user is allowed to use the non-volatile memory device.

[0032] Accordingly, in one embodiment, non-volatile memory device 105 conforms to the USB specification (i.e. can be accessed via a USB interface). Standard USB based application programming interface (API) may be used for reading or writing data.

[0033] Non-volatile memory device 105 appears to host 100 having a plurality of Logical Units (LUNs) of storage space and each LUN may appear to be of a different class of storage device. For example, non-volatile memory device 105 may appear to have both a standard Mass Storage Class volume (LUN 0, 107A), which imitates the behavior of a SCSI Hard Disk Drive, and a MMC Class volume, which imitates the behavior of a CD-ROM (LUN 1, 107B).

[0034] LUN 1 107B may store a plurality of software applications (116, Figure 1D), a minimal version of an operating system ("Micro-OS") (115, Figure 1D), code for a virtual engine (120, Figure 1D), and other information, discussed below with respect to Figure 1D.

[0035] Hidden area 107C is secured and is not available without proper authentication. Proprietary APIs may be used to access hidden area 107C. In one aspect, a protected (or secured) area means an area that is read-only and accessible only by an appropriate authenticated entity, for example, a host program, and the like. Hidden area

107C may store device certificates (118, Figure 1D) and security keys (119, Figure 1D) and other code, as described below with respect to Figure 1D.

[0036] It is noteworthy that although host system 100 has been described above as having a CPU, ROM, RAM and other components, the adaptive aspects of the present disclosure may be implemented on a "thin" client, i.e., a host system that has limited computing abilities. For example, a USB reader/executor with a keyboard, mouse, video card, network card and CPU can execute whatever code/application is stored on non-volatile memory device 105, instead of a desktop or notebook computer.

[0037] Figure 1D shows a plurality of software components that may be stored in non-volatile memory device 105 and used according to one aspect of the present disclosure. Firmware 117 is used to control the overall operation of non-volatile memory device 105 and is executed by controller 106.

[0038] Micro-OS 115 is a minimal version of an operating system, i.e., it has reduced functionality compared to a standard operating system. Micro-OS 115 is used to control the overall environment in which code for a virtual engine is executed. Micro-OS 115 may be customized to run code for the virtual engine, described below. Micro-OS may be stored in the read only segment 107B (Figure 1C).

[0039] Application 116 may be a software application that a user may want to execute on different hardware/software platforms. More than one application 116 may be stored in non-volatile memory device 105.

[0040] Application 116 may include a web browser, for example, Firefox®, that a user uses to browse websites. The web browser may run on any computer connected to the Internet. The web browser receives and sends requests to a web server and acquires information from a World Wide Web (WWW), a network of computers. A web server is a program that, upon receipt of a request, sends requested data to a requesting user.

[0041] Virtual engine (or machine) (VE) 120 includes code for providing a virtual environment. The virtual environment provides a software platform that is independent of a host operating system. Code that is executed in the virtual environment is not controlled by the host operating system, but instead is controlled by a virtual operating system executed within the virtual environment.

[0042] VE 120 also includes executable code for different operating systems executed in the virtual environment, independent of the underlying host operating system. Micro-OS 115 controls the overall execution of VE 120.

[0043] Code blocks for different operating systems are shown as VOS1, VOS2, VOS3 and VOSn. VOS1 may be used to execute a Windows® based operating system, VOS2 may be used for a Linux operating system, VOS2 may be used for a UNIX based operating systems and so forth. The operating system specific code (VOS1-VOSn) is executed in a virtual environment, independent of the host system 100 operating system. VE 120 allows a user to use non-volatile memory device 105 on different hardware/software platforms.

[0044] Different types of virtual engines 120 may be used to implement the adaptive aspects of the present disclosure. For example, VMWare Player and VMWare Ace available from VMware Corporation; VirtualPC available from Microsoft Corporation; and others may be used . It is noteworthy that more than one virtual engine may be stored and used for application execution. This will make it more difficult for pirates to break into the operating system because the viruses or other disruptive software will have to hook to low-level support for two or more virtual engines instead of one virtual engine.

[0045] The non-volatile memory device 105 during an authentication stage, as described below uses device certificates 118.

[0046] Security keys 119 may be used to generate a one-time password to authenticate a user/device. Security keys 119 may be used by crypto-engine 106A to encrypt stored content, using standard or proprietary encryption techniques.

[0047] Virtual private network (VPN) code 121 is provided to facilitate a VPN connection, as described below. Access to VPN code 121 is limited (by storing in LUN 1 (107B), Figure 1C) so that a virtual connection is difficult to pirate or break into.

[0048] It is noteworthy that executable code for a plurality of software components (Micro-OS 115, application 116, firmware 117, device certificates 118, security keys 119, virtual engine 120 and VPN code 122) may be stored in secured segment 107C or in read only segment 107B. Furthermore, executable code for the plurality of components may be bifurcated and partially stored in the read only segment 107B and secured segment 107C.

[0049] Figure 2 shows a process flow diagram for securing operating system/application execution in one aspect of the present disclosure. The process starts in step S200, when non-volatile memory device 105 is coupled to host system 100 that detects non-volatile memory device 105. Most host systems today have a "Plug-N-Play" option where a device is detected as soon as it is plugged in. Firmware 117 or hardware (not shown) detects the type of operating system that is running on host system 100.

[0050] After the host operating system is detected, in step S202, non-volatile memory device 105 and a user using the device are authenticated. In one aspect, server 114 authenticates non-volatile memory device 105 using device certificates 118. A standard or proprietary technique may be used to authenticate non-volatile memory device 105. For example, a public key infrastructure (PKI) certificate (for example, 118) may be used to authenticate non-volatile memory device 105. A user using non-volatile memory device 105 may also have to authenticate itself before being allowed access to non-volatile memory device 105. This may be performed by using a unique, user-specific password, generated by using security keys 119. Crypto-engine 106A may be used to authenticate non-volatile memory device 105 and the user. Step S202 attempts to prevent unauthorized use of non-volatile memory device 105.

[0051] In step S204, non-volatile memory device 105 loads code (VOS1, VOS2, VOS3, or VOSn) for a virtual machine into RAM 103. In one aspect of the present disclosure, controller 106 may execute virtual engine code 120 to initialize a virtual environment. In another aspect, virtual engine code 120 execution may be split such that one code segment is executed by the host CPU (101, Figure 1A) and another segment is executed by non-volatile memory device 105. This makes pirating or hacking difficult.

[0052] After virtual engine code 120 is initialized, all other applications/code (for example, application 116) is executed in a virtual environment independent of the host operating system.

[0053] In step S206, host system 100 opens a virtual private network (VPN) connection (not shown) to an enterprise server or gateway (not shown). The nature of the network connection will depend on the connection, e.g., whether the connection is to a web server or local area network. VPN code 121 may be used to open the VPN connection.

[0054] In step S208, application 116 is executed in the appropriate virtual operating system environment. Application 116 is executed in a virtual environment controlled by virtual engine 120, independent of the host operating system. Hence, it is difficult to break into (hack into) application 116 execution.

[0055] In one aspect of the present disclosure, code for application 116 is written so that it may be executed in a virtual environment, which may be independent of a host system operating system.. Hence, different versions for application 116 for different operating systems and platforms are not needed. This reduces overall cost of code development/maintenance.

[0056] In another aspect of the present disclosure, because virtual engine 120 and application 116 are stored in a read only segment (for example, 107B (or 107C)) of non-volatile memory device 105, they are difficult to pirate.

[0057] In yet another aspect of the present disclosure, a secure environment is provided to a user to conduct electronic commerce transactions, for example, bank transactions, without changing overall user experience. Once non-volatile memory device 105 is connected and the virtual environment is launched, the user simply navigates to a website with minimal pirating risk.

[0058] While the present disclosure is described above with respect to what is currently considered its preferred embodiments, it is to be understood that the disclosure is not limited to that described above. To the contrary, the disclosure is intended to cover various modifications and equivalent arrangements within the spirit and scope of the appended claims.

What is claimed is:

1. A non-volatile memory device comprising:

a plurality of memory cells, wherein a read only segment of a plurality of memory cells stores (a) code for a micro-operating system for running a virtual engine; (b) code for the virtual engine that provides a virtual environment, independent of a host operating system; (c) code for a virtual operating system that is executed in the virtual environment; and (d) code for a software application, wherein the code for the software application can be executed in different host system platforms in the virtual environment.

2. The non-volatile memory device of Claim 1, wherein a secure non-volatile memory device segment stores a device certificate used to authenticate the non-volatile memory device.

3. The non-volatile memory device of Claim 2, wherein a remote server authenticates the non-volatile memory device.

4. The non-volatile memory device of Claim 1, wherein the non-volatile memory device detects the host operating system.

5. The non-volatile memory device of Claim 1, wherein the software application is a web-browser.

6. The non-volatile memory device of Claim 1, wherein the code for the virtual engine, the code for the virtual operating system and the code for the software application are executed by a host system processor after being loaded by the non-volatile memory device to a host system memory.

7. The non-volatile memory device of Claim 1, wherein the code for the virtual engine, the code for the virtual operating system and the code for the software application are executed by a non-volatile memory device controller.

8. The non-volatile memory device of Claim 1, wherein the non-volatile memory device is a universal serial bus (USB) device that interfaces with the host system via a USB interface.

9. The non-volatile memory device of Claim 1, wherein the code for the virtual operating system and the code for the software application are stored in a secured area of the non-volatile memory device.

10. The non-volatile memory device of Claim 1, wherein the host system opens a virtual network connection based on code stored in the non-volatile memory device.

11. The non-volatile memory device of Claim 1, wherein the code for the software application can be executed in more than one hardware and software environment.

12. A system for code execution, comprising:

a host computing system; and

a non-volatile memory device operationally coupled to the host computing system, the non-volatile memory device comprising a plurality of memory cells, wherein a read only segment of a plurality of memory cells stores: (a) code for a micro-operating system for running a virtual engine; (b) code for the virtual engine that provides a virtual environment, independent of a host operating system; (c) code for a virtual operating system that is executed in the virtual environment; and (d) code for a software application, wherein the code for the software application can be executed in different host system platforms in the virtual environment.

13. The system of Claim 12, wherein a secure non-volatile memory device segment stores a device certificate used to authenticate the non-volatile memory device.

14. The system of Claim 13, wherein a remote server authenticates the non-volatile memory device.

15. The system of Claim 12, wherein the non-volatile memory device detects the host operating system.

16. The system of Claim 12, wherein the software application is a web-browser.

17. The system of Claim 12, wherein the code for the virtual engine, the code for the virtual operating system and the code for the software application are executed by a host system processor after being loaded by the non-volatile memory device to a host system memory.

18. The system of Claim 12, wherein the code for the virtual engine, the code for the virtual operating system and the code for the software application are executed by a non-volatile memory device controller.

19. The system of Claim 12, wherein the non-volatile memory device is a universal serial bus (USB) device that interfaces with the host system via a USB interface.

20. The system of Claim 12, wherein the code for the virtual operating system and the code for the software application are stored in a secured area of the non-volatile memory device.

21. The system of Claim 12, wherein the host system opens a virtual network connection based on code stored in the non-volatile memory device.

22. The system of Claim 12, wherein the code for the software application can be executed in more than one hardware and software environment.

23. A method for executing a software application, comprising:

authenticating a non-volatile memory device coupled to a host system;

detecting a host operating system, wherein the non-volatile memory device detects the host operating system;

executing a virtual operating system in a virtual environment, wherein the virtual operating system is stored in a read only segment of the non-volatile memory device; and

executing the software application in the virtual environment, wherein the software application is stored in a read-only segment of the non-volatile memory device; and the virtual operating system and the software application are executed independently of the host operating system execution.

24. The method of Claim 23, wherein the software application is a web-browser.

25. The method of Claim 23, wherein a remote server authenticates the non-volatile memory device using a device certificate stored in a secure area of the non-volatile memory device.

26. The method of Claim 23, wherein the virtual operating system and the software application are executed by a host system processor after the virtual operating system and the software application are loaded by the non-volatile memory device to a host system memory.

27. The method of Claim 23, wherein the virtual operating system and the software application are executed by a non-volatile memory device controller.

28. The method of Claim 23, wherein the non-volatile memory device is a universal serial bus (USB) device that interfaces with the host system via a USB interface.

29. The method of Claim 23, wherein the virtual operating system and the software application are stored in a secured area of the non-volatile memory device.

30. The method of Claim 23, wherein the host system opens a virtual network connection based on code stored in the non-volatile memory device.

31. The method of Claim 23, wherein the software application can be executed in more than one hardware and software environment.

32. The method of Claim 23, wherein the virtual environment is controlled by a virtual engine; and the virtual engine is stored in a read only segment of the non-volatile memory device.

33. A method for executing a software application, comprising:

detecting a host operating system;

executing a virtual operating system in a virtual environment, wherein the virtual operating system is stored in a non-volatile memory device; and executing a software application in the virtual environment, wherein the software application is stored in the non-volatile memory device; and the virtual operating system and the software application are executed independently of the host operating system execution.

34. The method of Claim 33, wherein the software application is a web-browser.

35. The method of Claim 33, wherein a remote server authenticates the non-volatile memory device using a device certificate stored in the non-volatile memory device.

36. The method of Claim 33, wherein the virtual operating system and the software application are stored in a read-only segment of the non-volatile memory device; and the virtual operating system and the software application are executed by a host system processor after the virtual operating system and the software application are loaded by the non-volatile memory device to a host system memory.

37. The method of Claim 33, wherein the virtual operating system and the software application are stored in a read-only segment of the non-volatile memory device; and the virtual operating system and the software application are executed by a non-volatile memory device controller.

38. The method of Claim 33, wherein the non-volatile memory device is a universal serial bus (USB) device that interfaces with the host system via a USB interface.

39. The method of Claim 33, wherein the virtual operating system and the software application are stored in a secured area of the non-volatile memory device.

40. The method of Claim 33, wherein the host system opens a virtual network connection based on code stored in the non-volatile memory device.

41. The method of Claim 33, wherein the software application can be executed in more than one hardware and software environment.

42. The method of Claim 33, wherein the virtual environment is controlled by a virtual engine; and code for the virtual engine is stored in a read only segment of the non-volatile memory device.

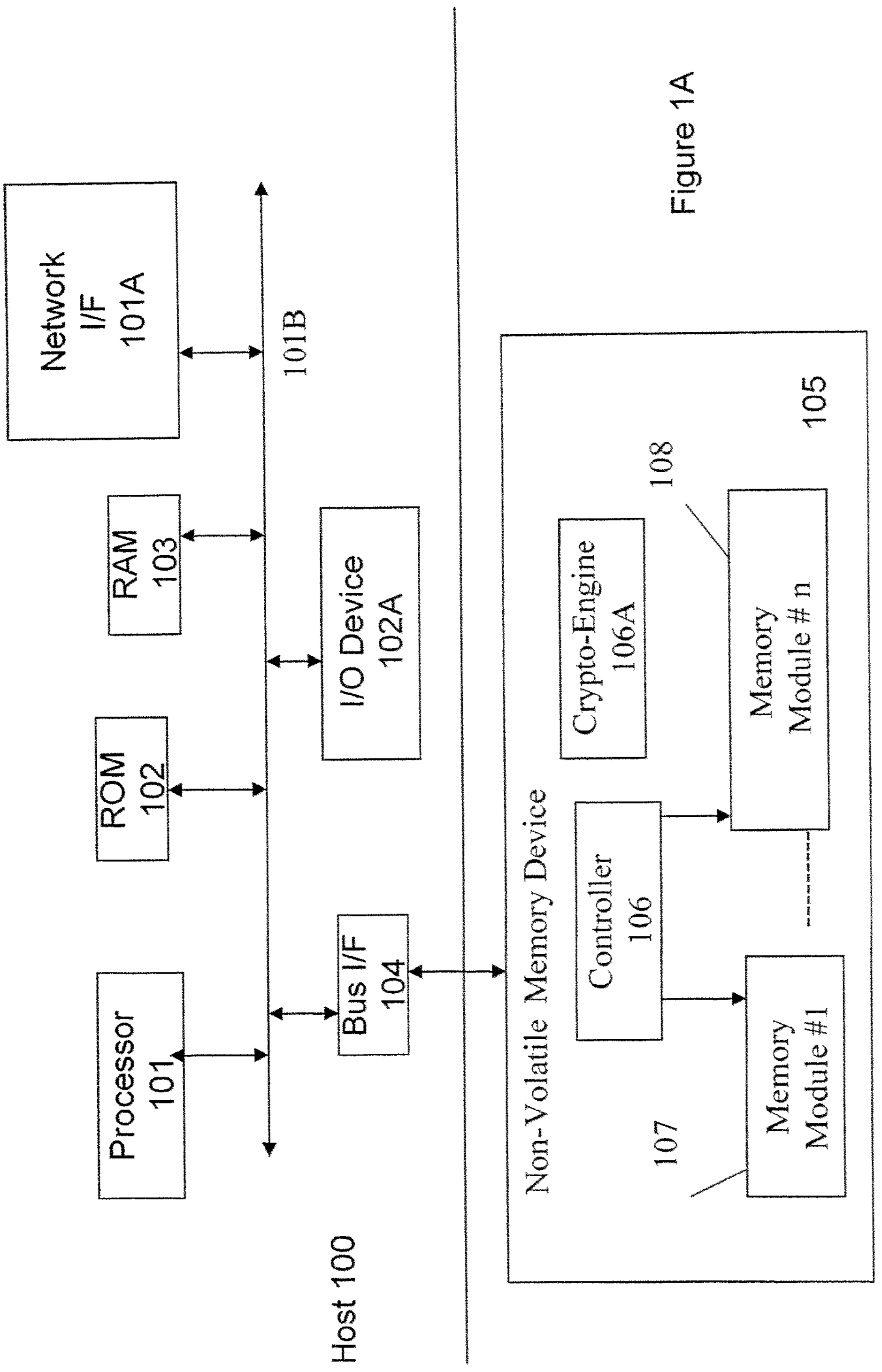

Network I/F 101A
RAM 103
ROM 102
Processor 101
101B
I/O Device 102A
Bus I/F 104
Host 100
Non-Volatile Memory Device
Crypto-Engine 106A
Controller 106
Memory Module # n
Memory Module #1
108
107
105

Figure 1A

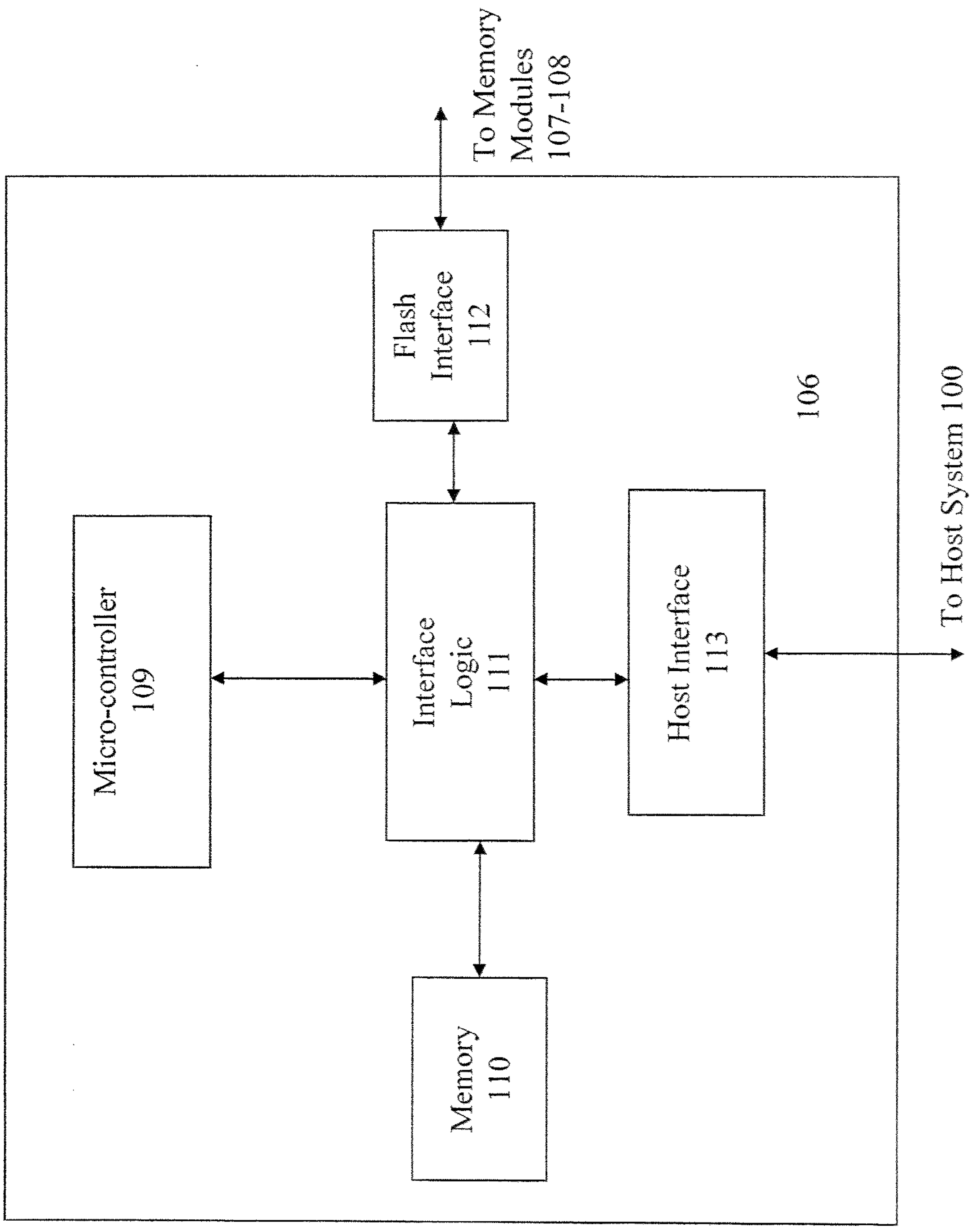

To Memory Modules 107-108
Flash Interface 112
106
Micro-controller 109
Interface Logic 111
Host Interface 113
To Host System 100
Memory 110

Figure 1B

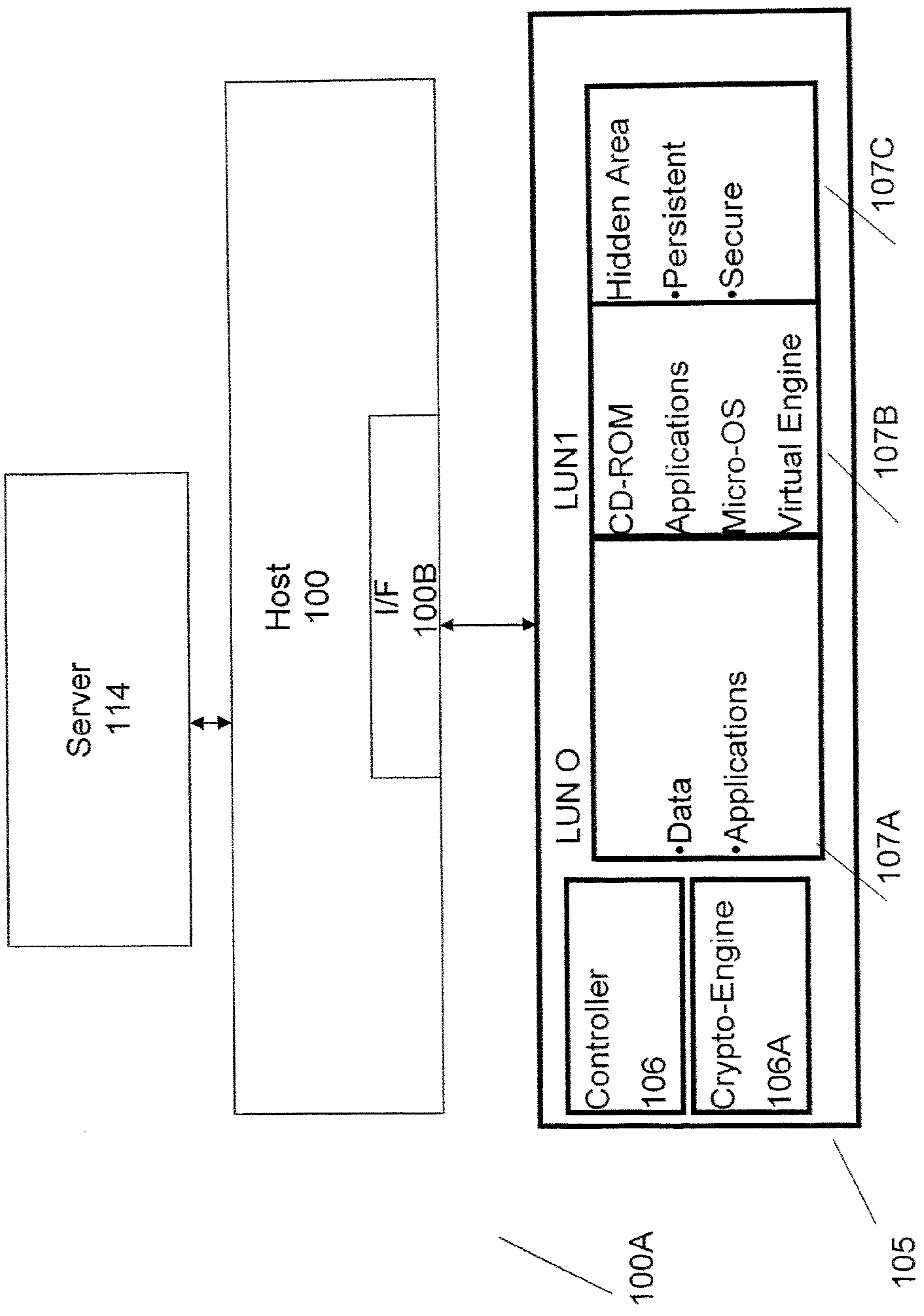

Server
114
Host
100
I/F
100B
LUN O
•Data
•Applications
LUN1
CD-ROM
Applications
Micro-OS
Virtual Engine
Hidden Area
•Persistent
•Secure
Controller
106
Crypto-Engine
106A
107A
107B
107C
105
100A

Figure 1C

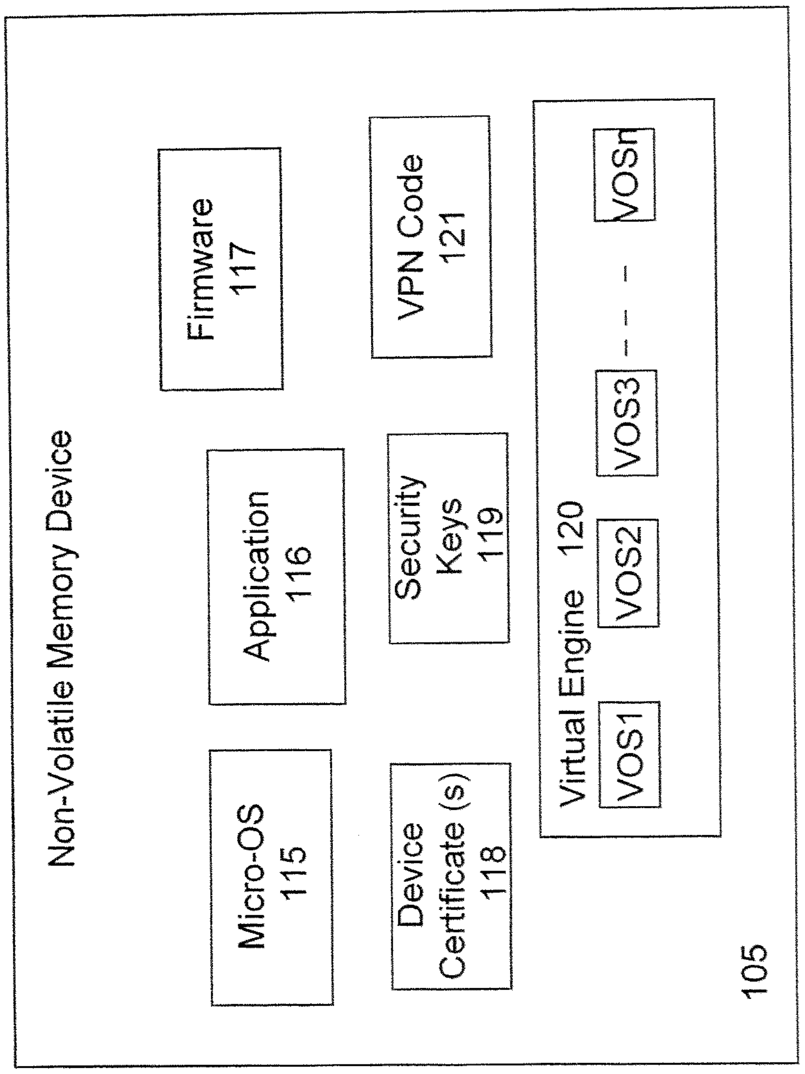

Non-Volatile Memory Device
Micro-OS 115
Application 116
Firmware 117
Device Certificate (s) 118
Security Keys 119
VPN Code 121
Virtual Engine 120
VOS1
VOS2
VOS3
VOSn
105

Figure 1D

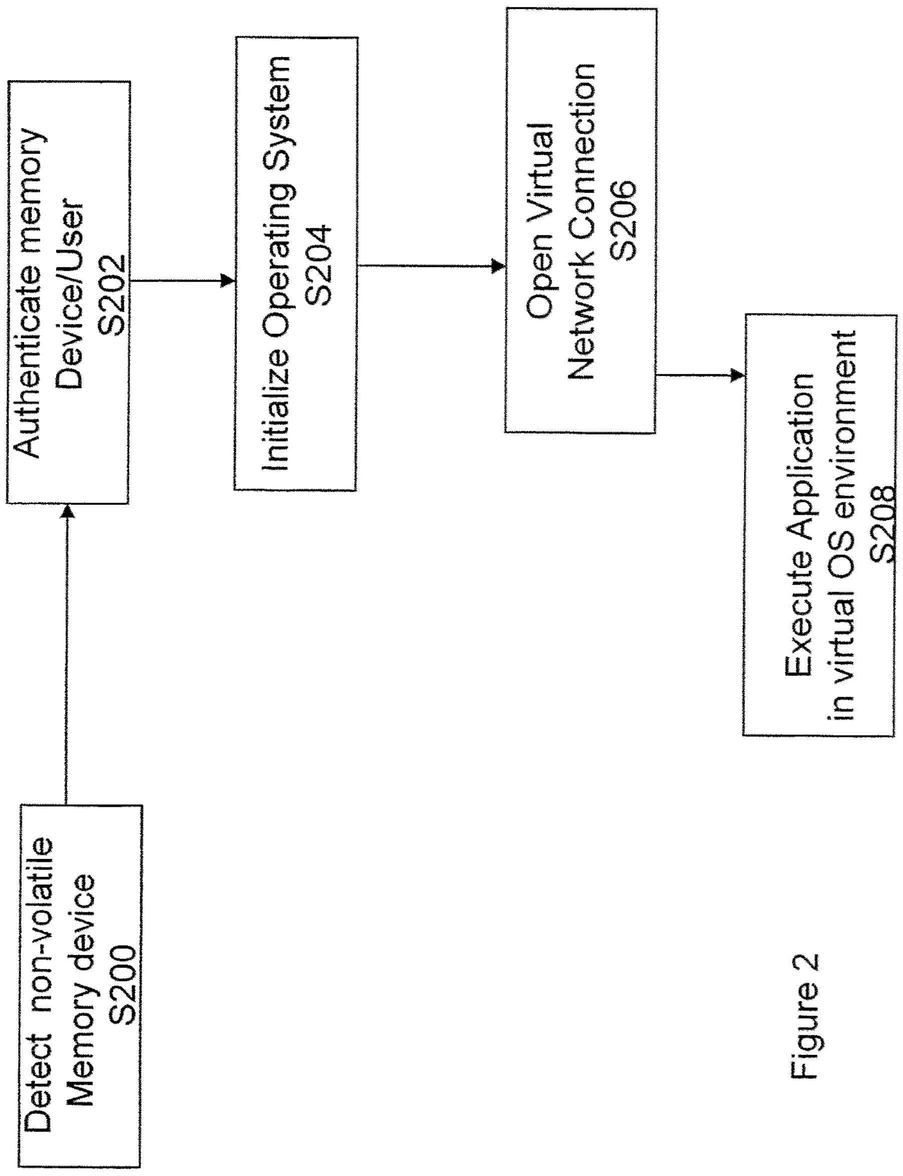

Detect non-volatile Memory device S200
Authenticate memory Device/User S202
Initialize Operating System S204
Open Virtual Network Connection S206
Execute Application in virtual OS environment S208

Figure 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2007/088839

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/445

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 03/079183 A (JAMES BARRY EDMUND [GB]) 25 September 2003 (2003-09-25) page 4, line 29 - page 21, line 10	1-42
X	"U3 Platform 1.0 SDK. Application Deployment Guide (Version 1.0/Revision 4.0)" INTERNET CITATION, [Online] September 2005 (2005-09), XP007903637 Retrieved from the Internet: URL:http://onlineconferencingsystems.com/access_tomorrow/ApplicationDeploymentGuide_1.0r4.pdf> [retrieved on 2007-12-04] the whole document	1-42
A	GB 2 424 095 A (UNISVR GLOBAL INFORMATION TECH [TW]) 13 September 2006 (2006-09-13) page 2, line 5 - page 5, line 12	1-42

-/--

[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 17 April 2008	Date of mailing of the international search report 24/04/2008
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Kusnierczak, Pawel

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2007/088839

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2006/294105 A1 (ROSENAN AVNER [IL] ET AL) 28 December 2006 (2006-12-28) paragraph [0030] - paragraph [0051] -----	1-42

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2007/088839

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 03079183 A	25-09-2003	AU 2003215744 A1	29-09-2003
		EP 1518171 A2	30-03-2005
		GB 2390709 A	14-01-2004
		JP 2005520247 T	07-07-2005
		US 2003212862 A1	13-11-2003
GB 2424095 A	13-09-2006	CN 1648863 A	03-08-2005
		DE 102006009943 A1	23-11-2006
		FR 2889879 A1	23-02-2007
		JP 2006252547 A	21-09-2006
		US 2006218549 A1	28-09-2006
US 2006294105 A1	28-12-2006	NONE	