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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 June 2012 (14.06.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/078690 A1

(51) **International Patent Classification**:
G06F 21/22 (2006.01) *G06F 9/44* (2006.01)

(21) **International Application Number**:
PCT/US2011/063615

(22) **International Filing Date**:
6 December 2011 (06.12.2011)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
12/961,854 7 December 2010 (07.12.2010) US

(71) **Applicant** *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) **Inventors**: **JARRETT, Michael Sean**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **JOHNSON, Joseph Jared**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **KAPOOR, Vishal**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **THOMAS, Anil Francis**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **NEYSTADT, Eugene John**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BATCHELDER, Dennis Scott**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title**: ANTIMALWARE PROTECTION OF VIRTUAL MACHINES

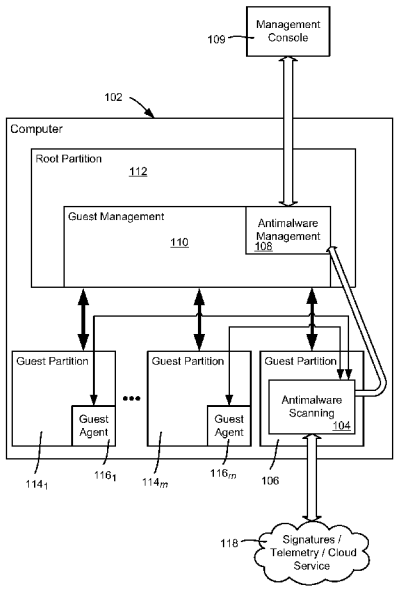


FIG. 1

(57) **Abstract**: The subject disclosure is directed towards protecting virtual machines on guest partitions from malware in a resource-efficient manner. Antimalware software is divided into lightweight agents that run on each malware-protected guest partition, a shared scanning and signature update mechanism, and a management component. Each agent provides the scanning mechanism with files to scan for malware, such as by running a script, and receives results from the scanning mechanism including possible remediation actions to perform. The management component provides the scanning mechanism with access to virtual machine services, such as to pause, resume, snapshot and rollback guest partitions as requested by the scanning mechanism.

WO 2012/078690 A1

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

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

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

ANTIMALWARE PROTECTION OF VIRTUAL MACHINES

BACKGROUND

[0001] A virtual machine (VM) comprises software that executes on a guest partition of a hosting computer system to generally act as if it was an independent physical machine. A computer system may host multiple virtual machines, each running on a virtual machine monitor (VMM), also referred to as a hypervisor, that controls the sharing of the computer system's resources among the virtual machines. Typically virtual machines are run to utilize a physical machine's hardware resources more fully than can be done by conventional programs, and/or to run different operating systems on the same physical machine at the same time.

[0002] Virtual machines are becoming more and more prevalent, and, like any computer system, virtual machines are vulnerable to malicious software, or malware. As such, there exists a need for antimalware products to protect them. This may be accomplished by running traditional antimalware software on each guest partition.

[0003] However, there are drawbacks to operating this way, including that antimalware components are duplicated on each guest partition, whereby each partition consumes network, memory, and CPU resources for the antimalware components. Further, guest antimalware products cannot take advantage of virtual machine services, such as the ability to snapshot or roll back.

SUMMARY

[0004] This Summary is provided to introduce a selection of representative concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used in any way that would limit the scope of the claimed subject matter.

[0005] Briefly, various aspects of the subject matter described herein are directed towards a technology by which an antimalware scanning mechanism (e.g., scanning components) are shared by a plurality of guest partitions that correspond to virtual machines in a virtual machine environment. To this end, guest partitions may include a guest antimalware agent that communicates with the scanning mechanism to use its shared antimalware scanning resources and shared antimalware scanning functionality. For example, the resources of the antimalware scanning mechanism may include antimalware signatures, so that each partition need not maintain its own signatures. The shared

antimalware scanning functionality may comprise (e.g., code that performs) scanning of data such as objects (e.g., files) that are received from the guest antimalware agents. To leverage the capabilities of the guest operating system, the guest antimalware agents may execute instructions provided by the antimalware scanning component to collect additional scanning or telemetry information, or take remedial actions against detected malware.

[0006] In one aspect, a management component is coupled to the antimalware scanning mechanism so as to provide virtual machine management services to the antimalware scanning mechanism. For example, the antimalware scanning mechanism may communicate with the management component to pause a guest partition, resume a guest partition, snapshot a guest partition, or rollback a guest partition to a previous snapshot. The management component may also provide shared orchestration for scanning any guest partition.

[0007] In one implementation, the antimalware scanning mechanism resides in a guest partition that is separate from the other guest partitions that share the antimalware scanning mechanism. In an alternative implementation, the antimalware scanning mechanism resides on the root partition of the virtual machine environment.

[0008] Other advantages may become apparent from the following detailed description when taken in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example and not limited in the accompanying figures in which like reference numerals indicate similar elements and in which:

[0010] FIGURE 1 is a block diagram representing an example virtual machine environment in which an antimalware scanning mechanism runs on a guest partition and is shared by guest partitions via guest agents.

[0011] FIG. 2 is a block diagram representing an example virtual machine environment in which an antimalware scanning mechanism runs on a root partition and is shared by guest partitions via guest agents.

[0012] FIG. 3 is a flow diagram representing example steps for implementing a shared antimalware scanning mechanism in a virtual machine environment.

[0013] FIG. 4 is a flow diagram representing example steps for scanning a guest partition in an offline state.

[0014] FIG. 5 is a block diagram representing exemplary non-limiting networked environments in which various embodiments described herein can be implemented.

[0015] FIG. 6 is a block diagram representing an exemplary non-limiting computing system or operating environment in which one or more aspects of various embodiments described herein can be implemented.

DETAILED DESCRIPTION

[0016] Various aspects of the technology described herein are generally directed towards an efficient way to protect virtual machines from malware, in which each virtual machine runs on a guest partition. In one implementation, the antimalware software is divided into separate components, including a lightweight agent, a shared scanning and signature update component, and a management component. An agent runs on supported guest partitions and provides real-time and online operating system interaction services. The scanning and signature update component, which may reside on a separate guest partition or the root partition, is configured to be used by each of the other guest agents. The management component provides centralized reporting and access to virtual machine services, and, for example, may reside on the root partition.

[0017] As will be understood, the technology described herein provides centralized anti-malware capabilities for multiple guest virtual machines in a virtual machine environment via the shared scanning component. This facilitates real-time antimalware protection by directing scan requests to the shared scanning component, including possibly on-demand scans and remediation on guest partitions, e.g., through the use of simple scripts provided by shared scanning component.

[0018] Moreover, the management component, by running on the root partition, provides pause / resume / snapshot / rollback and inspection services for the scanning component. This facilitates on-demand scans and remediation on guest partitions by the scanning component without the direct cooperation of the guest agent (e.g., if the guest agent is compromised or unavailable), while the guest partitions are paused via the management component, or while the guest is not running (offline), which may be used to detect malware that has stealth or protection capabilities from the perspective of the guest agent.

[0019] It should be understood that any of the examples herein are non-limiting. For example, while scanning of objects such as files is described, security evaluation of other content, such as for network intrusion protection, data leakage, guest verification and so forth may benefit from the technology described herein. As such, the present invention is not limited to any particular embodiments, aspects, concepts, structures, functionalities or examples described herein. Rather, any of the embodiments, aspects, concepts, structures, functionalities or examples described herein are non-limiting, and the present invention

may be used various ways that provide benefits and advantages in virtual computing and/or protection against malware in general.

[0020] FIG. 1 shows example components of a computer system 102 configured with virtual machine distributed antimalware. The components in this exemplified implementation include a scanning mechanism 104 (comprising one or more antimalware components) residing in a dedicated scanning guest partition 106, and an antimalware management component 108 (which may be interfaced with via a local or remote management console 109) as part of guest management services 110 on a root partition 112. Note that FIG. 1 is only a non-limiting example of a possible deployment, and others are feasible, including the example deployment represented in FIG. 2.

[0021] In general, the root partition 112 comprises a running operating system environment from which the state of other virtual machine guest partitions 106 and 114_1-114_m may be controlled. Each guest partition 106 and 114_1-114_m corresponds to any virtual machine or machine partition that is not the root partition 112.

[0022] Each guest partition 114_1-114_m for which real-time antimalware support is provided includes a respective guest agent 116_1-116_m, comprising software that provides real-time protection services for that guest partition, possibly along with other services. Each guest agent 116_1-116_m is specific to the operating system being run on its respective guest partition 114_1-114_m. Note that although not shown in FIG. 1, the guest partition 106 containing the antimalware scanning mechanism 104 also may include such a guest agent.

[0023] In order to protect guest agents from being tampered with by malicious code that successfully compromises a guest partition, a "privileged" protection component, (e.g., running inside the root partition or a dedicated security virtual machine) may monitor the integrity of the guest agent and other relevant components of guest virtual machine.

[0024] Each guest agent (e.g., 116_1) provides real-time system monitoring with the capability to detect and block access to objects. To this end, the guest agent 116_1 communicates bi-directionally (e.g., at high speed) with the scanning guest partition 106. Note that any communication mechanism is feasible, such as through the root partition, through a simulated network interface and so forth; however in one implementation communication is over a high-speed bus or shared memory block that exists between the partitions. Any guest agent (e.g., 116_1) may be configured with a user interface, such as if guest partitions are often used interactively. Such a user interface may provide an interactive user of the guest visibility into the current security state of the guest, or allow

an interactive user to request that the antimalware component begin a specific on-demand scan.

[0025] Each guest agent such as the agent 116_1 may be (optionally) configured with the ability to run simple scripts, e.g., provided by the scanning guest partition 106 over a suitable bi-directional communication mechanism as generally described herein. Configuring the agents with the ability to run scripts avoids the need for the agent to be coded with its own logic with respect to making decisions on what to scan, what to do when malware is detected, and the like. For example, to scan a guest partition's files, a script may request that the agent feed its files or some subset thereof to the antimalware scanning mechanism 104, which scans them, may perform some needed remediation such as to clean a file, and may return results of the scanning / remediation to the agent, including a script with actions to take, e.g., files to delete or quarantine. Such scripts may include the ability to touch resources (e.g., triggering real-time transport protocol capabilities), and also to modify or terminate / delete resources.

[0026] The antimalware scanning mechanism 104 performs scanning, remediation, signature update operations, and in general enforces antimalware aspects of security policy with the cooperation of the guest agent and/or the management components. In general, the antimalware scanning mechanism 104 provides antimalware scanning as a service to the guest agents 116_1-116_m. Further, the antimalware scanning mechanism 104 also may initiate scanning or remedial actions against a guest partition, such as cooperatively using services of the guest agent, or alternatively without the guest partition's knowledge or consent, (e.g., while the guest partition is paused / offline), through the support of the management component 108.

[0027] With respect to real-time monitoring, the antimalware scanning mechanism 104 communicates bi-directionally with the guest agents 116_1-116_m, including in one implementation to identify any malware in content transmitted from the guest partitions. In general, for real-time monitoring, each agent feeds data such as an object set (comprising one or more objects such as files, registry data, processes or the like) to the antimalware scanning mechanism 104, which then evaluates the data against antimalware signatures, and returns a result, possibly taking a remedial action (e.g., cleaning the object) and/or including scripted instructions for the agent to take a remedial action (e.g., remove a file or quarantine a file), such as via a script.

[0028] Note that it is feasible to provide some or all of the guest agents 116_1-116_m with some scanning capabilities / intelligence themselves, rather than have them simply

forward objects and receive and act on scripted results. For example, if a particular virus is currently widespread, the antimalware scanning mechanism 104 may provide the guest agent with a subset of signatures to look for with respect to a given file or file type, whereby the guest agent can handle scanning or remediation itself in the event such a file is encountered. This may be via a script, and/or possibly to some extent by coding basic scanning functionality into the agent.

[0029] Another benefit of the shared scanning mechanism 104 is that signatures as well as other scanning components may be updated, without experiencing significant scanning downtime. Further, information may be uploaded to a remote location, such as data, reports and sample object submission for subsequent analysis and so forth. This aspect of the shared scanning mechanism 104 is represented in FIG. 1 via signatures / telemetry / cloud service118. Note that the other guest partitions 114_1-114_m need not have access to the internet, for example, yet still benefit from the update and telemetry access of the shared guest partition 106.

[0030] Moreover, the remote access capabilities of the antimalware scanning mechanism 104 may include communicating with a shared "cloud scanning" service for a decision (infected or clean) on suspicious content not yet matched by signatures. The antimalware scanning mechanism 104 may make such queries on behalf of multiple guests, such that guests get the benefit of the cloud service without needing Internet access directly. Also, the antimalware scanning mechanism 104 may cache the results, so that it only has to make one request to the cloud service even if multiple guests are seeing the same suspicious content.

[0031] The antimalware scanning mechanism 104 also has a communication link with the antimalware management component 108. As described above, this provides the antimalware scanning mechanism 104 with the ability to integrate antimalware scanning with virtual machine management capabilities. For example, the antimalware scanning mechanism 104 may request that the management component 108 pause a guest partition, thereby providing the scanning mechanism 104 with the ability to scan a guest partition (or a snapshot thereof) offline, and/or with the ability to manipulate offline guest partition (or a snapshot thereof) to remove malware. Offline scanning may be performed if a serious problem is detected, that is, reactively, such as if the guest has crashed, or if an operating system file that cannot be cleaned is infected, but cannot be replaced online because the file is needed to run the operating system. Offline scanning also may be performed proactively, e.g., before staring a guest partition; if a partition is known to be

free of infections at startup, but then becomes infected while running, a rapid diagnosis may be made. This integration capability also provides the ability to perform scans and remediation on guests not supported by a guest agent.

[0032] The management component 108 thus comprises a component with access to the virtual machine management services 110. In the exemplified implementation of FIG. 1, the management component 108 is part of the virtual machine management services 110, and communicates with the scanning mechanism 104, but not with the individual guests agents 116_1-116_m. Among its operations, the management component 108 may monitor the scanning component's heartbeat, such that if the scanning component and/or scanning partition become unresponsive, the management component 108 may restart the scanning component or scanning partition, and /or raise an alert. Note that some malware actively tries to disable antimalware protection, and having a shared scanning service monitoring the guests helps in making the agents running in the guests tamper-resistant.

[0033] Further, the management component 108 is able to act as a centralized collection point and intermediary for communication to and from the security management console 109. The management component 108 may provide the scanning partition 106 with the online ability to manipulate guest partitions, including the ability to stop an infected guest, or revert a guest partition to a snapshot. The management component 108 may provide the scanning mechanism 104 with the ability to manipulate a guest partition offline.

[0034] Note that although the management component 108 has access to the virtual machine management services 110, along with the ability to coordinate with the scanning mechanism 104, and the ability to report (and potentially be reconfigured) by any central security management services, the management component 108 is a distinct component from virtual machine management, antimalware management, and the scanning mechanism 104. The management component 108 need not even be on the same computer system as these components, as long as they can communicate, e.g., over network connections. However, deploying the management component 108 on the root partition 112 (as exemplified in FIG. 1) generally reduces the latency of communication with the scanning partition 106 and the virtual machine management services 110.

[0035] As shown in FIG. 1, the scanning mechanism 104 may reside in the dedicated scanning guest partition 106; (note that "dedicated" as used herein refers to having resources reserved for scanning functionality, and does not mean that such a partition cannot also be used for other purposes). This implementation provides a security boundary between the antimalware scanning mechanism 104 (whose components are often

targeted by security vulnerabilities) and the root partition 112. Although not explicitly shown, multiple scanning partitions can be used, such as for failover capabilities, e.g., if one scanning partition fails, another one may quickly resume and take its place. Load balancing and/or workload distribution is another possible use of multiple scanning partitions, e.g., in the event that a single scanning partition is not able to meet scanning demands.

[0036] In an alternative implementation, as generally represented in FIG. 2 (where like components are labeled by 2xx instead of 1xx in FIG. 1), a scanning component 204 (or more than one) may be deployed in the root partition 212. This has the potential for significant optimization, saving the overhead of an entire guest partition / operating system, while providing direct access to management components and stored guest partition state. However, this requires the scanning component to be available for the operating system deployed on the root partition 212, and reduces the protection of the root partition 212 from potential exploits from content found in guest partitions.

[0037] By way of summary, FIG. 3 shows example steps that may be taken to provide malware protection using the above-described components. Steps 302 and 304 represent running the guest partition and an antimalware agent, and running the shared antimalware scanning mechanism, respectively; note that while only one agent is shown, it is understood that similar steps are performed with each other of the plurality of agents that are run. Further note that steps 302 and 304 are performed by virtual machine management in an implementation as in FIG. 1, however step 304 may be performed by other root partition software in an implementation as in FIG. 2 where the shared antimalware scanning mechanism is not run in a guest partition.

[0038] Steps 306-318 represent example actions performed by and from the perspective of the antimalware scanning mechanism, in which the agent relies on the antimalware scanning mechanism for the scan. Step 306 represents providing information such as a script to the agent. In a scan that is not a real-time scan, the script may identify what files, folders, or other operating system resources to provide for scanning, what file types to provide, and so forth. In a real time scan, the information may be an instruction or the like informing the agent that scanning is turned on, and that the agent is to provide each appropriate object to the antimalware scanning mechanism for real time scanning.

[0039] Step 308 represents the agent providing data such as an object set (e.g., one or more files, registry data or other data blobs) to the antimalware scanning mechanism, which receives it for scanning, as represented by step 310. If the data contains malware,

step 312 takes action with respect to that data as represented by step 314. As described above, this may be by performing remediation in the antimalware scanning mechanism, e.g., cleaning data before returning it, and/or constructing a result that instructs the agent to take some remediation action (e.g., remove a file, quarantine a file, write a cleaned file back).

[0040] Note that sometimes, malware cannot be cleaned from a compromised virtual machine. In a computer system that is not configured as a virtual environment, human operator intervention is needed, usually by reinstalling a machine. However in a virtual environment, a management component can automatically (possibly after asking for administrator approval) restore the virtual machine to a previous known good snapshot, or by rebuilding a virtual machine image.

[0041] Step 316 returns the result to the agent, which may include a script of one or more actions for the agent to perform, and/or a request for the next set of data. Step 318 represents ending the scanning process if the scan is complete, or continuing the scanning process if more scanning is needed, either because there is at least one more set of data to scan, or because the scan is a real time monitoring operation, which continues indefinitely by waiting for the next set of data.

[0042] FIG. 4 is an example of offline scanning of a guest partition, beginning at step 402 where the management component 108 receives a request (e.g., from the antimalware scanning mechanism 104) to move the guest partition into an offline state (step 404). Once in the offline state, Steps 406-412 represent example actions performed by and from the perspective of the antimalware scanning mechanism 104. If during the scan (step 406) malware is encountered at step 408, remediation is performed at step 410. Note that the agent is offline, and thus cannot participate in remediation, which may include cleaning, removing or quarantining a file, as well as possibly replacing a corrupted operating system file that cannot be replaced while in an online state. Also note that step 410 represents saving the results of the malware remediation, for analysis purposes, for informing the guest partition what occurred, to upload telemetry data, and so forth. Step 412 repeats the scanning until it is complete, e.g., all appropriate file system files have been scanned, for example.

[0043] When complete, steps 414 and 416 are performed by the management component 108 to restore the guest partition to an online state. e.g., as requested by the antimalware scanning mechanism 104.

EXEMPLARY NETWORKED AND DISTRIBUTED ENVIRONMENTS

[0044] One of ordinary skill in the art can appreciate that the various embodiments and methods described herein can be implemented in connection with any computer or other client or server device, which can be deployed as part of a computer network or in a distributed computing environment, and can be connected to any kind of data store or stores. In this regard, the various embodiments described herein can be implemented in any computer system or environment having any number of memory or storage units, and any number of applications and processes occurring across any number of storage units. This includes, but is not limited to, an environment with server computers and client computers deployed in a network environment or a distributed computing environment, having remote or local storage.

[0045] Distributed computing provides sharing of computer resources and services by communicative exchange among computing devices and systems. These resources and services include the exchange of information, cache storage and disk storage for objects, such as files. These resources and services also include the sharing of processing power across multiple processing units for load balancing, expansion of resources, specialization of processing, and the like. Distributed computing takes advantage of network connectivity, allowing clients to leverage their collective power to benefit the entire enterprise. In this regard, a variety of devices may have applications, objects or resources that may participate in the resource management mechanisms as described for various embodiments of the subject disclosure.

[0046] FIG. 5 provides a schematic diagram of an exemplary networked or distributed computing environment. The distributed computing environment comprises computing objects 510, 512, etc., and computing objects or devices 520, 522, 524, 526, 528, etc., which may include programs, methods, data stores, programmable logic, etc. as represented by example applications 530, 532, 534, 536, 538. It can be appreciated that computing objects 510, 512, etc. and computing objects or devices 520, 522, 524, 526, 528, etc. may comprise different devices, such as personal digital assistants (PDAs), audio/video devices, mobile phones, MP3 players, personal computers, laptops, etc.

[0047] Each computing object 510, 512, etc. and computing objects or devices 520, 522, 524, 526, 528, etc. can communicate with one or more other computing objects 510, 512, etc. and computing objects or devices 520, 522, 524, 526, 528, etc. by way of the communications network 540, either directly or indirectly. Even though illustrated as a single element in FIG. 5, communications network 540 may comprise other computing

objects and computing devices that provide services to the system of FIG. 5, and/or may represent multiple interconnected networks, which are not shown. Each computing object 510, 512, etc. or computing object or device 520, 522, 524, 526, 528, etc. can also contain an application, such as applications 530, 532, 534, 536, 538, that might make use of an API, or other object, software, firmware and/or hardware, suitable for communication with or implementation of the application provided in accordance with various embodiments of the subject disclosure.

[0048] There are a variety of systems, components, and network configurations that support distributed computing environments. For example, computing systems can be connected together by wired or wireless systems, by local networks or widely distributed networks. Currently, many networks are coupled to the Internet, which provides an infrastructure for widely distributed computing and encompasses many different networks, though any network infrastructure can be used for exemplary communications made incident to the systems as described in various embodiments.

[0049] Thus, a host of network topologies and network infrastructures, such as client/server, peer-to-peer, or hybrid architectures, can be utilized. The "client" is a member of a class or group that uses the services of another class or group to which it is not related. A client can be a process, e.g., roughly a set of instructions or tasks, that requests a service provided by another program or process. The client process utilizes the requested service without having to "know" any working details about the other program or the service itself.

[0050] In a client / server architecture, particularly a networked system, a client is usually a computer that accesses shared network resources provided by another computer, e.g., a server. In the illustration of FIG. 5, as a non-limiting example, computing objects or devices 520, 522, 524, 526, 528, etc. can be thought of as clients and computing objects 510, 512, etc. can be thought of as servers where computing objects 510, 512, etc., acting as servers provide data services, such as receiving data from client computing objects or devices 520, 522, 524, 526, 528, etc., storing of data, processing of data, transmitting data to client computing objects or devices 520, 522, 524, 526, 528, etc., although any computer can be considered a client, a server, or both, depending on the circumstances.

[0051] A server is typically a remote computer system accessible over a remote or local network, such as the Internet or wireless network infrastructures. The client process may be active in a first computer system, and the server process may be active in a second computer system, communicating with one another over a communications medium, thus

providing distributed functionality and allowing multiple clients to take advantage of the information-gathering capabilities of the server.

[0052] In a network environment in which the communications network 540 or bus is the Internet, for example, the computing objects 510, 512, etc. can be Web servers with which other computing objects or devices 520, 522, 524, 526, 528, etc. communicate via any of a number of known protocols, such as the hypertext transfer protocol (HTTP). Computing objects 510, 512, etc. acting as servers may also serve as clients, e.g., computing objects or devices 520, 522, 524, 526, 528, etc., as may be characteristic of a distributed computing environment.

EXEMPLARY COMPUTING DEVICE

[0053] As mentioned, advantageously, the techniques described herein can be applied to any device. It can be understood, therefore, that handheld, portable and other computing devices and computing objects of all kinds are contemplated for use in connection with the various embodiments. Accordingly, the below general purpose remote computer described below in FIG. 6 is but one example of a computing device.

[0054] Embodiments can partly be implemented via an operating system, for use by a developer of services for a device or object, and/or included within application software that operates to perform one or more functional aspects of the various embodiments described herein. Software may be described in the general context of computer executable instructions, such as program modules, being executed by one or more computers, such as client workstations, servers or other devices. Those skilled in the art will appreciate that computer systems have a variety of configurations and protocols that can be used to communicate data, and thus, no particular configuration or protocol is considered limiting.

[0055] FIG. 6 thus illustrates an example of a suitable computing system environment 600 in which one or aspects of the embodiments described herein can be implemented, although as made clear above, the computing system environment 600 is only one example of a suitable computing environment and is not intended to suggest any limitation as to scope of use or functionality. In addition, the computing system environment 600 is not intended to be interpreted as having any dependency relating to any one or combination of components illustrated in the exemplary computing system environment 600.

[0056] With reference to FIG. 6, an exemplary remote device for implementing one or more embodiments includes a general purpose computing device in the form of a

computer 610. Components of computer 610 may include, but are not limited to, a processing unit 620, a system memory 630, and a system bus 622 that couples various system components including the system memory to the processing unit 620.

[0057] Computer 610 typically includes a variety of computer readable media and can be any available media that can be accessed by computer 610. The system memory 630 may include computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) and/or random access memory (RAM). By way of example, and not limitation, system memory 630 may also include an operating system, application programs, other program modules, and program data.

[0058] A user can enter commands and information into the computer 610 through input devices 640. A monitor or other type of display device is also connected to the system bus 622 via an interface, such as output interface 650. In addition to a monitor, computers can also include other peripheral output devices such as speakers and a printer, which may be connected through output interface 650.

[0059] The computer 610 may operate in a networked or distributed environment using logical connections to one or more other remote computers, such as remote computer 670. The remote computer 670 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, or any other remote media consumption or transmission device, and may include any or all of the elements described above relative to the computer 610. The logical connections depicted in Fig. 6 include a network 672, such local area network (LAN) or a wide area network (WAN), but may also include other networks/buses. Such networking environments are commonplace in homes, offices, enterprise-wide computer networks, intranets and the Internet.

[0060] As mentioned above, while exemplary embodiments have been described in connection with various computing devices and network architectures, the underlying concepts may be applied to any network system and any computing device or system in which it is desirable to improve efficiency of resource usage.

[0061] Also, there are multiple ways to implement the same or similar functionality, e.g., an appropriate API, tool kit, driver code, operating system, control, standalone or downloadable software object, etc. which enables applications and services to take advantage of the techniques provided herein. Thus, embodiments herein are contemplated from the standpoint of an API (or other software object), as well as from a software or hardware object that implements one or more embodiments as described herein. Thus,

various embodiments described herein can have aspects that are wholly in hardware, partly in hardware and partly in software, as well as in software.

[0062] The word "exemplary" is used herein to mean serving as an example, instance, or illustration. For the avoidance of doubt, the subject matter disclosed herein is not limited by such examples. In addition, any aspect or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects or designs, nor is it meant to preclude equivalent exemplary structures and techniques known to those of ordinary skill in the art. Furthermore, to the extent that the terms "includes," "has," "contains," and other similar words are used, for the avoidance of doubt, such terms are intended to be inclusive in a manner similar to the term "comprising" as an open transition word without precluding any additional or other elements when employed in a claim.

[0063] As mentioned, the various techniques described herein may be implemented in connection with hardware or software or, where appropriate, with a combination of both. As used herein, the terms "component," "module," "system" and the like are likewise intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on computer and the computer can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

[0064] The aforementioned systems have been described with respect to interaction between several components. It can be appreciated that such systems and components can include those components or specified sub-components, some of the specified components or sub-components, and/or additional components, and according to various permutations and combinations of the foregoing. Sub-components can also be implemented as components communicatively coupled to other components rather than included within parent components (hierarchical). Additionally, it can be noted that one or more components may be combined into a single component providing aggregate functionality or divided into several separate sub-components, and that any one or more middle layers, such as a management layer, may be provided to communicatively couple to such sub-components in order to provide integrated functionality. Any components described herein may also interact with one or more other components not specifically described herein but generally known by those of skill in the art.

[0065] In view of the exemplary systems described herein, methodologies that may be implemented in accordance with the described subject matter can also be appreciated with reference to the flowcharts of the various figures. While for purposes of simplicity of explanation, the methodologies are shown and described as a series of blocks, it is to be understood and appreciated that the various embodiments are not limited by the order of the blocks, as some blocks may occur in different orders and/or concurrently with other blocks from what is depicted and described herein. Where non-sequential, or branched, flow is illustrated via flowchart, it can be appreciated that various other branches, flow paths, and orders of the blocks, may be implemented which achieve the same or a similar result. Moreover, some illustrated blocks are optional in implementing the methodologies described hereinafter.

CONCLUSION

[0066] While the invention is susceptible to various modifications and alternative constructions, certain illustrated embodiments thereof are shown in the drawings and have been described above in detail. It should be understood, however, that there is no intention to limit the invention to the specific forms disclosed, but on the contrary, the intention is to cover all modifications, alternative constructions, and equivalents falling within the spirit and scope of the invention.

[0067] In addition to the various embodiments described herein, it is to be understood that other similar embodiments can be used or modifications and additions can be made to the described embodiment(s) for performing the same or equivalent function of the corresponding embodiment(s) without deviating therefrom. Still further, multiple processing chips or multiple devices can share the performance of one or more functions described herein, and similarly, storage can be effected across a plurality of devices. Accordingly, the invention is not to be limited to any single embodiment, but rather is to be construed in breadth, spirit and scope in accordance with the appended claims.

CLAIMS

1. In a computing environment, a system, comprising:

a plurality of guest partitions corresponding to virtual machines in a virtual machine environment, each guest partition including a guest antimalware agent; and

an antimalware scanning mechanism comprising one or more antimalware-related components, the antimalware scanning mechanism configured to communicate with the guest antimalware agents on the guest partitions, the antimalware scanning mechanism further configured to provide shared antimalware scanning resources and shared antimalware scanning functionality to the guest partitions via the guest antimalware agents.

2. The system of claim 1 further comprising, a management component configured to protect the guest agents, the management component residing in a root partition and further configured to suspend, resume, recover and rebuild virtual machines to enable scanning and remediate infections, or a management component coupled to the antimalware scanning mechanism, the management component configured to provide virtual machine management services to the antimalware scanning mechanism, or both.

3. The system of claim 2 wherein the antimalware scanning mechanism communicates with the management component to use the management services to pause a guest partition, resume a guest partition, snapshot a guest partition, rollback a guest partition to a previous known good snapshot, rebuild a virtual machine image, or place a guest partition into an offline state for scanning by the antimalware scanning mechanism.

4. The system of claim 1 wherein the antimalware scanning mechanism is further configured to obtain signature updates from a remote data location, upload telemetry data to a remote data location, or to communicate with a cloud service with respect to obtaining a decision on suspicious content, or any combination of being further configured to obtain signature updates from a remote data location, upload telemetry data to a remote data location, or to communicate with a cloud service with respect to obtaining a decision on suspicious content.

5. In a computing environment, a method performed at least in part on at least one processor, comprising:

running a plurality of guest partitions in a virtual machine environment; and

running a shared orchestration mechanism to scan or restore a guest partition.

6. The method of claim 5 wherein running the shared orchestration mechanism comprises:

placing a guest partition into an offline state, scanning the offline guest partition while in the offline state, and taking any needed remedial actions against the offline state, or

communicating between the shared orchestration mechanism and a scanning component to restore a guest partition to a prior state, or

placing a guest partition into an offline state, scanning the offline guest partition while in the offline state, and taking any needed remedial actions against the offline state, and communicating between the shared orchestration mechanism and a scanning component to restore a guest partition to a prior state .

7. The method of claim 5 further comprising, running a guest antimalware agent on a guest partition, receiving data provided by the guest agent at a scanning mechanism, scanning the data at the scanning mechanism, and returning information to the guest agent corresponding to a scanning result, wherein receiving the data and scanning the data comprise performing a real-time monitoring operation, or an offline operation.

8. The method of claim 5 further comprising, providing instructions from the scanning mechanism to the guest antimalware agent for the guest antimalware agent to execute, including instructions requesting at least one object to scan, or including at least one instruction specifying a remediation action for the guest antimalware agent to perform, or both instructions requesting at least one object to scan, and at least one instruction specifying a remediation action for the guest antimalware agent to perform.

9. One or more computer-readable media having computer-executable instructions, which when executed perform steps, comprising:

receiving a first object set comprising one or more objects at a shared antimalware scanning mechanism from a first guest agent that runs in a first guest partition of a virtual machine environment;

performing antimalware scanning of the first object set at the shared scanning mechanism, and returning information corresponding to a result of the scanning to the first guest agent;

receiving a second object set comprising one or more objects at the shared antimalware scanning mechanism from a second guest agent that runs in a second guest partition of a virtual machine environment;

performing antimalware scanning of the second object set at the shared scanning mechanism and returning information corresponding to a result of the scanning to the second guest agent.

10. The one or more computer-readable media of claim 9 having further computer-executable instructions comprising, performing remediation to at least one object at the shared scanning mechanism, or constructing a result at the shared scanning mechanism that instructs the first agent to perform remediation to at least one object, or both performing remediation to at least one object at the shared scanning mechanism and constructing a result at the shared scanning mechanism that instructs the first agent to perform remediation to at least one object.

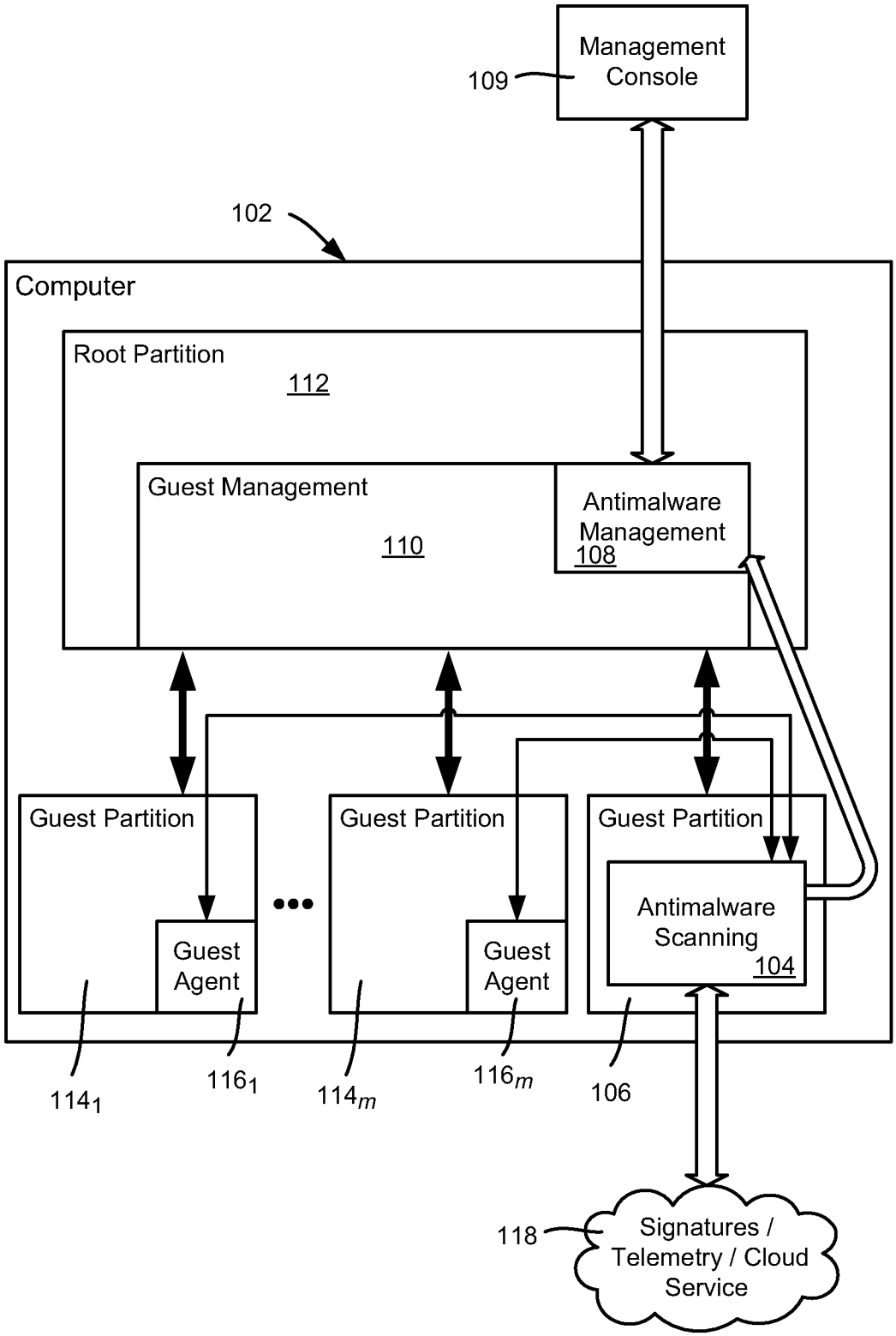

Management Console
109
102
Computer
Root Partition
112
Guest Management
110
Antimalware Management
108
Guest Partition
Guest Partition
Guest Partition
Guest Agent
Guest Agent
Antimalware Scanning
104
114_1
116_1
114_m
116_m
106
118
Signatures / Telemetry / Cloud Service

FIG. 1

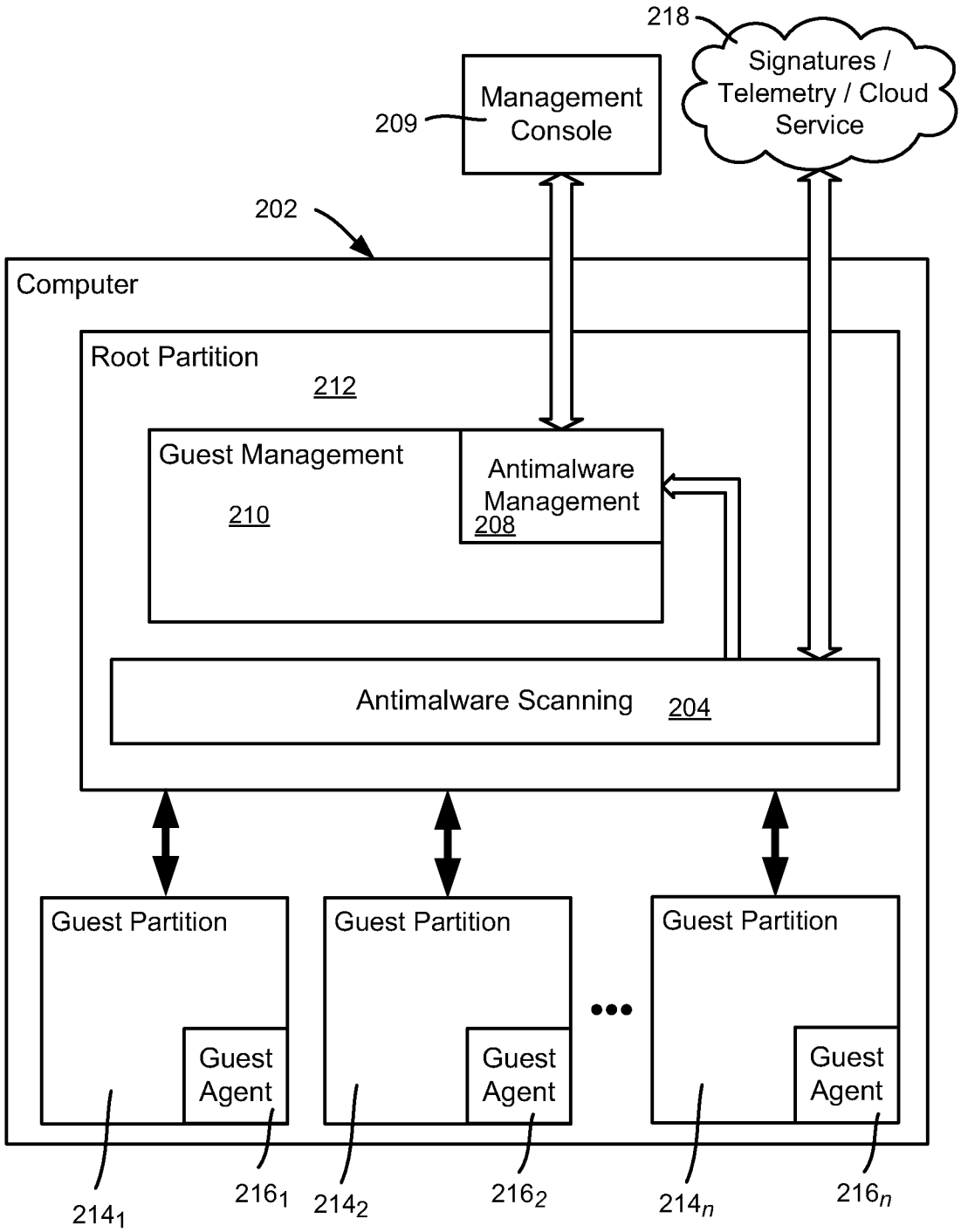

218
Signatures / Telemetry / Cloud Service
209
Management Console
202
Computer
Root Partition
212
Guest Management
210
Antimalware Management
208
Antimalware Scanning
204
Guest Partition
Guest Agent
Guest Partition
Guest Agent
Guest Partition
Guest Agent
214_1
216_1
214_2
216_2
214_n
216_n

FIG. 2

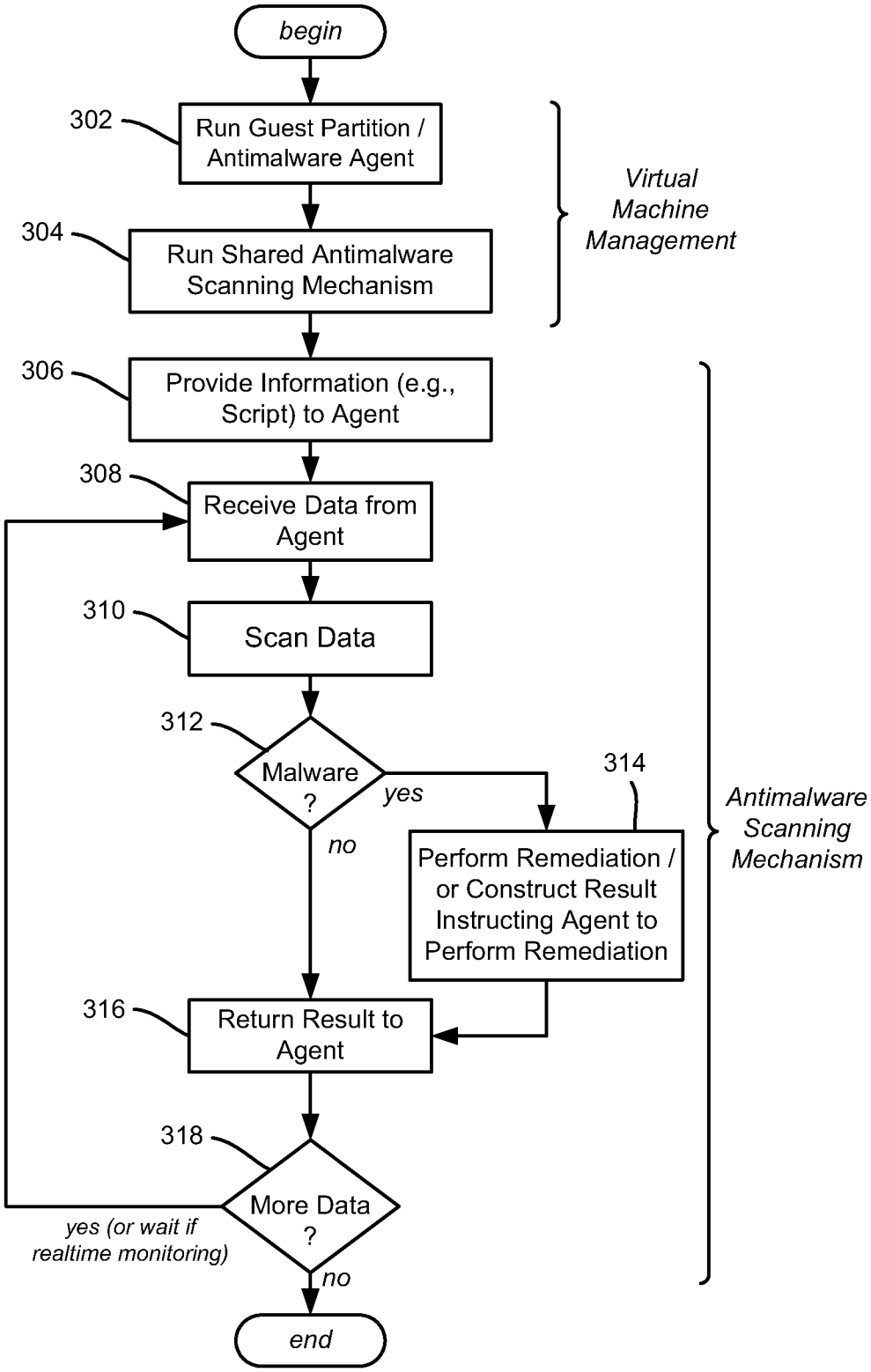

begin
302
Run Guest Partition / Antimalware Agent
304
Run Shared Antimalware Scanning Mechanism
Virtual Machine Management
306
Provide Information (e.g., Script) to Agent
308
Receive Data from Agent
310
Scan Data
312
Malware ?
yes
no
314
Perform Remediation / or Construct Result Instructing Agent to Perform Remediation
Antimalware Scanning Mechanism
316
Return Result to Agent
318
More Data ?
yes (or wait if realtime monitoring)
no
end

FIG. 3

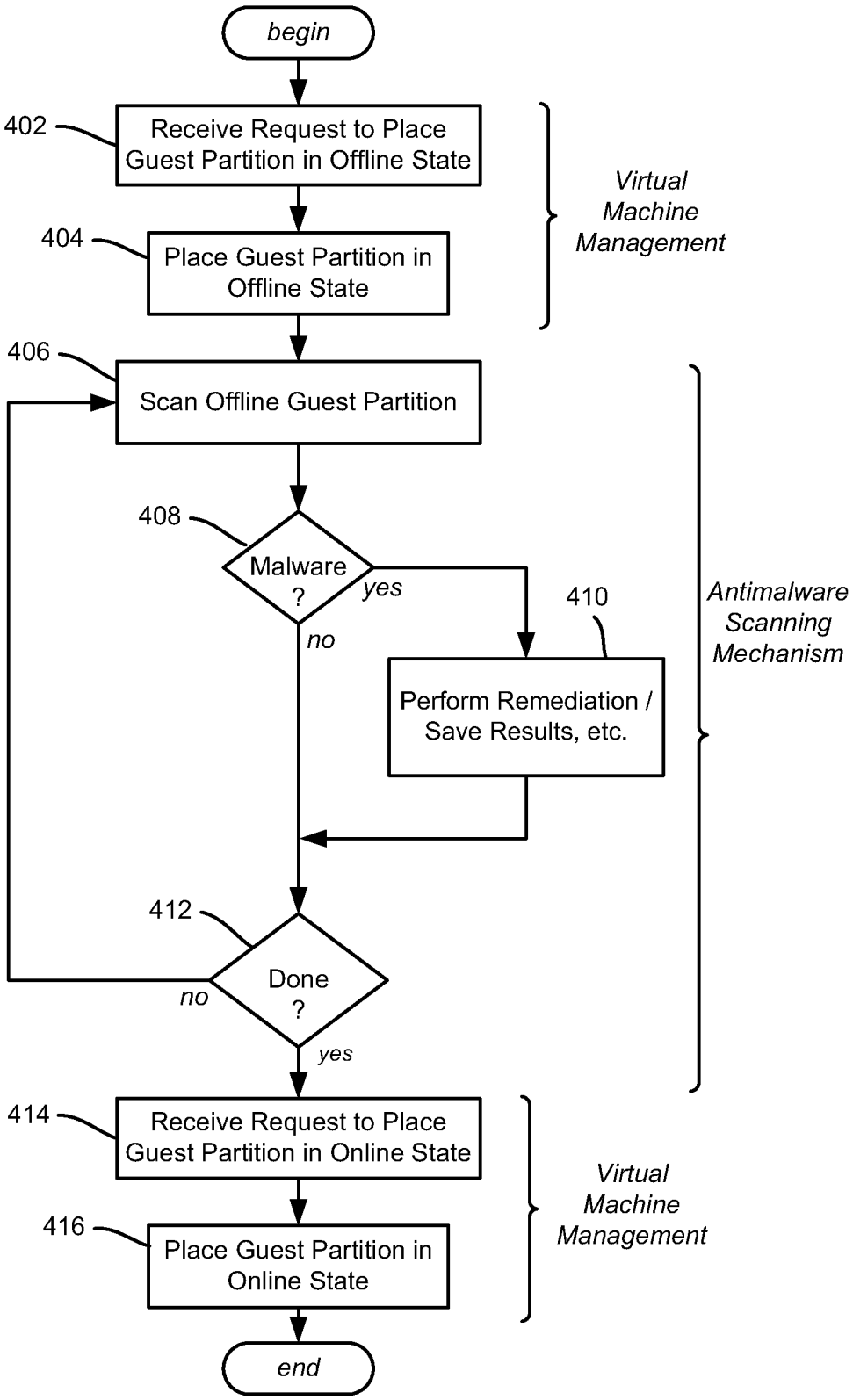

begin
402
Receive Request to Place Guest Partition in Offline State
404
Place Guest Partition in Offline State
Virtual Machine Management
406
Scan Offline Guest Partition
408
Malware ?
yes
no
410
Perform Remediation / Save Results, etc.
Antimalware Scanning Mechanism
412
Done ?
no
yes
414
Receive Request to Place Guest Partition in Online State
416
Place Guest Partition in Online State
Virtual Machine Management
end

FIG. 4

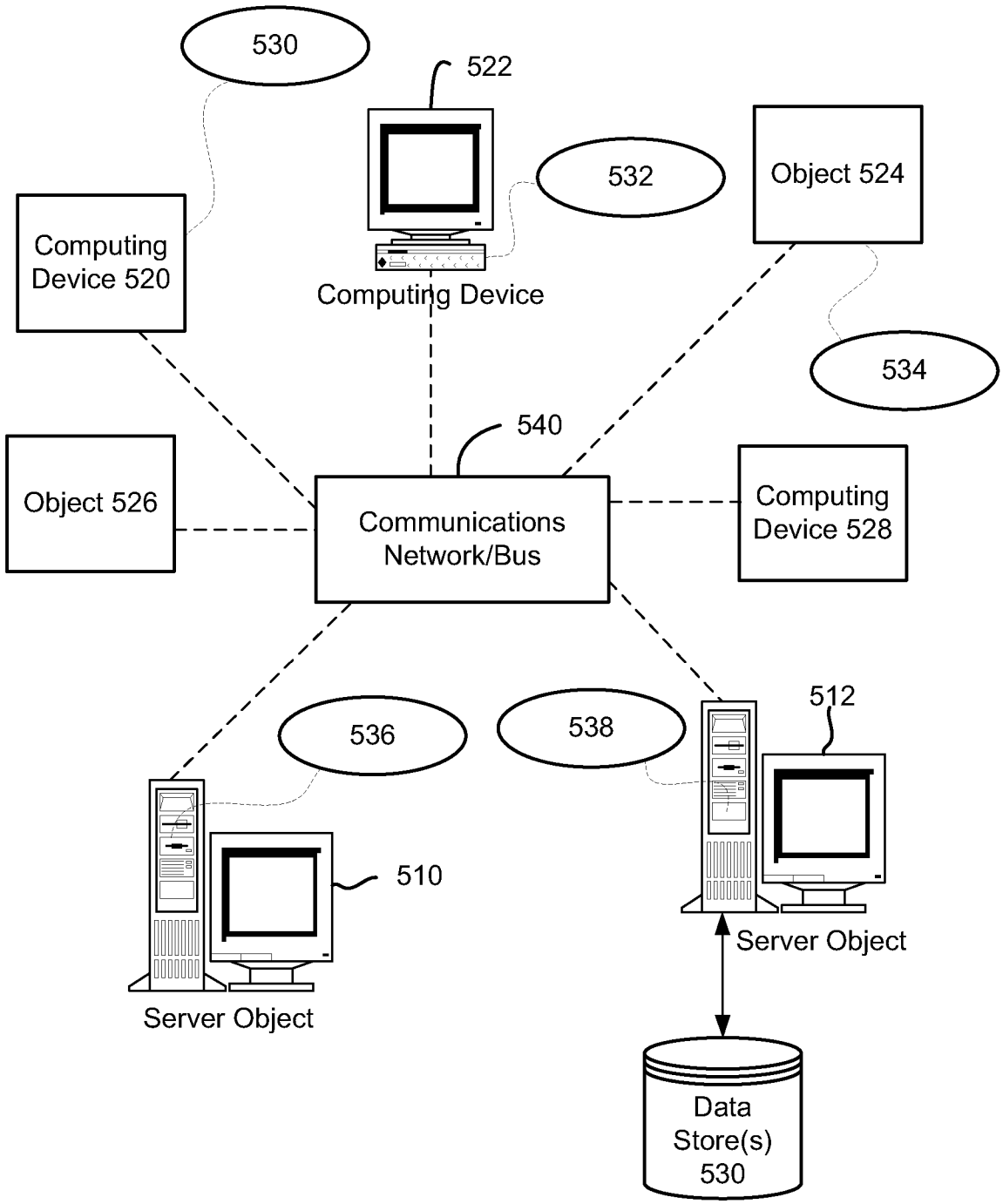

530
522
532
Object 524
Computing Device 520
Computing Device
534
540
Object 526
Communications Network/Bus
Computing Device 528
536
538
512
510
Server Object
Server Object
Data Store(s) 530

FIG. 5

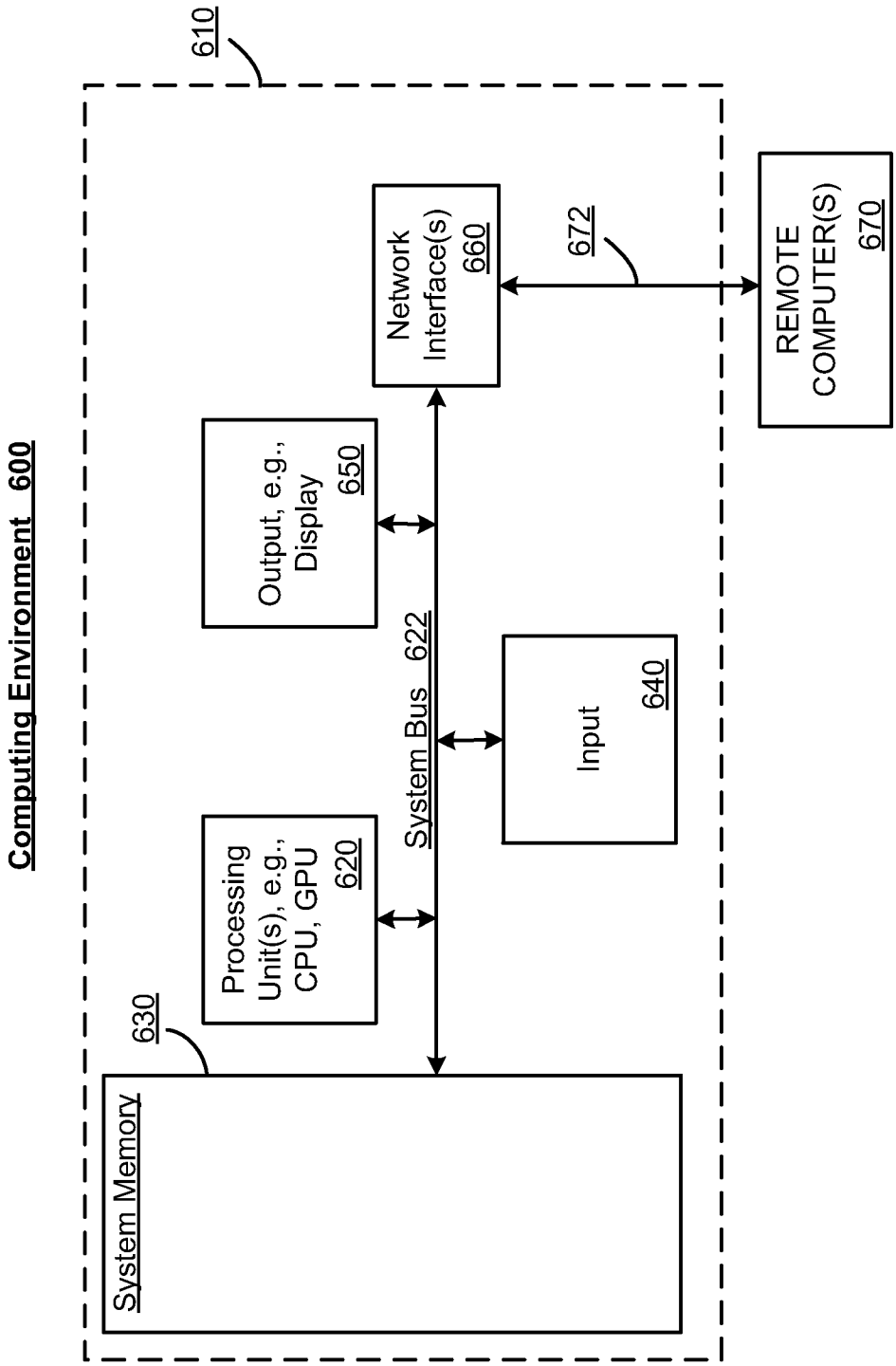

610
Computing Environment 600
Network Interface(s) 660
672
REMOTE COMPUTER(S) 670
Output, e.g., Display 650
System Bus 622
Input 640
Processing Unit(s), e.g., CPU, GPU 620
630
System Memory

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/063615

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/22(2006.01)i, G06F 9/44(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 21/22; G06F 21/20; H04L 9/32; G06F 12/00; H04L 9/00; G06F 21/00; G06F 11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: virtual, machine, protect, malware, virus, guest, partition, scan, agent, remediate, remedy, resume, snapshot, offline, signature, cloud

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2010-0058432 A1 (NEYSTADT JOHN et al.) 04 March 2010 See abstract; paragraphs [19] - [42]; figures 1-5.	1,5
A	See the whole document.	2-4,6-10
Y	US 2008-0320594 A1 (JIANG XUXIAN) 25 December 2008 See abstract; paragraphs [52] - [79]; figures 3-5.	1,5
A	See the whole document.	2-4,6-10
A	WO 2008-048665 A2 (UNIV. OF VIRGINIA PATENT FOUNDATION et al.) 24 April 2008 See abstract; page 8, line 10 - page 9, line 33; figures 1-4.	1-10
A	US 2008-0184373 A1 (TRAUT ERIC et al.) 31 July 2008 See abstract; paragraphs [25] - [31], [60] - [86]; figures 1, 4-11.	1-10
A	US 2008-0263658 A1 (MICHAEL MICHAEL L. et al.) 23 October 2008 See abstract; paragraphs [27] - [52]; figures 2-4.	1-10

☐ Further documents are listed in the continuation of Box C.

☒ 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 application or patent 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 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
05 APRIL 2012 (05.04.2012)	**27 APRIL 2012 (27.04.2012)**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office Government Complex-Daejeon, 189 Cheongsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea	Shin Sang Gil
Facsimile No. 82-42-472-7140	Telephone No. 82-42-481-8480

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/063615

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0058432 A1	04.03.2010	CN 102132287 A	20.07.2011
		EP 2318975 A2	11.05.2011
		WO 2010-025007 A2	04.03.2010
		WO 2010-025007 A3	22.04.2010
		WO 2010-025007 A3	04.03.2010
US 2008-0320594 A1	25.12.2008	None	
WO 2008-048665 A2	24.04.2008	US 2011-0047618 A1	24.02.2011
		WO 2008-048665 A3	03.07.2008
		WO 2008-048665 A3	24.04.2008
US 2008-0184373 A1	31.07.2008	CN 101589364 A	25.11.2009
		EP 2115570 A1	11.11.2009
		EP 2115570 A4	16.06.2010
		JP 2010-517162 A	20.05.2010
		JP 2010-517162 T	20.05.2010
		WO 2008-091452 A1	31.07.2008
US 2008-0263658 A1	23.10.2008	None	