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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 September 2010 (02.09.2010)

PCT

(10) International Publication Number
WO 2010/099174 A1

(51) International Patent Classification:
G06F 15/173 (2006.01)

(21) International Application Number:
PCT/US2010/025198

(22) International Filing Date:
24 February 2010 (24.02.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/154,995 24 February 2009 (24.02.2009) US

(71) Applicant *(for all designated States except US)*: **TELCORDIA TECHNOLOGIES, INC.** [US/US]; One Telcordia Drive 5g116, Piscataway, NJ 08854-4157 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ALEXANDER, D., Scott** [US/US]; 75 Hillcrest Road, Warren, NJ 07059 (US). **CHENG, Yuu-Heng** [—/US]; 300 N. Randolphville Road, Apt. 141, Piscataway, NJ 08854 (US). **POYLISHER, Alexander** [US/US]; 1717 Avenue N, Apt. 4M, Brooklyn, NY 11230 (US).

(74) Agents: **FEIG, Philip, J.** et al.; Telcordia Technologies, Inc., One Telcordia Drive 5G116, Piscataway, NJ 08854-4157 (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, RO, RS, RU, 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.

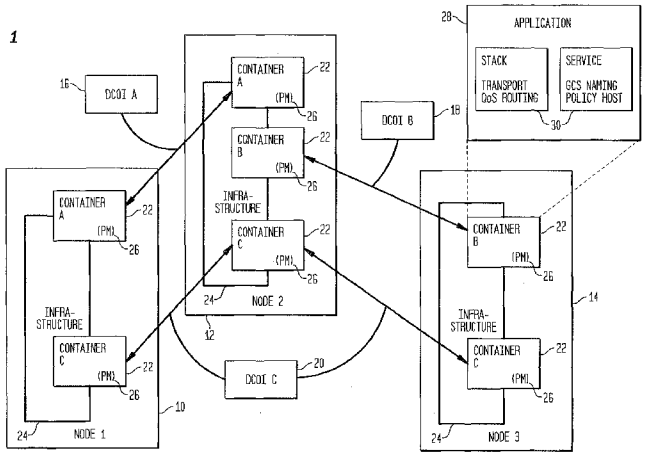
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS,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

Published:

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

(54) **Title**: SYSTEM AND METHOD FOR POLICY BASED MANAGEMENT FOR A HIGH SECURITY MANET

FIG. 1



(57) **Abstract**: A system and method for policy based management for a high security MANET comprises policy managers, each performing policy decision-making and policy enforcement using multiple policies, containers, each related to an application and each container having one policy manager, nodes, each having an infrastructure and at least one container, and dynamic community building blocks associating the containers having a same application, the containers being in different nodes, the associated containers maintained by the dynamic community building blocks on a secure network. Each container can define a security boundary around the node. Each container can be a lightweight virtual machine. The system can also have a special container having a policy manager only evaluating policies for conflicts, hi one embodiment, a node can consist of multiple network devices and each network device is a container of its own.

WO 2010/099174 A1

SYSTEM AND METHOD FOR POLICY BASED MANAGEMENT FOR A HIGH SECURITY MANET

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present invention claims the benefit of U.S. provisional patent application 61/154,995 filed February 24, 2009, the entire contents and disclosure of which are incorporated herein by reference as if fully set forth herein.

FIELD OF THE INVENTION

[0002] The present invention relates generally to network management, network security, and in particular to policy-based network management for mobile ad-hoc networks (MANETs).

BACKGROUND OF THE INVENTION

[0003] With the development of increasingly complex systems, policies are widely used to allow users a level of customization and automation of a system without the need for rebuilding or restarting. Existing policy-based management systems (PBNM or PBMS) define customization for access control, obligated behavior, and/or flow control, but do not address architecture security. Most existing policy systems have a centralized approach to security and are defined with centralized policy control, where security is provided by the network authentication process, though some may utilize authentication information as part of the policy condition. Typically, both connectivity and trust are assumed, allowing the deployment of a centralized or hierarchical system for both distribution and evaluation of policies. Generally, one trusted node is assumed to exist in the PBNM. However, any node in the network may be an undetected attacker.

[0004] In the established Policy Decision Point (PDP) to Policy Enforcement Point (PEP) model, e.g., IETF's COPS-PR and NetConf for obligation policies, such as Smart Firewalls for authorization policies, policy decisions are made at a logically central point and then each decision is dispatched to the endpoints. Existing systems attempt to secure the PDP-to-PEP communication with transport-level security and user-level authentication (e.g., ssh for NetConf). The hierarchical paradigm has been adapted for use in MANETs, by providing dynamic service discovery and redundancy.

[0005] In an alternative approach, policies are created and processed centrally on the node rather than being isolated by application.

[0006] Any dispatching of policy decisions over the network increases chances of decision hijacking or interference and allows hostile network traffic observers to perform traffic analysis and deduce system behavior. Since the policy decision is made by one node, an attacker can target one node to compromise the whole network. Additionally, making policy decisions for an entire node from a single PDP increases the ability of an attacker who compromised a single application to compromise an entire node. Both policy distribution and policy enforcement are protected, at best, only by transport-level security, and authenticated only at the user level. The protection is insufficient in a hostile environment and authentication is too coarse-grain for a secure network with application-specific policies.

[0007] Typically, most policy languages support any combination of conditions. A potential security concern is implicitly allowing unexpected conditions when using the negate operation in the condition.

[0008] To illustrate this concern, the following is a sample policy condition. In a system with only "blue" and "purple" tracks, this policy condition allows only "blue" tracks to join the group A, B, or C. However, when a new track, say "red", is introduced to the system, the policy implicitly permits both the "red" and "blue" tracks to join the group, which is potentially undesirable.

```
...
Conditions : (request == "join" ) &&
        ( group == "A" || group == "B" || group == "C" ) &&
        ( t r a c k != "p u r p l e " ) -> " t r u e " ;
...
```

[0009] Accordingly, to provide a high-security MANET, several new challenges are identified for the design of its PBMS.

[0010] First, strict enforcement of communication restrictions necessitates a highly decentralized network management, even within a node, with a PBMS instance for each

container, shared resources and secure infrastructure. This in turn creates more pressure on the resource consumption of a given instance, as node resources are typically scarce.

[0011] Second, high security requires explicit protection of policy integrity in transit, authentication of policy creators and authorization before a policy is enforced.

[0012] Third, policies must be made available to the PBMS instances in a timely manner to ensure enforcement. Given the dynamic nature of community building block creation and the intermittent connectivity in MANETs, this requires appropriate policy distribution mechanisms.

[0013] Fourth, the general policy deconfliction problem is complicated by the fully decentralized nature of policy creation and enforcement, and by the need to deconflict policies for a particular communication domain against shared resource and node policies within each node.

[0014] Thus, there exists a need for a PBNM system that is resilient to insider and external attacks. An inventive solution to this need includes identification of the security, performance and usability requirements of PBMS for high-security MANETs, and architectural and design solutions to meet these requirements.

SUMMARY OF THE INVENTION

[0015] An inventive system and method with mobile ad hoc networking with a particular emphasis on security is presented. The inventive system is one in which policy is not part of the trusted computing base. The approach directly supports information integrity, availability, reliability, confidentiality, and safety as a matter of design. The network enforces authentication and authorization of all actions, deny by default for all actions, resistance to Byzantine faults and insider threats, and support a selected set of functionality implemented in trusted hardware. The inventive system and method enables isolation of policy decision and enforcement to nodes and to security containers within nodes. Policies cannot affect actions in other containers. All policies received are fully authenticated. Policy defaults to deny, so all policy statements add privileges.

[0016] The inventive system, in an exemplary embodiment, comprises one or more policy managers, each policy manager performing policy decision-making and policy enforcement; one or more containers, each container related to an application and each container having one policy manager of the one or more policy managers; one or more nodes, each node having an infrastructure and at least one container of the one or more containers; and dynamic community building blocks associating the one or more containers having a same application, the containers being in different nodes, the associated containers maintained by the dynamic community building blocks on a secure network. In one embodiment, each container defines a security boundary around a communication domain on the node. In one embodiment, each container is a lightweight virtual machine. In one embodiment, the system also comprises a special container having a policy manager only evaluating policies for conflicts. In one embodiment, a node can consist of multiple network devices and each network device is a container of its own.

[0017] The inventive method, in an exemplary embodiment, comprises performing, on one or more containers, policy decision-making and policy enforcement using a policy manager in each of the one or more containers; associating the one or more containers having a same application; and maintaining the associated containers on a secure network.

[0018] A computer readable storage medium storing a program of instructions executable by a machine to perform one or more methods described herein also may be provided.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] The invention is further described in the detailed description that follows, by reference to the noted drawings by way of non-limiting illustrative embodiments of the invention, in which like reference numerals represent similar parts throughout the drawings. As should be understood, however, the invention is not limited to the precise arrangements and instrumentalities shown. In the drawings:

Figure 1 is a schematic diagram of the present invention;

Figure 2 shows components within a node in the present invention;

Figure 3 shows a general purpose policy-based management system with feedback; and

Figure 4 shows policy management components within a container.

DETAILED DESCRIPTION

[0020] The invention comprises a method and a system for a PBMS which addresses high security. Communications domains are assumed to be isolated within the node using security containers, called herein DCoI (Dynamic Community of Interest). A node can be a network device or a collection of network devices. Since one node may be running multiple applications and/or services, the node may have multiple containers. DCoI is illustrated in Figure 1, where nodes 1 **10** and 2 **12** belong to DCoI A **16**, nodes 2 **12** and 3 **14** to DCoI B **18**, and nodes 1 **10**, 2 **12** and 3 **14** belong to DCoI C **20**. A network device with multiple containers **22** consists of a base infrastructure **24** (or DOM0 for virtual machine) and several containers **22**. The communication among the containers **22** over the network is restricted to only the DCoI **16**, **18**, **20** that the container belongs to.

[0021] A container **22** defines a security boundary around the node **10**, **12**, **14** and includes resources for a single DCoI**16**, **18**, **20**. In one embodiment, it is implemented as a lightweight virtual machine. A container **22** is the security boundary of which application and its supporting service can manipulate resources restrained by the operating system or virtual machine. A policy manager (PM) **26** resides in each container **22** of the network device as shown in Figure 1.

[0022] Infrastructure **24** is the lower-level or device-level resource controller for the actual CPUs, memory and lower layers of the network stack (MAC and below) of the node **10**, **12**, **14** and containers **22**.

[0023] The behavior of the network nodes **30** must be automatically customizable at run-time as manual reconfiguration is impractical in the target operating conditions. While there are certain "rules" that must be statically built into the stack and services **30**, e.g., all traffic must be encrypted, other aspects, e.g., the list of potential DCoI members, are specific to the way that the system is deployed and the way a particular DCoI is being used. In one embodiment, the customization can be policy-managed and can allow the rules of behavior to be expressed and analyzed outside of the managed components. A unique set of requirements exist for the PM to satisfy supporting high security in a MANET environment. They include low bandwidth use (potentially low CPU and memory), support of the dynamic nature of DCoIs, minimal dependence of a node on another node for policy enforcement, operations are denied by default, policy integrity

must be ensured in storage and distribution, policy author and recipient must be authenticated, the operation of the policy-based management system must not breach exfiltration constraints, and access to resources used both by containers and within a container must be policy-managed.

[0024] Both computational (CPU, memory) and communication resources in a MANET environment can be very limited. Thus, a PM's processing footprint must be small, and policies themselves must be encodable in concise form. The PM user interface needs to allow administrators to create and update policies under the stress of tactical environments. This, apart from UI design, necessitates high usability of the policy language. In order to address Byzantine attacks, it is highly undesirable to allow a management system on a compromised node to execute operations on a remote node. Each node must have the authority and responsibility to enforce policy to protect itself and the network.

[0025] In a permit-all, explicit-deny authorization scheme, one can only deny the known operations, and the set of permitted operations is unknown, which opens attack possibilities. Conversely, in a deny-all, explicit-permit scheme, one can permit the exact minimal set of operations necessary. Additionally, unanticipated actions are denied, thus avoiding the issue that unanalyzed actions may result in security holes in the system. This is the basic rationale for a deny-all scheme in a secure environment. Individual policies must not be corrupted and must be up-to-date. The set of policies enforced in all containers of the same communication domain must be identical and self-consistent.

[0026] In policy distribution, it is important to authenticate the author because only an explicit set of users are trusted to create policies. In order to reduce the potential of an attacker to infer mission details from policies, policies should be distributed only to authorized nodes. The authentication of the policy distribution can be achieved by using existing secure transportation protocols such as secure FTP, ssh, etc.

[0027] Another requirement is to minimize the opportunity for exfiltration between communication domains, so policy distribution should only occur within the affected communication domain.

[0028] The inventive system is a fully decentralized policy system. Each PM instance evaluates its own set of policies and makes decisions to control the managed components within a container. The only communication over the network is the policy contents. An alternative solution is to dispatch policy decisions from one or several locations over the network. In general, a fully decentralized solution is preferred because of the following advantages. Policy contents do not change nearly as frequently as policy decision results, low network latency is far less critical for policy contents as it is for policy decisions, and it is easier to support need-to-know policy evaluation and Byzantine fault tolerance. Policy content is, of course, sensitive on its own. Knowing the access control policies, for example, may reveal the usage of a communication domain. Because of this sensitivity, not only are policies in transit protected, but also, to ensure that even if the protections fail, the set of policies to which an attacker has access is minimized.

[0029] As described above, each PM is instantiated inside of a container. PM **26** is duplicated in each container **22** as shown in Figure 2. This design reduces both the ability of an attacker to inspect policies from other containers and to affect the behavior of other containers if she manages to gain control of a container.

[0030] PM **26** supports both authorization and obligation policies. Because of the default deny-all authorization requirement, only positive authorization is supported. The behavior of the PM is identical within each container, however, the policies the PM evaluates belong to different scopes. The three different scopes are: container **22**, shared-resource **32**, and node **10**. Details of each scope are described below. These scopes are predetermined and orthogonal to the authorization/obligation categorization.

[0031] The inventive system defines a specific container, Shared Policy Container **32**, which runs a PM **26** that evaluates policies of the scope "shared resource." The PM in the shared policy container receives shared-resource policies from PM in other containers. This special PM only evaluates policies for conflict and does not enforce any policies. All of the enforcement is performed within each container. The PM receives its policies from the PMs in other containers. If there are conflicts, the corresponding PMs will be notified of a conflict. Note that the PM in the shared policy container can reside in the

infrastructure as an optimization. However, it may increase the risk of a compromised network device. The communication between PMs are constrained to occur along the paths created and enforced by Infrastructure **24** to ensure no other operations are allowed.

[0032] The PM can be based on the IETF policy-based management architecture [RFC 2753] with a feedback loop as shown in Figure 3. The PDP can be implemented by any policy engine such as: Keynote, XACML, etc.

[0033] Additional information regarding the scopes of the policies are discussed. One scope is Container. These container policies affect only a single container and the corresponding containers in other network devices that communicate with this container. For example, a container policy can address what type of network devices are allowed to communicate in this container, and/or the membership list for the container.

[0034] A second scope is Shared-resource. These shared-resource policies affect the usage of shared resources within a network device or node required for a container. These types of policies can be considered as a subset of policies with the scope container. For example, a shared-resource policy can address bandwidth allocation policies for each container.

[0035] A third scope is Node. These node policies affect the behavior of the network device. For example, a node policy can address the maximum number of containers that can be instantiated on a given node, such as: at most five containers can exist under specific conditions on this node.

[0036] Communications between PM instances, whether in communication domain containers **22**, the Policy Container **32**, or Infrastructure **24**, are constrained to occur along the paths created and enforced by Infrastructure. This helps reduce the opportunity for unauthorized access to policy information.

[0037] Figure 3 depicts the information flow for the basic functional components. In one embodiment, the basic functional components of the PM are similar to IETF policy-based management architecture, which includes the following. A Policy Decision Point (PDP)

34 evaluates the stored policies when triggered by a request (passive) or an event (proactive). The decision made is then passed to a Policy Enforcement Point (PEP) **36** which enforces policy decisions. Typically, the PEP **36** resides in the subsystem that the policy system manages. Policy Repository (PR) **38** stores the policies used in the policy system.

[0038] The feedback loop **40** allows the PDP **34** to retrieve (or be notified of) information from the managed system when evaluating policies and automating behaviors of the managed system. Although not part of the IETF-proposed architecture, the feedback loop is commonly used, particularly with obligation policies.

[0039] PM adopts the architecture described above with three additional components as follows. A Policy User Interface (PUI) allows policy authorities (administrators) to create and modify policies. In a military deployment, the PUI is enabled only on specific nodes determined by the mission. A Conflict Detection Point (CDP) detects conflicts between a newly created or modified policy against the existing set of stored policies. A Policy Distribution Interface (PDI) is used to securely distribute policies over the network.

[0040] Figure 4 depicts one embodiment of the inventive system in which the PM components and their relationships with another system within a container **22** are illustrated. These PM components contain the logic to control the policy-managed components. Each policy-managed component implements the PEP **36** that exports operations controllable by the PM. The PDP **34** manages the node by reacting to requests and events from other subsystems, evaluating matching policies and directly invoking PEP **36** operations.

[0041] Details of policy evaluation are described below. The requests include authorization and configuration requests. Services that started later than the PM can acquire its settings via a configuration requisition.

[0042] A limited set of events is defined to provide situation awareness for the node. The events serve as feedback information to the PDP. The usage of the events needs to be carefully designed. If one is not careful, an unstable system could result. For example, if

an event causes changes to the system that will publish the same event again, the system will be trapped in an endless loop. Currently PM only recognizes events that identify the current information operations conditions (INFOCON). Obligation policies can be applied to different communication mechanisms for the DCoI based on different INFOCON level.

[0043] Usually direct invocation of the PEP operation is triggered by the PDP receiving an event or request.

[0044] Deconfliction of the PDP can also be addressed. Additional meta-information can be brought into PM for conflict detection.

[0045] All policies are stored in the Policy Repository after the Policy Deconfliction Component ensures that the policies are conflict-free. The policies can be input into PM from either the PUI or PDI.

[0046] If PM were willing to accept any policy from any source, an attacker could subvert the system by providing her own (unauthorized) policy to facilitate the attack. In order to protect policy integrity and authenticate the policy author, all policies are signed by trusted entities. Over-the-air policy updates and even policy updates by commanders and their delegates in the field can be supported.

[0047] Only valid policies issued by a trusted entity are accepted by PM. Typically the trusted entity creates policies that follow the operational intent. Some operator errors are prevented by policy conflict detection. All policies are signed directly or indirectly by a trusted entity to ensure the policy integrity and allow traceability.

[0048] In the PDP engine, an implicit condition for matching policies is the validity of the policy signature. The validity of both the signature and the signer's privilege is verified since trusted entities may change over time, for example due to change of roles for adversary action, e.g., node capture, etc. The former check relies on the existing encoding algorithms, and the latter check is implemented using a certificate revocation list. In one embodiment, the cryptographic and keys can be pre-deployed. The signature verification

can be implemented by existing encoding/signing algorithms. The change of validity can be implemented, for example, using certificate revocation list [RFC 5280].

[0049] For a system with a significant number of policies, verifying the policy signature for each policy upon each request may not be efficient. The policy evaluation performance may be optimized by removing policies with invalid signatures before storing the policy or upon revoking a certificate.

[0050] After initial network deployment, administrators may need to create a new policy or modify an existing one. A modified policy set then needs to be distributed to all nodes belonging to the same communication domain. Since MANET connectivity can be unstable, the distribution mechanism should be able to handle intermittent connections. When a node is temporarily out of reach, that node should obtain the modified policy set when connectivity is resumed. Accordingly, each node must know that a new policy set is required, and that nodes that do not enforce the new policy set are prevented from participating in the communication domain. Since PM manages the node based on the content of the policies, the policy set for a communication domain needs to be consistently duplicated among all communication domain members.

[0051] During the lifetime of a communication domain, policy updates can be distributed via multicast to all members with simple synchronization mechanisms for tolerating unstable network connectivity. In one embodiment, the system can re-key when policy distribution occurs. In order to get the new key, a node must have a copy of the current policy set.

[0052] Keynote can be used in one embodiment as a base for the PDP implementation. It offers an evaluation environment where the default state is denial and policies specify the authorizations and actions allowed in the system. In the inventive system, Keynote can be extended to allow specification of obligation policies so that based on occurrence of particular events, policy evaluation returns corresponding multi-dimensional vectors containing specifications for configuration parameter changes or actions that need to be executed by PEPs.

[0053] A potential security concern is using the negate operation (for example, the 'not equal' operation '!=' in Keynote) in the policy condition statement. This is because PM policies are permissive policies, that is, policies specify only what the outcome should be, or what operations are permitted. Unexpected outcomes and declined operations are simply not mentioned in the policies. The condition in the policy of Listing (below) illustrates this concern. Specifically, the condition for track is changed to "not purple." In a system with only "blue" and "purple" tracks, the change makes no difference. However, when a new track, for example "red", is introduced to the system, the policy implicitly permits both the "red" and "blue" tracks to join the DCoI, which is potentially undesirable for a high security environment.

Listing: Example of negate condition

```
...                                                                    1
Conditions: (DCOI == "Chat") &&
            (group == "A" || group == "B" || group == "C") &&          3
            (track != "purple") &&
            (request == "join")  -> "true;                             5
...
```

[0054] Though the Keynote implementation of verify negative conditions is disabled, from a usability perspective, there can be some assistance in the user interface to expand the condition when editing a policy.

[0055] In the embodiment discussed above, the system by default denies all operations. But some operations, e.g. bootstrap, needs to be permit by default and denied afterwards. These specific operations are not part of the general policy system. During the bootstrap process for a container, PM starts with an initial set of policies if this node is the policy administrator of the communication domain.

[0056] Integrity includes the integrity of policies themselves and also the integrity of PDP decisions. All nodes are identified by their node creators or modifiers. During policy evaluation, a PDP only considers authenticated policies. PM utilizes a certificate revocation list to maintain the authorized set of policy signers. Policies that are deployed without a proper signature can be revoked from the repository.

[0057] Confidentiality indicates that specific information is known only by the information provider and its intended receiver. Besides network encryption for policy distribution, each DCoI container can have its own PBMS instance to process policy information.

[0058] In a MANET, network communication is not as reliable as in wired networks. Policies need to be made available to all nodes in a DCoI, so appropriate reliable protocols (unicast or multicast) are used. Though PM denies all operations by default, bootstrap processes are permitted to ensure the ability to setup initial communication between nodes.

[0059] It is likely that the solutions that fit the security-related requirements of PM can be applied to other network environments, e.g., wireline networks. Advantageously, this reduces the ability of attackers to attack the policy system. By attacking policy on one node, the attacker does not directly affect other nodes. By attacking policy for one application, the attacker does not affect policy for other applications. The attacker cannot look at policies for unrelated applications which prevents him from inferring information about users' future actions or missions. Authentication and validation of polices means that the system will not enforce corrupted policies.

[0060] As will be appreciated by one skilled in the art, the present invention may be embodied as a system, method or computer program product. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system."

[0061] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not

preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0062] The corresponding structures, materials, acts, and equivalents of all means or step plus function elements, if any, in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

[0063] Various aspects of the present disclosure may be embodied as a program, software, or computer instructions embodied in a computer or machine usable or readable medium, which causes the computer or machine to perform the steps of the method when executed on the computer, processor, and/or machine. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform various functionalities and methods described in the present disclosure is also provided.

[0064] The system and method of the present disclosure may be implemented and run on a general-purpose computer or special-purpose computer system. The computer system may be any type of known or will be known systems and may typically include a processor, memory device, a storage device, input/output devices, internal buses, and/or a communications interface for communicating with other computer systems in conjunction with communication hardware and software, etc.

[0065] The terms "computer system" and "computer network" as may be used in the present application may include a variety of combinations of fixed and/or portable computer hardware, software, peripherals, and storage devices. The computer system may

include a plurality of individual components that are networked or otherwise linked to perform collaboratively, or may include one or more stand-alone components. The hardware and software components of the computer system of the present application may include and may be included within fixed and portable devices such as desktop, laptop, server. A module may be a component of a device, software, program, or system that implements some "functionality", which can be embodied as software, hardware, firmware, electronic circuitry, or etc.

[0066] The embodiments described above are illustrative examples and it should not be construed that the present invention is limited to these particular embodiments. Thus, various changes and modifications may be effected by one skilled in the art without departing from the spirit or scope of the invention as defined in the appended claims.

What is claimed is:

1. A system for policy based management, comprising:

one or more policy managers, each performing policy decision-making and policy enforcement using one or more policies;

one or more containers, each container related to an application and each container having one policy manager of the one or more policy managers;

one or more nodes, each node having an infrastructure and at least one container of the one or more containers; and

communication domain building blocks associating the one or more containers having a same application, the containers being in different nodes, the associated containers maintained by the communication domain building blocks on a secure network.

2. The system of claim 1, wherein each container defines a security boundary within the node for the container.

3. The system of claim 1, wherein each container is a lightweight virtual machine.

4. The system of claim 1, further comprising a special container having a policy manager only evaluating policies for conflicts.

5. The system of claim 1, wherein each policy is signed by a trusted entity.

6. The system of claim 1, wherein a policy is updated over-the-air or in the field.

7. The system of claim 1, wherein the policies follow operational intent.

8. The system of claim 1, wherein some policies of the one or more policies are container policies, each container policy affecting only one container and the communication domain corresponding to the one container.

9. The system of claim 1, wherein some policies of the one or more policies are shared-resource policies, each shared-resource policy affecting usage of shared resources in one container and the node for the one container.

10. The system of claim 1, wherein some policies of the one or more policies are node policies, each node policy affecting behavior of a network device.

11. A method for policy based management, comprising:

performing, on one or more containers, policy decision-making and policy enforcement using one or more policies using a policy manager in each of the one or more containers;

associating the one or more containers having a same application; and

maintaining the associated containers on a secure network.

12. The method of claim 11, wherein each container defines a security boundary within a node.

13. The method of claim 11, wherein each container is a lightweight virtual machine.

14. The method of claim 11, further comprising a special container having a policy manager only evaluating policies for conflicts.

15. The method of claim 11, further comprising a step of signing each policy by a trusted entity.

16. The method of claim 11, wherein a policy is updated over-the-air or in the field.

17. The method of claim 11, wherein the policies follow operational intent.

18. The method of claim 11, wherein some policies of the one or more policies are container policies, each container policy affecting only one container and the communication domain that associated with the one container.

19. The method of claim 11, wherein some policies of the one or more policies are shared-resource policies, each shared-resource policy affecting usage of shared resources in one container and a node for the one container.

20. The method of claim 11, wherein some policies of the one or more policies are node policies, each node policy affecting behavior of a network device.

21. A computer readable medium storing a program of instructions executable by a machine to perform a method of policy based management, comprising:

performing, on one or more containers, policy decision-making and policy enforcement for one or more policies using a policy manager in each of the one or more containers;

associating the one or more containers having a same application; and

maintaining the associated containers on a secure network.

22. The program of claim 21, wherein each container defines a security boundary around the node.

23. The program of claim 21, wherein each container is a lightweight virtual machine.

24. The program of claim 21, further comprising a special container having a policy manager only evaluating policies for conflicts.

25. The program of claim 21, wherein each policy is signed by a trusted entity.

26. The program of claim 21, wherein a policy is updated over-the-air or in the field.

27. The program of claim 21, wherein the policies follow operational intent.

28. The program of claim 21, wherein some policies of the one or more policies are container policies, each container policy affecting only one container and communication domain that associated with the one container.

29. The program of claim 21, wherein some policies of the one or more policies are shared-resource policies, each shared-resource policy affecting usage of shared resources in one container and a node for the one container.

30. The program of claim 21, wherein some policies of the one or more policies are node policies, each node policy affecting behavior of a network device.

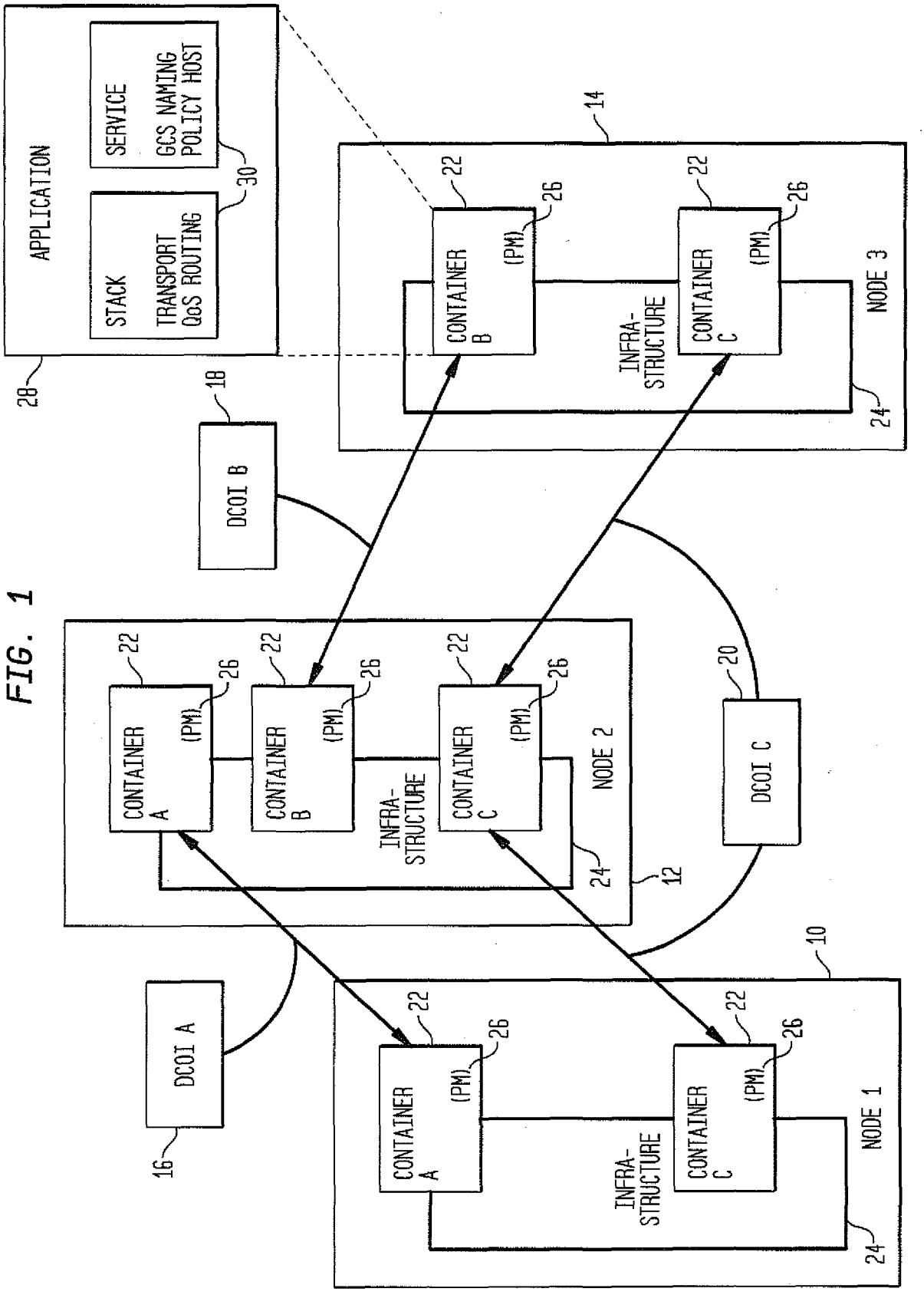

APPLICATION
SERVICE
GCS NAMING
POLICY HOST
STACK
TRANSPORT
QoS ROUTING
30
28
18
DCOI B
DCOI A
16
DCOI C
20
14
NODE 3
NODE 2
NODE 1
12
10
24
CONTAINER
A
B
C
(PM)
22
26
INFRA-
STRUCTURE

FIG. 1

FIG. 2

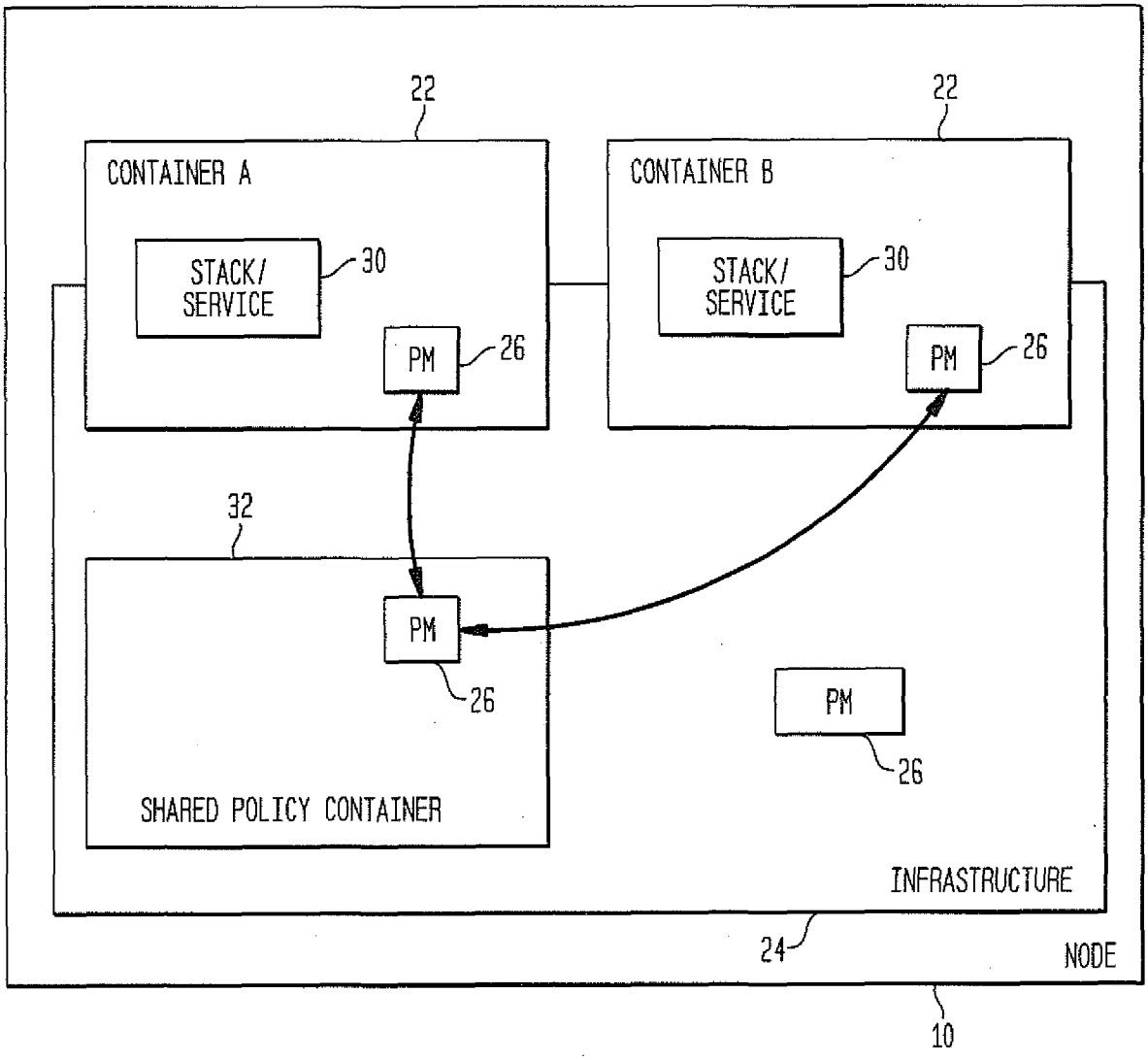

22
22
CONTAINER A
CONTAINER B
STACK/
SERVICE
30
STACK/
SERVICE
30
PM
26
PM
26
32
PM
26
PM
26
SHARED POLICY CONTAINER
INFRASTRUCTURE
24
NODE
10

FIG. 3

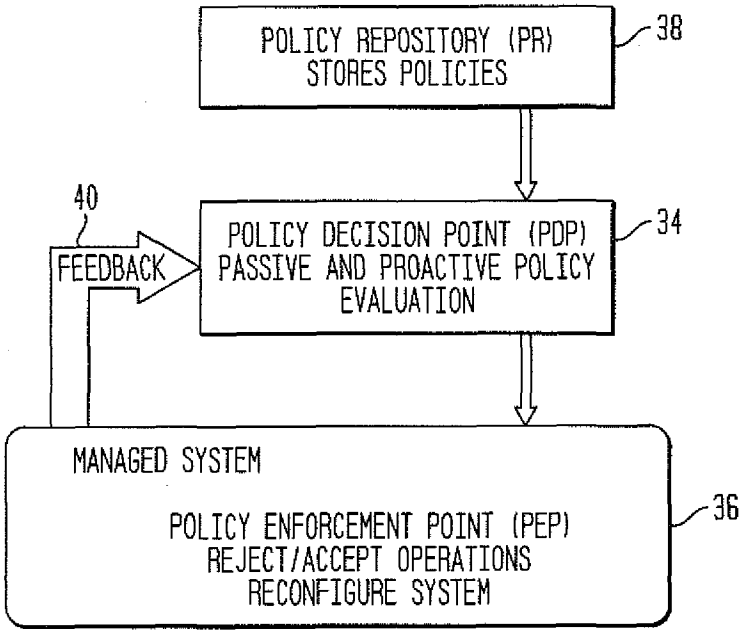
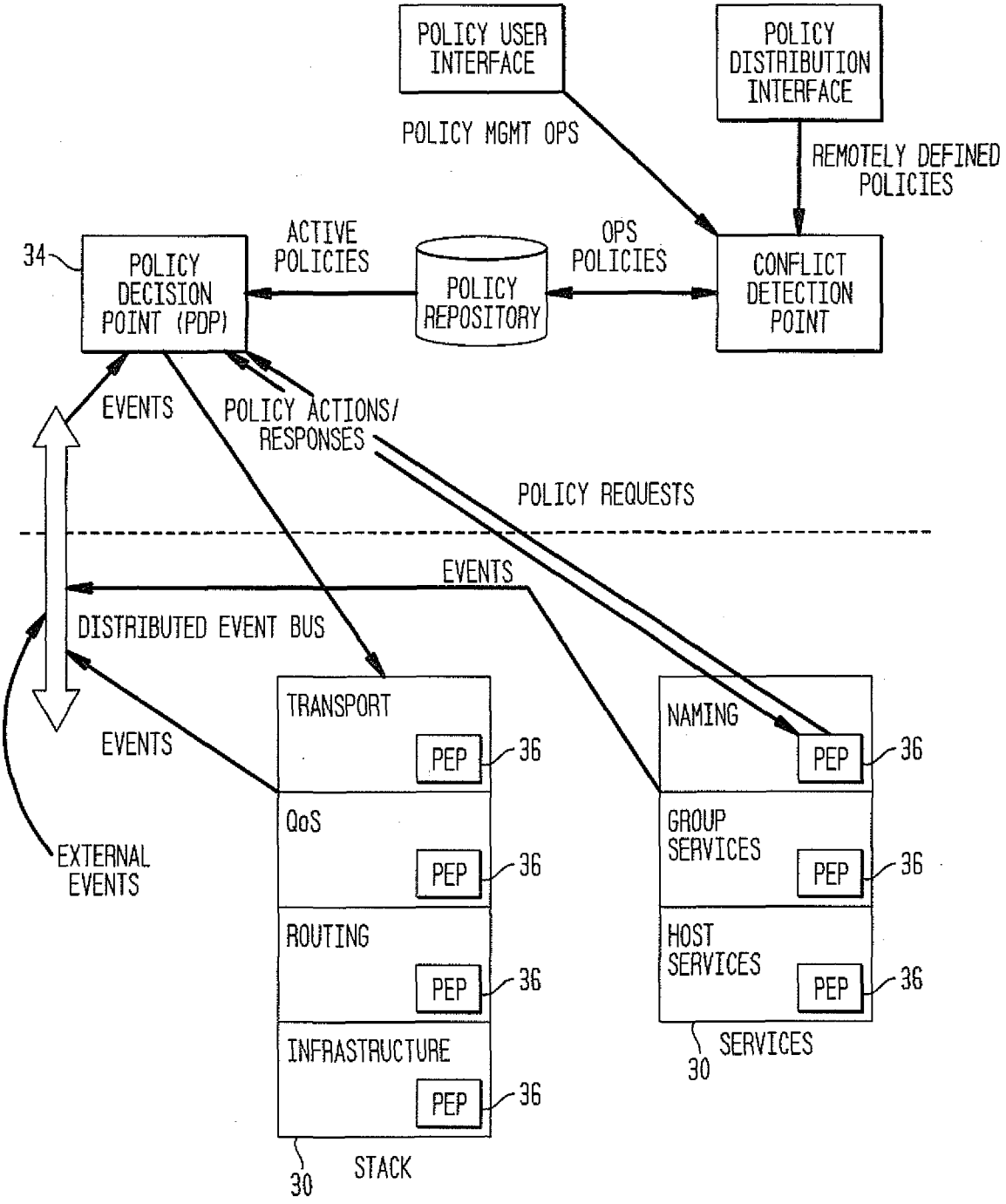

POLICY REPOSITORY (PR)
STORES POLICIES
38
40
FEEDBACK
POLICY DECISION POINT (PDP)
PASSIVE AND PROACTIVE POLICY
EVALUATION
34
MANAGED SYSTEM
POLICY ENFORCEMENT POINT (PEP)
REJECT/ACCEPT OPERATIONS
RECONFIGURE SYSTEM
36

FIG. 4


POLICY USER INTERFACE
POLICY DISTRIBUTION INTERFACE
POLICY MGMT OPS
REMOTELY DEFINED POLICIES
34
POLICY DECISION POINT (PDP)
ACTIVE POLICIES
POLICY REPOSITORY
OPS POLICIES
CONFLICT DETECTION POINT
EVENTS
POLICY ACTIONS/ RESPONSES
POLICY REQUESTS
EVENTS
DISTRIBUTED EVENT BUS
EVENTS
EXTERNAL EVENTS
TRANSPORT
PEP
36
QoS
PEP
36
ROUTING
PEP
36
INFRASTRUCTURE
PEP
36
30
STACK
NAMING
PEP
36
GROUP SERVICES
PEP
36
HOST SERVICES
PEP
36
30
SERVICES

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 10/25198

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 15/173 (2010.01)
USPC - 709/223
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)
USPC: 709/223

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 709/219,221,223,224,229

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Electronic Databases Searched: pubWEST(PGPB,USPT,USOC,EPAB,JPAB); GoogleScholar
Search Terms Used: container, policy, policySet, mobile ad-hoc networks, mobile device/node/entity/client, MANET, Chinese-wall, segregation, policy managers, application containers, communication domain, lightweight/java/VMworks virtual-machine, XACML

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008/0109679 A1 (WRIGHT et al.) 08 May 2008 (08.05.2008) Entire document, especially: para [0011], [0042], [0046]-[0048], [0064], [0161], [0167], [0171], [0177], [0179], [0271], [0341] and Fig. 1, 2A, 2B	1-30
Y	SCAGLIOSO, P.G., BASILE, C., LIOY, A., "Modern Standard-based Access Control in Network Services: XACML in action," IJCSNS International Journal of Computer Science and Network Security, Vol 8, No. 12, December 2008 (retrieved 23 March 2010 (23.03.2010)] Retrieved from the Internet <URL: http://paper.ijcsns.org/07_book/200812/20081242.pdf> Entire document, especially: pg 297, col 1, para 1 and col.2, para 2; pg 299, col 1, para 4 and col 2, para 1-5 and 12; pg 300, col 2, para 12; pg 301, col 1, para 6; pg 304, col 1, para 2; and Fig. 1	1-30
A	US 2008/0077971 A1 (WRIGHT et al.) 27 March 2008 (27.03.2008)	1-30
A	US 2008/0052395 A1 (WRIGHT et al.) 28 February 2008 (28.02.2008)	1-30
A	US 2008/0046965 A1 (WRIGHT et al.) 21 February 2008 (21.02.2008)	1-30
A	US 2006/0224742 A1 (SHAHBAZI) 05 October 2006 (05.10.2006)	1-30

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

* 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 another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 March 2010 (23.03.2010)	08 APR 2010
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774