Patentability Search Report
Endpoint Management with a Peer-To-Peer Mesh Architecture
(Docket No: LEX099P001)

Prepared By:

Date: September 19, 2025

A10, 8th Floor, iSprout- Shilpitha Tech Park, 55/3 55/4,
osite Sakra World Hospital, Devarabisanahalli, Bellandur,
Bengaluru, Karnataka-560103
Phone: +91-9916193248/+91-9540474969; Email: info@lexanalytico.com

Table of Contents

Table of Contents 2
1. Key Inventive Concept 3
2. Search Focus 4
3. Search Methodology 5
4. Database Used 6
5. Relevant Keywords 6
6. Classification Codes 7
7. Relevant References: Patent Literature 8
US20250004784A1 8
US20250047588A1 12
US20250047690A1 16
US12081558B2 20
US20230030990A1 25
8. Non-Patent Literature 30
NPL1 30
9. List of Additional Patents and Non-Patent Literature 32
10. Exemplary Search Strings 33
11. Disclaimer 35

1. Key Inventive Concept

The proposed invention relates to…

2. Search Focus

Both patents & non-patent literature disclosing the below concepts were considered. The shortlisted documents are provided with respective relevant excerpts:

1. An…

    1.1.

3. Search Methodology

Step 1: Understanding and Making Search Strategy

- An in-depth understanding of the **"Endpoint Management with a Peer-To-Peer Mesh Architecture"** was analyzed in terms of project requirements.
- A thorough study of the technology domain was performed by web research to gather relevant information.
- Key concepts are identified and defined using keywords and their synonyms.
- Key strings are prepared based on identified search terms, and relevant patent classifications.

Step 2: Searching and Analysis

- A broad-to-narrow search strategy (or narrow-to-broad) was employed using various search strings on a few commercial/free databases to identify patents/applications.
- The extracted documents are analyzed in detail to identify potentially relevant documents which were further segregated as relevant and related depending on number of features matching with the technical features of the study.
- For Patent literature only one member per family is considered for analysis.
- For non-English documents, the analysis is carried out based on machine-translated text available from free/commercial sources.

Step 3: Additional Searches

To ensure search comprehensiveness, the following searches were performed**:**

- Inventor/Assignee based search - The assignee/inventor of client's interest or the assignee/inventor from the identified shortlisted documents.
- IPC/CPC/ECLA/US search - Various classes are used with/without the combination of keywords.
- Semantic - Commercial databases are used to search on contextual meaning of terms.
- Similarity search - A similarity search of the target patent and identified relevant prior art is conducted in the commercial databases
- Citation Search - Two level citation searches of closely identified prior arts are executed.

Step 4: Report

- The shortlisted relevant documents along with the bibliographic details and text mapping are provided in a user-friendly, MS Word/PDF report.
- Related documents are provided in the form of list in the report.
- Bibliographic details of both relevant and related documents are provided in the report.
- All the documents are provided with hyperlinks to respective patent office sites or Espacenet.
- A tabulated summary of the relevant references is provided with executive summary.

4. Database Used

Patent Databases	Non-Patent Databases
✓ Questel Orbit ✓ Dolcera Patent Categorization System ✓ Google Patents ✓ Espacenet ✓ USPTO ✓ Free Patents Online	✓ General Google search ✓ Google Scholar ✓ Science Direct ✓ IEEE ✓ Springer, etc.

5. Relevant Keywords

Key words	Synonyms/Alternative terms
Endpoint management	unified endpoint management, UEM, patch management, vulnerability management, endpoint monitoring
Peer-to-peer	distributed, decentralised, decentralized, peer to peer
Mesh	grid, network, distributed system, peer-to-peer network
Agents	virtual captain, cognitive agents
Artificial intelligence	LLM, large learning models, AI, machine learning
Sensor	detector, transducer, indicator
Orchestrator	coordinator, organiser, administrator, leader
Fault tolerant	fail safe, redundant, error tolerant, reliable, impervious
Device discovery	device detection, device identification, device recognition, device finding
Availability	Accessibility, readiness, convenience, load, predictability
Tasks	threat detection, vulnerability, scan, patching, malware detection

6. Classification Codes

IPC/CPC Codes	Definition
G06F15/00	Electric digital data processing - Digital computers in general; data processing equipment in general
G06F15/16	Combinations of two or more digital computers each having at least an arithmetic unit, a program unit and a register, e.g. for a simultaneous processing of several programs
G06F8/65	Arrangements for software engineering - Updates
G06F9/4401	Arrangements for program control, e.g. control units- Bootstrapping
H04J3/00	Multiplex communication - Time-division multiplex systems
H04J3/06	Synchronising arrangements
H04L12/16	Arrangements for providing special services to substations contains provisionally no documents
H04L12/18	For broadcast or conference, e.g. multicast
H04L12/1836	With heterogeneous network architecture
H04L12/184	With heterogeneous receivers, e.g. layered multicast
H04L45/02	Routing or path finding of packets in data switching networks - Topology update or discovery
H04L45/22	Alternate routing
H04L45/74	Address processing for routing
H04L47/00	Traffic regulation in packet switching networks
H04L47/70	Admission control or resource allocation
H04L63/00	Network architectures or network communication protocols for network security
H04L63/1433	Vulnerability analysis
H04L63/10	For controlling access to network resources
H04L63/101	Access control lists [acl]
H04L67/141	Provided for setup of an application session

7. Relevant References: Patent Literature

Publication No.	US20250004784A1		
Title	State management with distributed control plane		
Assignee/Applicant	Dell Products LLP		
Earliest Priority Date	2023-06-27	Publication Date	2025-01-02

Abstract

Methods and systems for manage operation of endpoint devices are disclosed. To manage the operation of endpoint devices, management systems may monitor and update the state of the endpoint devices. To manage incongruencies among state updates and the states of the endpoint devices, the endpoint devices may implement a state management model that vests authority over the true states of the endpoint devices in the endpoint devices. Consequently, the endpoint devices may resolve incongruencies by rejecting some state updates that do not reflect the true states of the endpoint devices.

Description

[0011] In general, embodiments disclosed herein relate to **methods and systems for distributed management of the operation of endpoint devices. To manage the operation of endpoint devices, a system may include any number of management systems. The management systems may be tasked with managing the operation of the endpoint devices.**

[0033] Additionally, in distributed scenarios, different requesting entities may have different understandings regarding the services provided by endpoint devices 100. Consequently, requests from different requesting entities may be incongruent. Endpoint devices 100 may be unable to resolve how to implement multiple incongruent requests for computer implemented services.

[0043] Endpoint devices 100 may provide computer implemented services. To provide the computer implemented services, endpoint devices 100 may manage its states using a state management model. **When managing its state, an endpoint device may (i) obtain desired state updates from management systems 110 and/or other components, (ii) evaluate the desired state updates based on whether the desired state updates are based on an accurate understanding of the states of the endpoint device, (iii) for positively evaluated desired state updates, update its states based on the desired state updates, (iv) attempt to update its operation based on its states, and/or (iv) distribute information regarding its states and attempts to align its states to other entities such as management systems 110.** By updating its state based on positively evaluated desired state

updates, endpoint devices 100 may update their operation over time to provide desired computer implemented services while gracefully resolving conflicts between different requests to modify their states. Refer to FIGS. 2A-2C for additional details regarding updating the operation of endpoint devices 100.

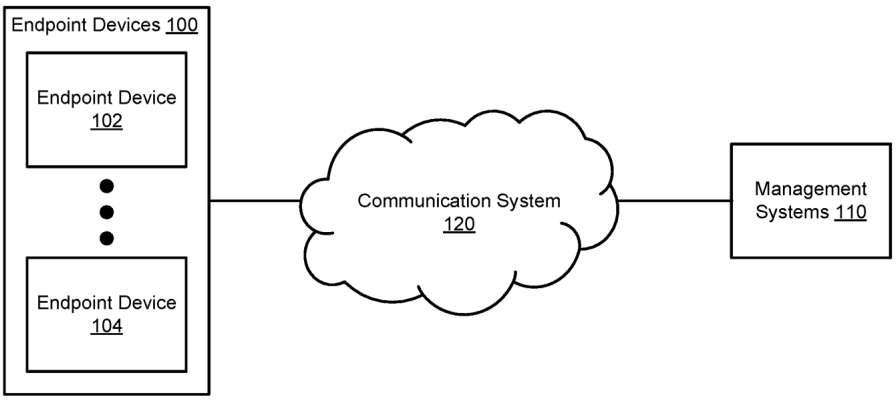


FIG. 1

[0044] Management systems 110 may manage the computer implemented services provided by endpoint devices 100. To do so, management systems 110 may **(i) attempt to track the state of endpoint devices 100 based on a state model (e.g., that tracks the desired, configured, and actual states of endpoint devices 100, as well as state transition progress), (ii) generate and provide desired state updates to endpoint devices 100 (e.g., depending on the computer implemented services desired by management systems 110 and/or other entities), and (ii) attempt to resolve state transitions delays impacting endpoint devices 100. Refer to FIGS. 2A-2C for additional details regarding management of endpoint devices 100 by management systems 110.**

[0059] Once provided to endpoint device 102, state management process 250 may ingest the desired state updates. If the desired state updates express a configured state for endpoint device 102 that is consistent with the configured state of endpoint device 102 as understood by endpoint device 102, then endpoint device may deem the desired state updates as being valid and implement them. The desired state updates not deemed as valid may be rejected. In this manner, only management systems that have a consistent understanding of the configured state of endpoint device 102 may be able to modify the states of endpoint device 102. Consequently, incongruencies between state updates may be resolved through rejection of some of the state updates.

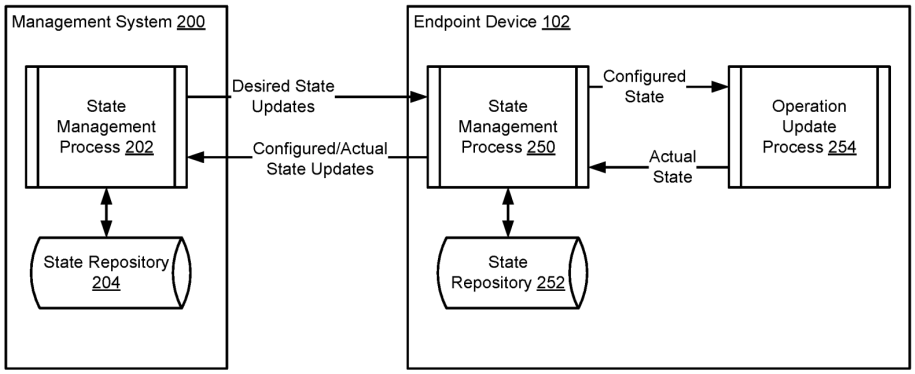


FIG. 2A

[0103] Turning to FIG. 3B, a flow diagram illustrating a method for distributing state updates for endpoint devices in accordance with an embodiment is shown. **The method may be performed by any of endpoint devices 100, management systems 110, and/or other components of the system shown in FIG. 1** .

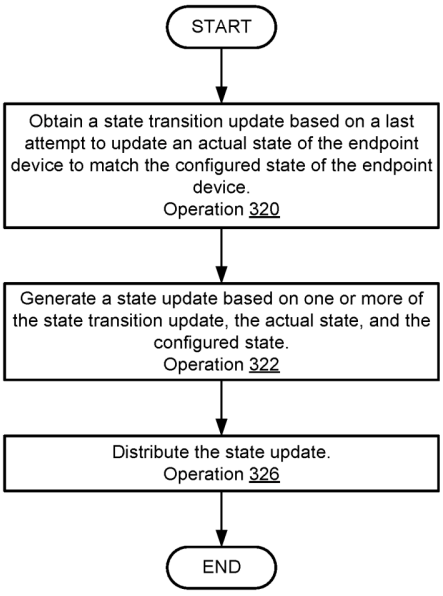


FIG. 3B

[0104] At operation 320, a state transition update based on a last attempt to update an actual state of the endpoint device to match the configured state of the endpoint devices is obtained. The state transition update may be obtained similarly to as described with respect to operation 306.

[0105] At operation 322, **a state update is generated based on one or more of the state transition update, the actual state, and the configured state. The state update may be an endpoint side state update. The endpoint side state update may be generated by adding the aforementioned information to a data structure.**

Publication No.	US20250047588A1		
Title	System and method for dynamic routing and scalable management of endpoint device communications		
Assignee/Applicant	Sensus Spectrum LLC		
Earliest Priority Date	2023-08-01	Publication Date	2025-02-06

Abstract

Described herein is a distributed endpoint processing system and method for managing and routing communications within the distributed endpoint processing system. The distributed endpoint processing system can dynamically accommodate additional endpoint devices or new endpoint types, enhancing flexibility and adaptability. By applying a customizable routing map generation policy, the distributed endpoint processing system assigns processing nodes for handling endpoint device communications, facilitating efficient and effective operation. Incoming messages from endpoint devices are processed and routed to the appropriate processing nodes based on the generated routing maps. Additionally, the distributed endpoint processing system can support API path configurations, which can be utilized to access and manage various system components.

Description

[0025] FIG. 1 illustrates an embodiment of a distributed endpoint processing system 100 that includes **an endpoint device 102, a dynamic data dispatcher 110, a route management system 120, an endpoint processing system 130, and an endpoint proxy manager 140. To simplify discussion and not to limit the present disclosure, FIG. 1 illustrates only endpoint device 102, though multiple endpoint devices may be part of the distributed endpoint processing system 100. In some cases, one or more of the endpoint device 102, the dynamic data dispatcher 110, the route management system 120, the endpoint processing system 130, or the endpoint proxy manager 140 may be excluded or separate from the distributed endpoint processing system 100.**

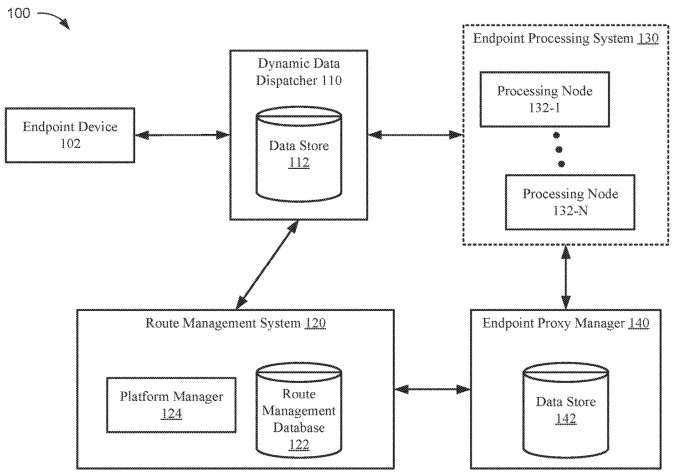


FIG. 1

[0065] FIG. 2 is a data flow diagram illustrating an embodiment of **data flow and communications between various components in the distributed endpoint processing system 100 for managing and processing messages from endpoint devices 102. The data flow diagram of FIG. 2 demonstrates an example of data flow and communications between the endpoint device 102, the dynamic data dispatcher 110, the route management system 120, and a first processing node 232. It will be understood that, in some of embodiments, one or more of the functions described herein with respect to FIG. 2 can be omitted, performed concurrently or in a different sequence, and/or carried out by another component of the distributed endpoint processing system 100. Accordingly, the illustrated embodiment and description should not be construed as limiting.**

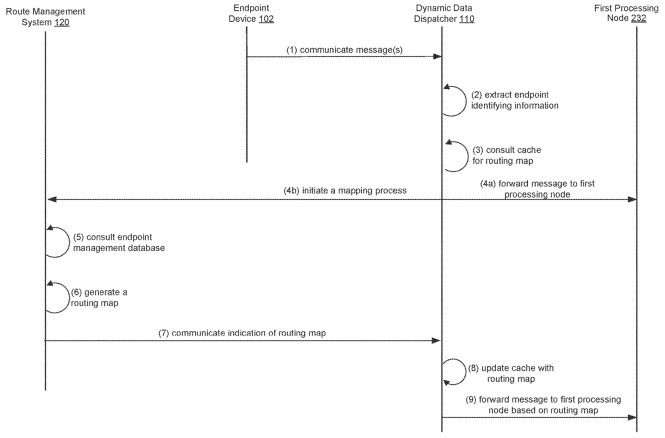


FIG. 2

[0066] At (1), the dynamic data dispatcher 110 obtains communications from the endpoint device 102. These communications may include data, requests, or other messages sent by the endpoint device 102. In some

cases, the dynamic data dispatcher 110 may collect this information in response to a new endpoint device being introduced or initiated within the distributed endpoint processing system 100. For example, when introduced within the distributed endpoint processing system 100, the endpoint device 102 can broadcast messages. The communications obtained by the dynamic data dispatcher 110 can include endpoint identifying information, as described herein.

[0072] In some cases, at interaction (4b), **the dynamic data dispatcher 110 may be able to identify a routing map in the route management database 122 itself, without requiring communication with the route management system 120. For example, if the endpoint identifying information is sufficient to identify an existing mapping in the route management database 122, the dynamic data dispatcher 110 may retrieve the routing information from the route management database 122 directly. This can allow the dynamic data dispatcher 110 to quickly obtain the routing information and forward the communications to the appropriate processing node 132 for processing, without having to initiate a routing map process with the route management system 120.** When the dynamic data dispatcher 110 finds a routing map for the endpoint device in the route management database 122, the dynamic data dispatcher 110 can forward the communications to the appropriate processing node. In cases where the routing information is not available in the data store 112 or the route management database 122, the dynamic data dispatcher 110 may communicate with the route management system 120 to obtain the mapping information.

[0085] FIG. 3 is a flow diagram illustrating an embodiment of a routine 300 implemented by a computing device within the distributed endpoint processing system 100. **While described as being executed by the route management system 120, it should be understood that the elements outlined for routine 300 can be implemented by one or more computing devices or components associated with the distributed endpoint processing system 100, such as, but not limited to, one or more endpoint devices 102, the dynamic data dispatcher 110, the route management database 122, or the endpoint processing system 130.** Consequently, the following illustrative embodiment should be regarded as non-limiting. Additionally, the routine 300 can include fewer, more, or different blocks, depending on the specific implementation.

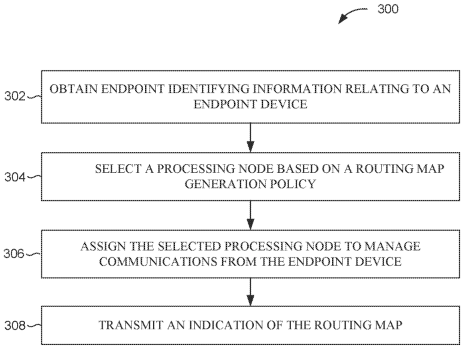


FIG. 3

[0086] At block 302, the route management system 120 obtains endpoint identifying information that is indicative of at least one of an identity or a characteristic of an endpoint device within a distributed endpoint processing system. As described herein, the endpoint identifying information can be broadcasted by the endpoint device as a part of one or more messages in response to introduction of the endpoint device into the distributed endpoint processing system.

Publication No.	US20250047690A1		
Title	Security management for endpoint nodes of distributed processing systems		
Assignee/Applicant	Dell Products LLP		
Earliest Priority Date	2023-08-02	Publication Date	2025-02-06

Abstract

An apparatus includes at least one processing device configured to determine, for endpoint nodes of a distributed processing system, node security information characterizing security issues encountered on one or more of the endpoint nodes. The processing device is also configured to identify, based on the node security information, a first type of security issues encountered on a first endpoint node and a second type of security issues encountered on a second endpoint node. The processing device is further configured to select first and second sets of corrective actions for the first and second types of security issues. The processing device is further configured to apply, to the first endpoint node, the first set of corrective actions, and to apply the second set of corrective actions by deploying an additional endpoint node in the distributed processing system and migrating workloads running on the second endpoint node to the additional endpoint node.

Description

[0011] FIG. 1 shows an information processing system 100 configured in accordance with an illustrative embodiment. **The information processing system 100 comprises a plurality of host devices 101-1, 101-2 . . . 101-N, collectively referred to herein as host devices 101, and a distributed storage system 102 shared by the host devices 101. The distributed storage system 102 is an example of what is more generally referred to herein as a distributed processing system, which may include a combination of one or more compute and storage nodes.** The host devices 101 and distributed storage system 102 in this embodiment are configured to communicate with one another via a network 104 that illustratively utilizes protocols such as Transmission Control Protocol (TCP) and/or Internet Protocol (IP), and may therefore be referred to herein as a TCP/IP network, although it is to be appreciated that the network 104 can operate using additional or alternative protocols. In some embodiments, the network 104 comprises a storage area network (SAN) that includes one or more Fibre Channel (FC) switches, Ethernet switches or other types of switch fabrics.

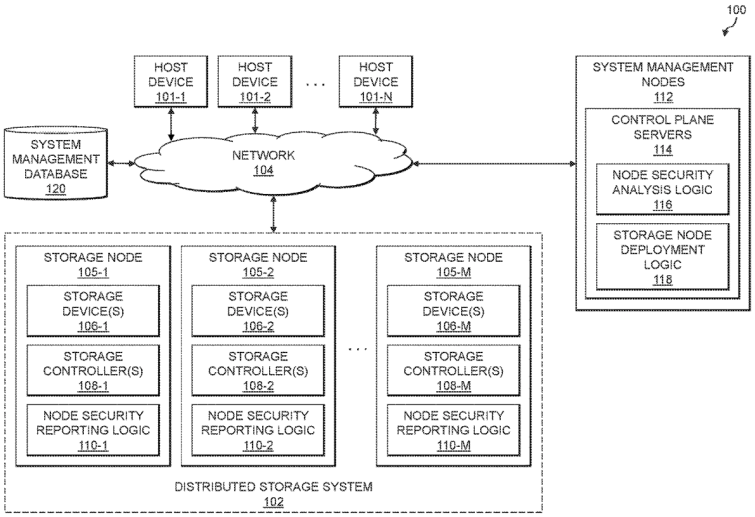


FIG. 1

[0012] The distributed storage system 102 more particularly comprises a plurality of storage nodes 105-1, 105-2, . . . 105-M, collectively referred to herein as storage nodes 105. The values N and M in this embodiment denote arbitrary integer values that in the figure are illustrated as being greater than or equal to three, although other values such as N=1, N=2, M=1 or M=2 can be used in other embodiments.

[0013] The storage nodes 105 collectively form the distributed storage system 102, which is just one possible example of what is generally referred to herein as a "distributed storage system." Other distributed storage systems can include different numbers and arrangements of storage nodes, and possibly one or more additional components. For example, as indicated above, a distributed storage system in some embodiments may include only first and second storage nodes, corresponding to an M=2 embodiment. Some embodiments can configure a distributed storage system to include additional components in the form of a system manager implemented using one or more additional nodes.

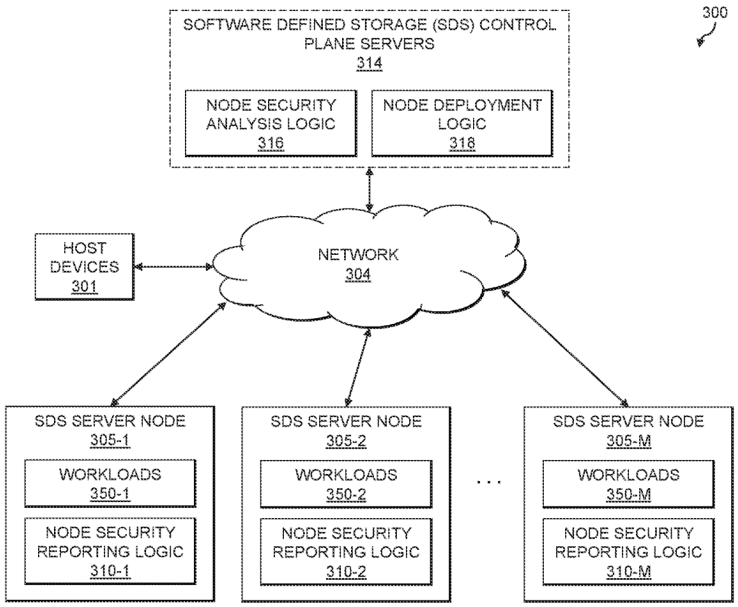


FIG. 3

[0017] Each of the storage nodes 105 is illustratively configured to interact with one or more of the host devices 101. The host devices 101 illustratively comprise servers or other types of computers of an enterprise computer system, cloud-based computer system or other arrangement of multiple compute nodes associated with respective users.

[0018] The host devices 101 in some embodiments illustratively provide compute services such as execution of one or more applications on behalf of each of one or more users associated with respective ones of the host devices 101. Such applications illustratively generate input-output (IO) operations that are processed by a corresponding one of the storage nodes 105. The term "input-output" as used herein refers to at least one of input and output. For example, IO operations may comprise write requests and/or read requests directed to logical addresses of a particular logical storage volume of one or more of the storage nodes 105. These and other types of IO operations are also generally referred to herein as IO requests.

[0127] FIG. 4 shows a system 400 including a multi-cloud orchestration layer 414 which manages a set of cloud endpoint nodes 405-1, 405-2 . . . 405-M (collectively, cloud endpoint nodes 405). **The multi-cloud orchestration layer 414 and the cloud endpoint nodes 405 communicate over network 404. The multi-cloud orchestration layer 414 is shown in dashed outline as the functionality of the multi-cloud orchestration layer 414 may be distributed over at least a subset of the cloud endpoint nodes 405 rather than being implemented on separate**

servers or other nodes. In some embodiments, different ones of the cloud endpoint nodes 405 run on different clouds of one or more different cloud service providers. The cloud endpoints nodes 405-1, 405-2 . . . 405-M run workloads 450-1, 450-2 . . . 450-M (collectively, workloads 450) on behalf of one or more requesting host devices 401. Such workloads 450 may introduce security vulnerabilities or other security issues on the cloud endpoint nodes 405. The cloud endpoint nodes 405-1, 405-2 . . . 405-M implement respective instances of node security reporting logic 410-1, 410-2 . . . 410-M (collectively, node security reporting logic 410) which reports node health information (e.g., vulnerabilities or other security issues encountered on the cloud endpoint nodes 405, which may in some cases be a result of running the workloads 450) to the multi-cloud orchestration layer 414.

[0132] FIG. 5 shows an example processing platform comprising cloud infrastructure 500. **The cloud infrastructure 500 comprises a combination of physical and virtual processing resources that may be utilized to implement at least a portion of the information processing system 100. The cloud infrastructure 500 comprises multiple virtual machines (VMs) and/or container sets 502-1, 502-2 . . . 502-L implemented using virtualization infrastructure 504. The virtualization infrastructure 504 runs on physical infrastructure 505, and illustratively comprises one or more hypervisors and/or operating system level virtualization infrastructure. The operating system level virtualization infrastructure illustratively comprises kernel control groups of a Linux operating system or other type of operating system.**

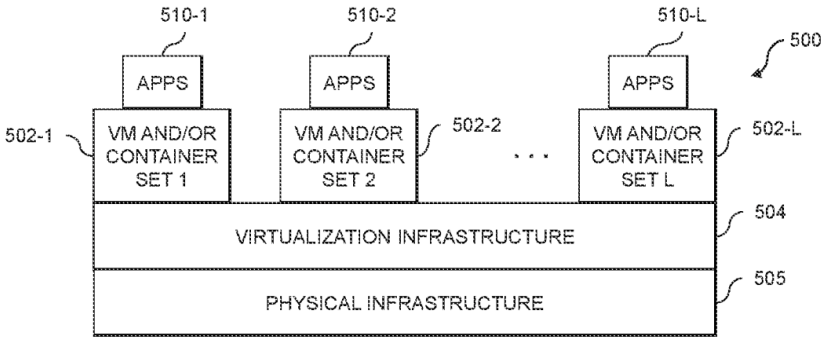


FIG. 5

Patent No.	US12081558B2		
Title	Distributed security in a secure peer-to-peer data network based on real-time guardian protection of network devices		
Assignee/Applicant	Whitestar Communications Inc.		
Earliest Priority Date	2021-06-29	Grant Date	2024-09-03

Abstract

In one embodiment, a method comprises: securing, by a security agent executed within a network device, first secure data structures for secure storage in the network device and second secure data structures for secure communications in a secure peer-to-peer data network; monitoring, by the security agent, a corresponding mandatory lifecycle policy for each of the first secure data structures; and cryptographically erasing one of the first secure data structures in response to expiration of the corresponding mandatory lifecycle policy.

Description

[Col. 5-7, Lines 49-2]

FIG. 1 illustrates a secure data network 5 comprising an example secure private core network 10, according to an example embodiment. **The secure private core network 10 is: a (1) cloudless (2) hybrid peer-to-peer overlay network that (3) can utilize artificial intelligence (AI) to extend security features and operations beyond end-to-end encryption between two endpoint devices 12, for example wireless smartphone devices, wireless smart tablet devices, wireless Internet of Things (IoT) devices, etc. The secure private core network 10 comprises a master control program (MCP) device 14, and one or more replicator devices (e.g., "R1") 16. Each replicator device 16 can be connected to every other replicator device 16, forming a pairwise topology (e.g., a "mesh") 98 of interconnected replicator devices 16; each replicator device 16 also is connected to the MCP device 14; hence, each replicator device 16 provides a connection to zero or more endpoint devices 12 for reaching the MCP device 14 and/or another endpoint device 12, described in further detail below.** The devices 12 also can have peer to peer connections to one another allowing direct communications without the aid of the core network 10 (hence the name hybrid peer to peer network). Devices 12 can simultaneously communicate either exclusively with each other, peer to peer, with some devices peer to peer and other devices via the core network 10 or with all other devices 12 via the core network 10.

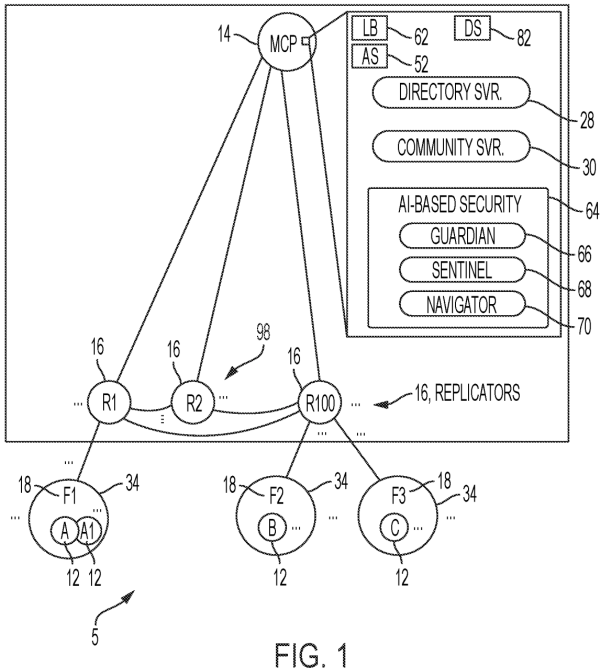


FIG. 1

The peer-to-peer network in the secure private core network 10 is based on a trusted aggregation of strict two-way trusted relationships ("cohorts") between two entities: an "entity" can be based on a physical device (e.g., an endpoint device 12 or a physical network device in the secure private core network 10 such as the MCP device 14) having a verified secure relationship with at least an individual person utilizing the physical device; the verified secure relationship also can be with an identified organization associated with the physical device (e.g., a prescribed manufacturer of an endpoint device 12 such as an IoT device, a service provider offering services based on purchase or rental of an endpoint device 12, etc.); the verified secure relationship also can be with another physical device attempting a communication with the physical device (e.g., a physical device executing the MCP device 14 and/or the replicator device 16, another endpoint device 12, etc.). Hence, the secure private core network 10 requires establishment of a strict two-way trusted relationship between two physical devices (also referred to as a "cohort"), where each physical device either is operated by a user, or is a physical device associated with an identified organization (including a corresponding physical device executing the MCP device 14).

Since an individual person (or identified organization) may utilize one or more endpoint devices 12 for network communications, the secure private core network 10 can identify an individual person (or identified organization) based on the allocation of a "federation" identifier (illustrated as "F1") 18 that has a verified secure relationship with one or more physical network devices (e.g., "A" 12, "A1" 12, etc.) that are utilized by the individual person (or identified organization) for communications within the secure data network 5; hence, the secure data network 5 also is referred to herein as a "secure peer-to-peer data network" based on

the trusted aggregation of two-way trusted relationships. As described below, the federation ID 18 is generated by an endpoint device 12 during initial registration of a user (e.g., individual person or identified organization) using a secure random number generator that results in a universally unique identifier (UUID) of at least one-hundred twenty eight (128) bits: an example 128-bit UUID can be implemented as proposed by the Internet Engineering Task Force (IETF) (see RFC 4122).

FIG. 2 illustrates example data structures that can identify secure relationships between different entities, for example different endpoint devices 12, different individual persons or organizations, etc. The secure private core network 10 causes each endpoint device 12 during registration with the secure private core network 10 to securely and randomly generate its own self-assigned 128-bit UUID as a unique endpoint identifier 20: the endpoint ID 20 is stored in a data structure referred to as an endpoint object 22 that stores all attributes associated with the corresponding endpoint device 12 in the secure data network 5. As illustrated in FIG. 2 and as described in further detail below, the secure private core network 10 can cause the endpoint device "A" 12 to generate its own endpoint identifier "E1" 20; the secure private core network 10 also can cause the endpoint device "A1" 12 to generate its own endpoint identifier "E2" 20. The endpoint ID 20 provides a permanent (i.e., unchangeable) cryptographically-unique identity for the endpoint device "A" 12.

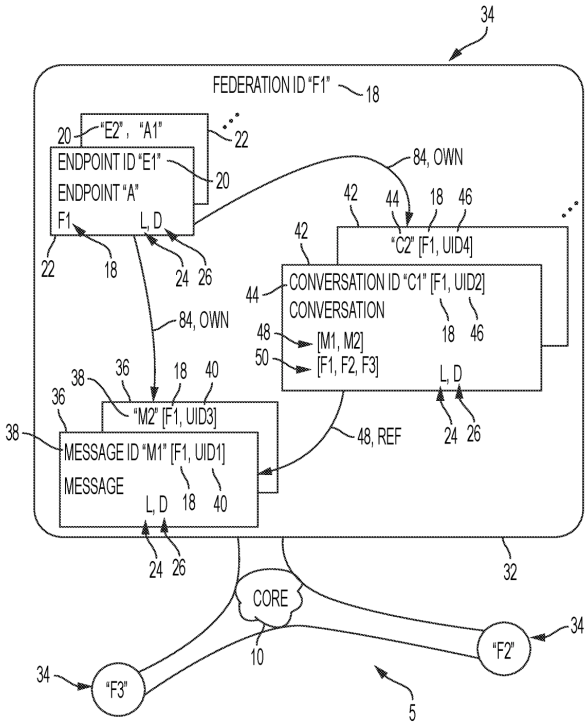


FIG. 2

FIGS. 5A, 5B and 6 illustrate an example identity management system 86 that can be implemented in the secure private core network 10 for secure establishment of trusted relationships in the secure data network 5, according to an example embodiment. A new subscriber "P1" can operate his or her physical network device (88 a of FIG. 6) to cause the processor circuit 92 of the physical network device 88 a to download and install, for example via an external data network 96 distinct from the secure peer-to-peer data network 5, an executable application (e.g., an "app") that includes a desired application (e.g., a messenger application 72 of FIG. 3) and the network operating system (NOS) 56. The new subscriber "P1" as a "requesting party" can enter via the device interface circuit 90 of the physical network device 88 a command that causes the processor circuit 92 to start ("instantiate") the executable application executing the secure private core network operations 56 on the physical network device 88 a as an endpoint device "A" 12, causing an account management service executed in the secure network services 76 to prompt the new subscriber "P1" to register by entering an external network address such as a valid email address of the new subscriber "P1" (e.g., "P1@AA.com"), a mobile number used to receive text-based or image-based messages, etc., where the external network address is used by the requesting party "P1" for reachability via an external data network 96 distinct from the secure peer-to-peer data network 5.

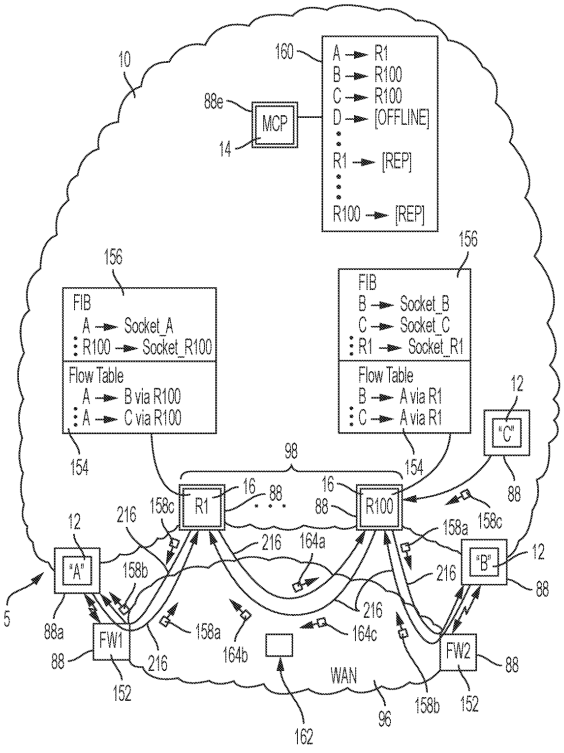


FIG. 7

[Col. 36, Lines 3-20]

Each of the guardian security agent 66, the sentinel security agent 68, and the navigator security agent 70 (e.g., within the endpoint device "A" 12 illustrated in FIG. 9) in operation 196 of FIG. 10B can utilize its local autonomic synchronizer 52 for autonomic synchronization of feature data with a corresponding guardian security agent 66, sentinel security agent 68, and/or navigator security agent 70 that is executed in another two-way trusted network device (e.g., the MCP device 14 or the endpoint device "B" 12). For example, any one of the guardian security agent 66 or sentinel security agent 68 of the endpoint device "A" 12 in operation 196 can execute autonomic synchronization for autonomically exchanging cyber-attack feature data with a corresponding guardian security agent 66 or sentinel security agent 68 in the MCP device 14 (and/or the endpoint device "B" 12), for autonomic aggregation of machine learning-based cyber-attack feature data in the secure peer-to-peer data network.

Publication No.	US20230030990A1		
Title	Peer-to-peer software distribution		
Assignee/Applicant	VMware LLC		
Earliest Priority Date	2021-07-23	Publication Date	2023-02-02

Abstract

Systems and methods are described for performing peer-to-peer software distribution in a Unified Endpoint Management environment. In an example, an unenrolled user device can request enrollment from an enrollment server. The enrollment server can send a list of resources to the unenrolled user device that the unenrolled user device needs based on a group that the unenrolled user device is assigned to. The unenrolled user device can send an identifier of the group to a notification server, and the notification server can respond with an ordered list of enrolled user devices that the unenrolled user device can retrieve the resources from. The unenrolled user device can request the resources from the enrolled user devices on the list until the unenrolled user device receives all the resources from the resource list. The unenrolled user device can then install the resources to complete enrollment.

Description

[0018] Systems and methods are described for **performing peer-to-peer software distribution in a UEM environment. In an example, an unenrolled user device can request enrollment from an enrollment server. The enrollment server can send a list of resources to the unenrolled user device that the unenrolled user device needs based on a group that the unenrolled user device is assigned to.** The unenrolled user device can send an ID of the group to a notification server, and the notification server can respond with an ordered list of enrolled user devices that the unenrolled user device can retrieve the resources from. The unenrolled user device can request the resources from the enrolled user devices on the list until the unenrolled user device receives all the resources from the resource list. The unenrolled user device can then install the resources to complete enrollment.

[0019] FIG. 1 is an illustration of a system for performing peer-to-peer software distribution. **An unenrolled user device ("unenrolled device") 110 and enrolled user devices ("enrolled device") 120 a . . . n can be connected to a local network 130. The local network can include two or more devices connected to each other such that they can exchange electronic communications.** For example**, the local network can include a local area network ("LAN"), a wireless local area network ("WLAN"), a virtual private network ("VPN") internet connection, or two or more devices connected directly, such as through BLUETOOTH, WIFI DIRECT, or near-field communications ("NFC"). The network can include one or more nodes, such a routers or switches, that assign IP addresses to connect devices and route network traffic to and from the connect devices.**

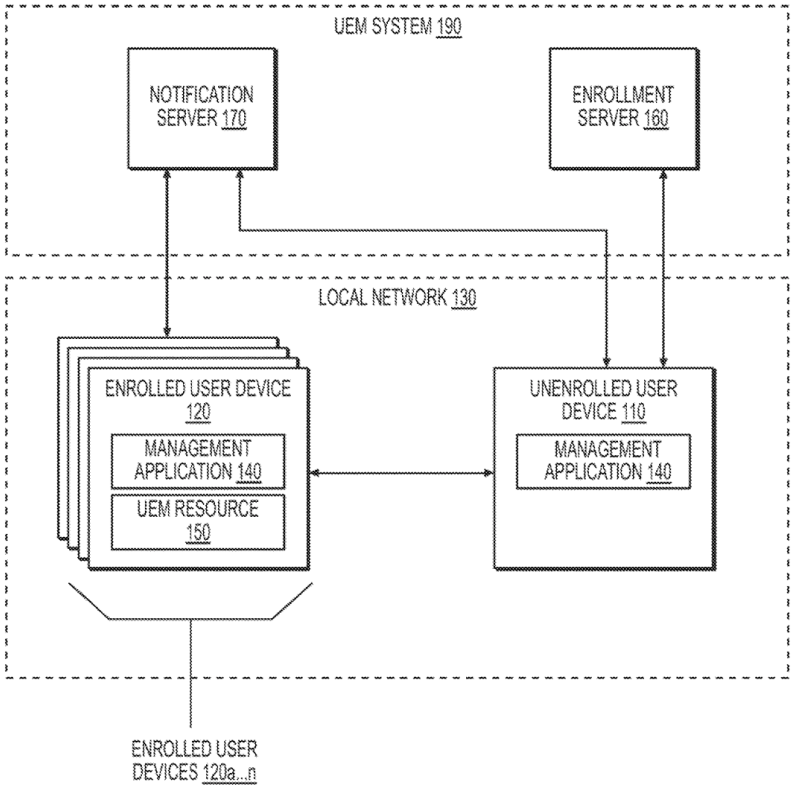


FIG. 1

[0020] The enrolled devices 120 a . . . n can be one or more processor-based devices, such as a personal computer, tablet, or cell phone, that is enrolled in a UEM system 190 or other similar system that manages user devices for an organization. The enrolled devices 120 a . . . n are referred to throughout as just the enrolled device or devices 120 and are meant to include one or more enrolled devices 120, depending on the example.

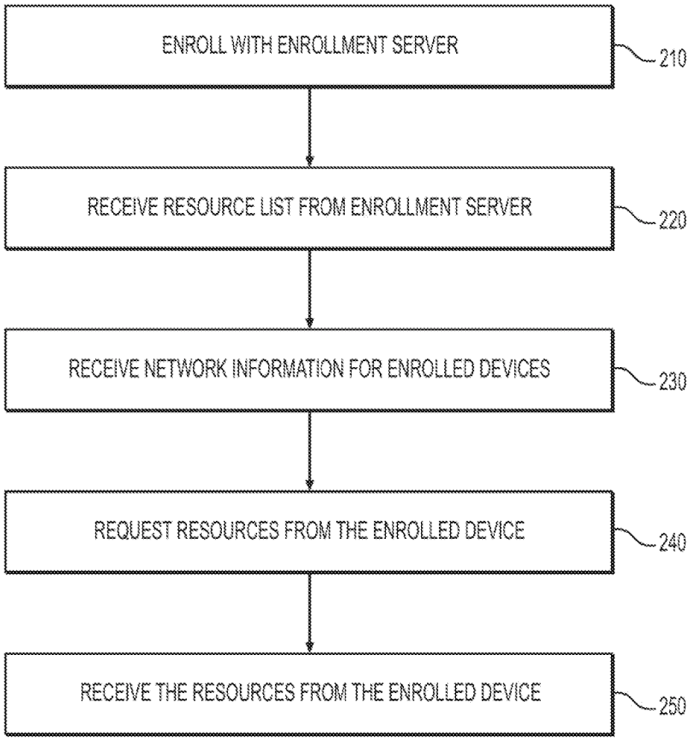


FIG. 2

[0022] In an example, the management application 140 can be responsible for ensuring that the enrolled devices 120 are up to date with compliance and security settings prior to accessing enterprise data and resources. The management application 140 can communicate with the enrollment server 160, allowing UEM management of the enrolled devices 120 based on compliance and security settings at the enrollment server 160. The management application 140 can enforce compliance at the enrolled devices 120, such as by wiping enterprise data when compliance standards are not met. Example compliance standards can include ensuring a device is not jailbroken, that particular encryption standards are used in enterprise data transmission, that the device does not have certain blacklisted applications installed or running, and that the device is located within a geofenced area when accessing certain enterprise resources. In one example, the enrolled devices 120 can access enterprise or UEM resources through the enrollment server 160.

[0043] Some example factors can be based on available computing resources at the enrolled devices 120, such as available CPU power, available network bandwidth, available memory, and remaining battery power. For example, priority can be given to enrolled devices 120 with more available computing resources. This can help prevent enrolled devices 120 from getting overburdened by responding to resource requests from unenrolled devices 110.

[0044] One example factor can prioritize enrolled devices 120 on the same subnet as the unenrolled device 110. This can help ensure that enrolled devices 120 on the local network 130 are prioritized over enrolled devices 120 outside the local network 130, such as enrolled devices 120 that are connected through a VPN. In one example, this factor can be executed by the unenrolled device 110. For example, the unenrolled device 110 can initially skip enrolled devices 120 from the list that are not on the same local network 130. If the unenrolled device 110 is unable to retrieve the resources 150 from the enrolled devices 120 on the same subset, it can then begin to make requests to the other enrolled devices 120.

[0076] In another example, the unenrolled device 110 can deny the request if its available upload bandwidth drops below a predetermined threshold. For example, if the network connection of the first enrolled device 120 a suddenly weakens or the first enrolled device 120 a begins uploading a large file, the upload bandwidth of the enrolled device 120 may be limited. The first enrolled device 120 a can be configured to deny the request because another enrolled device 120 can send the resources 150 at a much faster transfer rate or because accepting the request will overly burden the first enrolled device 120 a.

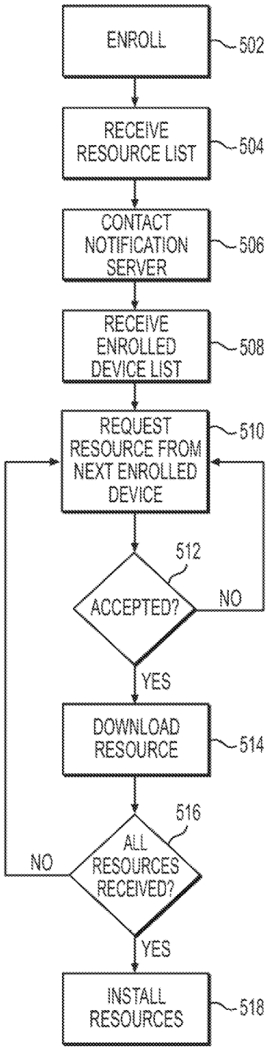


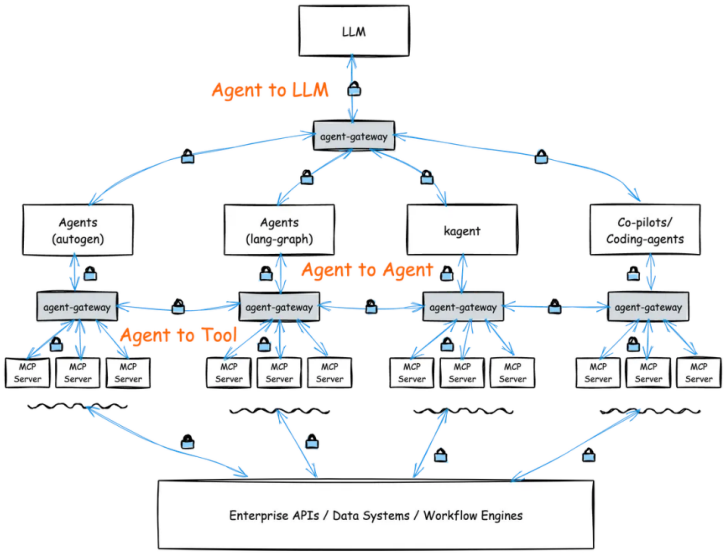
FIG. 5

8. Non-Patent Literature

NPL1	
Title	An Agent Mesh for Enterprise Agents
Author	Solo.io
Abstract	
Learn about the industry's first complete connectivity solution for AI agent ecosystems supports both agent-to-agent and agent-to-tool communication across any environment	
Description	

[Page 2]

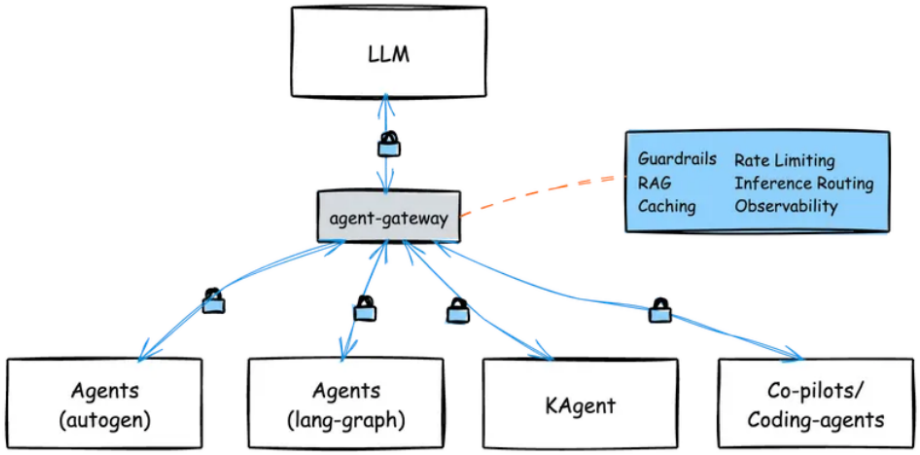
We believe enterprises building and deploying agentic systems will need to solve common security, observability, tenancy, and guardrail concerns. An agent mesh solves these challenges consistently for enterprise agent deployments (self-built, SaaS, developer tools like Cursor, etc) leveraging an agent gateway, a data plane tailored for these use cases.



[Page 5]

Agents individually cannot do interesting things without access to tools. Connecting LLMs to tools is a very open-ended problem which is recently solved with the Model Context Protocol (MCP). Describing tools, exposing them to agents, and allowing them to be invoked in a consistent manner (protocol) regardless of what the backing tool (database, API, file system, SaaS, etc) is addressed by MCP. The challenge with MCP is that it is a nice foundation for RPC communication but it lacks enterprise readiness. Security, discovery,

registration, tenancy and observability are left out of the protocol. Since agents select tools based on their names and descriptions, semantic understanding and guardrailing of these tools is crucial. Overlooking this fact can lead to serious security issues.



[Page 6]

The challenge with building agents isn't just making them smart—it's right-sizing them for the specific workflows they're meant to tackle. **Giving a single, monolithic agent access to too many tools or responsibilities often leads to agent confusion. The agent will hallucinate, wander off-task, or fail entirely. Instead, we're finding that organizations benefit more from smaller, focused agents aligned to well-bounded goals or tasks.** But increasing the number of agents introduces complexity. Agents may need to communicate. How do we ensure observability, traceability, and secure interactions—things like authentication, authorization, and behavioral guardrails?

Google recently announced a new specification for agent-to-agent communication, called the A2A protocol. This spec defines how agents declare their capabilities and skills to coordinate tasks, get task updates, and execute on tasks regardless of the underlying framework used to build them. The protocol publishes skills and capabilities of agents through Agent Cards which are discoverable at runtime and contain both technical and semantic details for calling a specific agent. While the specification mentions transport security and observability, it intentionally keeps these outside of the core protocol. Registration, discovery, and routing are also left out.

9. List of Additional Patents and Non-Patent Literature

Back to Analysis Summary

Related Published Patents and Non-Patent Literature found during this search, which may be of relevance for the client				
S. No.	**Publication/Patent Number**	**Title**	**Assignee**	**Filing Date/Date of Download**
1	US7185199B2	Apparatus and methods for providing secured communication	Xerox Corp.	2002-08-30
2	US20110113481A1	Ip security certificate exchange based on certificate attributes	Microsoft Technology Licensing LLC	2009-11-12
3	NPL	The benefits of using peer-to-peer mesh network technology to overcome connectivity issues	Resilio	2025-09-18

10. Exemplary Search Strings

Back to Analysis Summary

S. No.	Scope	Query	No. Of Hits
1	Patent Literature	(ttl:("endpoint management" OR ("endpoint" NEAR2 management)) AND tac:("PEER-TO-PEER" OR "P2P" OR "PEER TO PEER" OR (peer NEAR2 peer)))	5
2		(ab:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND tac:("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer)))	12
3		desc:(((endpoint OR end?point OR (end NEAR2 point)) NEAR5 (management)) NEAR10 (decentralised OR decentralized OR distributed))	187
4		desc:(((endpoint OR end?point OR (end NEAR2 point)) NEAR5 (security OR vulnerability OR management)) NEAR10 (decentralised OR decentralized OR distributed))	318
5		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:((sensors OR detectors OR transducer OR indicator)))	49
6		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:((orchestrator OR coordinator OR organiser OR administrator OR leader)))	30
7		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:(("Fault tolerant" OR "fail safe" OR redundant OR "error tolerant" OR impervious)))	24
8		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:(((device OR devices) NEAR3 (discovery OR detection OR recognition OR finding))))	25
9		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:(((device OR devices) NEAR3 (availability OR accessibility OR readiness OR convenience OR load OR predictability))))	12
10		(tac:("endpoint management" OR ("endpoint" NEAR2 management) OR ("end point" NEAR2 management)) AND desc:(("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer))) AND desc_en:((task? OR "thread detection" OR vulnerability OR scan OR patching OR patches OR "malware detection")))	61
11		(tac:("endpoint management" OR ("endpoint" NEAR1 management) OR ("end point" NEAR1 management)) AND desc:("PEER-TO-PEER" OR "PEER TO PEER" OR (peer NEAR2 peer)) AND desc:("ai" OR "Artificial intelligence"))	28
12		pn:(US-20110113481-A1 EP-1395019-B1 US-20250004784-A1 US-20250047588-A1 US-20250047690-A1 US-12081558-B2) OR (pcitpnfw:(US-20110113481-A1 EP-1395019-B1 US-20250004784-A1 US-20250047588-A1 US-20250047690-A1 US-	39

		12081558-B2) OR pcitpn:(US-20110113481-A1 EP-1395019-B1 US-20250004784-A1 US-20250047588-A1 US-20250047690-A1 US-12081558-B2))	
13	Non-Patent Literature	“endpoint management” “peer-to-peer”	>200
14		“endpoint management” “distributed security”	>200
15		“endpoint management” “peer to peer” "ai agent"	>200
16		“endpoint agents" "peer to peer"	>200

Data Availability:

- ✓ Documents from non-English countries was restricted to the availability of the text in free and commercial databases (most of the cases only Title and Abstract are available in English)
- ✓ Search was performed with English keywords and their synonyms. Any language other than English such as French, Korean, and Chinese was not considered for the key string.

11. Disclaimer

LexAnalytico has prepared this report based on database and information sources that are believed to be reliable. We disclaim all warranties as to the accuracy, completeness, or adequacy of such information. The patentability search is performed for all published patent and non-patent literature and for all jurisdictions covered by reliable patent and non-patent databases. The above report is prepared based on the search conducted on the keywords and other information extracted from the invention disclosure and subjectivity of the researcher and analyst. The analyst is not intending to provide legal advice in this matter.