

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETSKOPE, INC.,
Petitioner,

v.

FORTINET, INC.,
Patent Owner.

IPR2023-00456
Patent 11,036,856 B2

Before JAMES P. CALVE, THOMAS L. GIANNETTI, and
STEPHEN E. BELISLE, *Administrative Patent Judges*.

CALVE, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. BACKGROUND

Netskope, Inc. (“Petitioner”) filed a petition requesting *inter partes* review of claims 1–28 (the “challenged claims”) of U.S. Patent No. 11,036,856 B2 (“the ’856 patent”) (Ex. 1001). Paper 2, 1 (“Pet.”). Fortinet, Inc. (“Patent Owner”) timely filed a Preliminary Response. Paper 6 (“Prelim. Resp.”).

We have authority to determine whether to institute an *inter partes* review. *See* 35 U.S.C. § 314 (2018); 37 C.F.R. § 42.4(a) (2022). An *inter partes* review may not be instituted “unless . . . the information presented in the petition . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a). Upon consideration of the evidence and arguments in the record, we determine that the information presented shows a reasonable likelihood that Petitioner would prevail with respect to at least one of the challenged claims. We therefore grant institution of an *inter partes* review. *See* 37 C.F.R. § 42.108(a) (“When instituting . . . review, the Board will authorize the review to proceed on all of the challenged claims and on all grounds of unpatentability asserted for each claim.”).

A. Related Proceedings

The parties identify the following proceeding involving the ’856 patent: *Netskope, Inc. v. Fortinet, Inc.*, No. 3:22-cv-01852-TLT (N.D. Cal.). Pet. 1 (noting the case is currently stayed); Paper 3, 2. Petitioner identifies the following *inter partes* review proceedings involving these same parties: IPR2023-00457, IPR2023-00458, IPR2023-00459, IPR2023-00175, IPR2023-00030, and IPR2022-01587. Pet. 1.

B. Real Parties in Interest

Petitioner identifies Netskope, Inc. as the sole real party in interest.

Pet. 1. Patent Owner identifies Fortinet, Inc. as the sole real party in interest.
Paper 3, 2.

C. The '856 Patent (Ex. 1001)

The '856 patent describes systems and methods for continuously scanning and/or sandboxing files residing on cloud storage to protect users of an enterprise network from accessing malware infected files by natively mounting public cloud file stores. Ex. 1001, 1:20–23, 2:41–46.

A public cloud file store is natively mounted on a network security device and includes a first repository in which an untrusted file is stored until the network security device can determine whether the file is clean, i.e., whether it is free of malicious content, by applying security checks to it. Ex. 1001, 2:46–53. When the network security device determines that the file is clean, “the network security device makes the clean file accessible to the users by copying the clean file from the first repository to a second repository that is accessible to the users.” *Id.* at 2:53–57. The network security device copies only those files that pass the security checks to a sanitized storage area and data location that is accessible to users. *Id.* at 3:27–36, 5:45–56. Network security devices provide security to a private network via network firewalling, antivirus intrusion protection, content filtering, data leak prevention, antispam, antispymware, network access control, and the like. *Id.* at 5:4–15. Public cloud file stores can be provided by third-party cloud storage service providers such as Amazon Simple Storage Service (Amazon S3)TM, Microsoft AzureTM, Google Cloud PlatformTM, and IBM CloudTM. *Id.* at 6:23–28.

Figure 1A of the '856 patent is reproduced below to illustrate this arrangement.

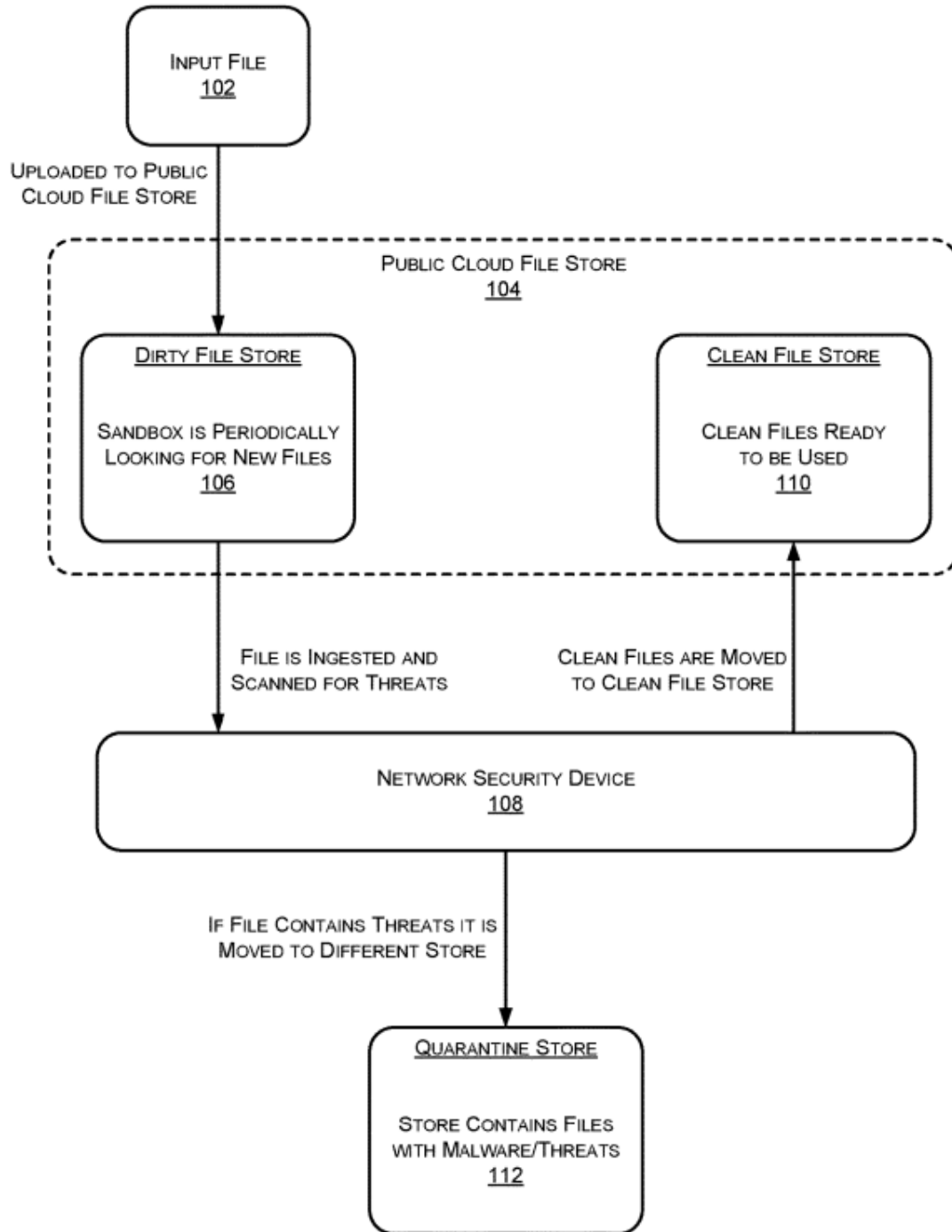


FIG. 1A

Figure 1A of the '856 patent illustrates input file 102 loaded into dirty file store 106 natively mounted on network security device 108 to protect an enterprise network. Ex. 1001, 7:40–57. Clean files are *moved* to clean file store 110. Input file 102 may be an email attachment, a file uploaded from external sources (e.g., a web server), a file from websites, a file from another cloud service, and a file from enterprise network users. *Id.* at 7:59–8:5.

Network security device 108 uses a sandbox/inspection engine to scan untrusted files uploaded into dirty file store 102, which is inaccessible to end users. Ex. 1001, 8:8–17. As input file 102 is stored in dirty file store 106, a notification service of public cloud file store 104 sends a notification via an Application Programming Interface (API) call from a notification service of the public cloud file store to alert network security device 108 to the new file so network security device 108 can perform security checks to determine whether the new file is free of malicious content. *Id.* at 8:17–25, 6:46–55.

If network security device 108 determines input file 102 is free of malware/malicious content, network security device 108 can “*transfer* file 102 from first repository, i.e., dirty file store 106, to a second repository, e.g., clean file store 110, which is accessible to users.” Ex. 1001, 8:26–31 (emphasis added). “[N]etwork security device 108 can make the clean file accessible to the users by *copying and/or transferring* the clean file from dirty file store 106 to clean file store 110 that is accessible to the users.” *Id.* at 8:53–57 (emphasis added). Network security device 108 can be a Unified Threat Management (UTM) appliance, sandbox appliance, firewall, content scanning engine, anti-virus engine, gateway device, and/or virtual machine (VM) controlled by a sandbox appliance. *Id.* at 8:31–36, 6:1–4. Figure 1A indicates that “Clean Files are *Moved* to Clean File Store” (emphasis added).

Clean file store 110 contains only clean files that are available to users of the enterprise network based on sharing rights granted to them. Ex. 1001, 8:58–60. Network security device 108 can remove malware that is detected from the untrusted file so the transfer to the clean file store 110 is made after freeing the file from malicious content. *Id.* at 9:7–12. Files associated with malicious content can be quarantined by storing files in quarantine store 112 that cannot be accessed by users, or files associated with malicious content can be deleted. *Id.* at 9:12–16. In another embodiment, not shown, clean file store 110 may be part of network security device 108 or part of external storage of enterprise networks, rather than a part of public cloud store 104 depicted in Figure 1A, so users can access clean file store 110 seamlessly. *Id.* at 9:17–23, Fig. 1B (“Clean Files are Moved to Clean File Store”).

Figure 2 of the ’856 patent is reproduced below to illustrate secure data transfer system 200 comprising untrusted file processing module 202 and clean file transfer module 204. Ex. 1001, 9:28–35.

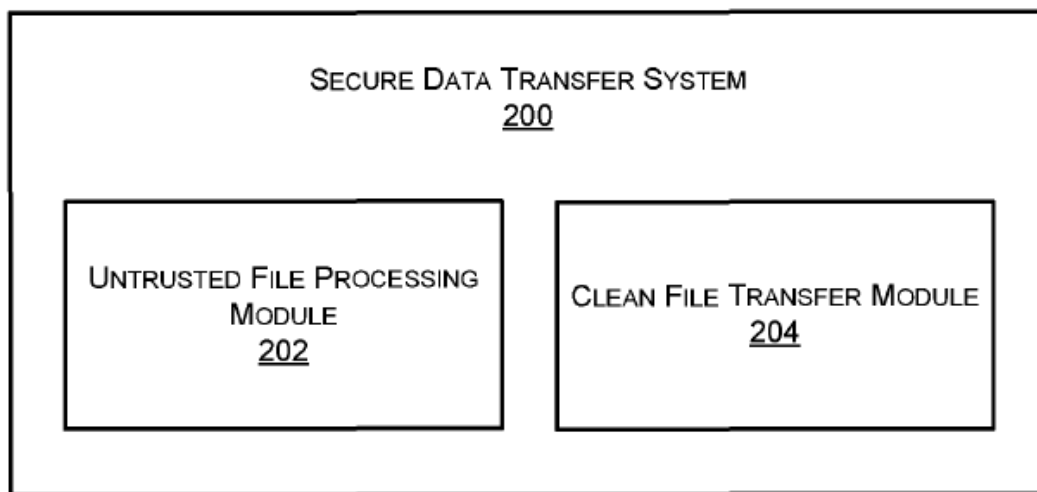


FIG. 2

Figure 2 depicts secure data transfer system 200 in which untrusted file processing module 202 can access an untrusted file in a first repository in a public cloud file store natively mounted on a network security device that implements secure data transfer system 200. Ex. 1001, 9:35–41. When a new file is stored in the first repository, notification is sent to untrusted file processing module 202 via an API call of a notification service of the public cloud file store to alert untrusted file processing module 202 of the existence of an untrusted file so module 202 can apply security checks to the untrusted file. *Id.* at 9:54–63. Security checks include (1) content scanning using any combination of file signature, file hash, file path, file attributes, file source, file pre-scan parameters, file extension, file content, and file name and/or (2) behavioral-based malware detection. *Id.* at 9:63–10:3. Untrusted file processing module 202 can remove any malware that is detected from the untrusted file and quarantine or delete the untrusted file when the results of the determination indicate the presence of malicious content. *Id.* at 10:4–16.

Clean file transfer module 204 makes a clean file accessible to users by “*copying or transferring* the clean file from a first repository (e.g., dirty file store 104) that is inaccessible to end users to a second repository (e.g., clean file store 110) that is accessible to end users” when it is determined to be clean. Ex. 1001, 10:17–23 (emphasis added). Clean files are files found or made free of malware or undesired attributes/behavior. *Id.* at 10:24–27. “[C]lean file transfer module 204 can *copy the file* to the second repository *by sharing it* by means of any or a combination of network file system (NFS) file transfer protocol (FTP), common Internet file system (CIFS), Internet Small Computer Systems Interface (iSCSI), Storage Area Network (SAN), and local storage.” *Id.* at 10:28–33 (emphasis added); *id.* at 6:5–10.

D. Challenged Claims

Petitioner challenges claims 1–28 of the '856 patent. Pet. 1, 3. Claims 1, 13, and 21 are independent. Claims 2–12 depend from claim 1. Claims 14–20 depend from claim 13. Claims 22–28 depend from claim 21. Claim 1 is reproduced below with Petitioner's annotations of its limitations.

1[pre] 1. A secure data transfer system comprising:

1[a] a non-transitory storage device having embodied therein one or more routines operable to prevent users of an enterprise network from accessing malware infected files that are stored within public cloud file stores; and

1[b] one or more processors coupled to the non-transitory storage device and operable to execute the one or more routines, *1[c][i]* wherein the one or more routines include: an untrusted file processing module, which when executed by the one or more processors:

accesses an untrusted file stored within a first repository of a public cloud file store, *1[c][ii]* wherein the public cloud file store is natively mounted on a network security device that is protecting the enterprise network and *1[c][iii]* wherein the users do not have read access to the first repository of the public cloud file store; and

1[d] causes the network security device to make a determination regarding whether the untrusted file is a clean file that is free of malicious content by applying one or more security checks to the untrusted file; and

1[e] a clean file transfer module, which when executed by the one or more processors, makes the clean file accessible to the users by, when a result of the determination is affirmative, *copying* the clean file from the first repository to a second repository that is accessible to the users.

Ex. 1001, 15:2–29 (emphasis added).

E. Asserted Grounds of Unpatentability

Petitioner asserts that the challenged claims are unpatentable on the following three grounds (Pet. 3):

Ground	Challenged Claim(s)	35 U.S.C. §	Reference(s)
1	1–7, 9–18, 20–26, 28	103 ¹	Burdett ² , FortiSandbox ³
2	1, 2, 4, 5, 7–10, 12–14, 16, 18–22, 24, 26–28	103	Sharpe ⁴ , Seese ⁵ , Kamalapuram ⁶
3	3, 6, 11, 15, 17, 23, 25	103	Sharpe, Seese, Kamalapuram, Dods ⁷

Petitioner relies on a Declaration of Dr. Michael Franz. Ex. 1002.

Patent Owner relies on a Declaration of Dr. John Black Jr. Ex. 2001.

¹ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. § 103 that became effective on March 16, 2013. Changes made to 35 U.S.C. § 103 in the AIA do not apply to any application for patent filed before March 16, 2013. Because the ’856 patent has an effective filing date after March 16, 2013, we refer to the AIA version of § 103.

² US 11,281,775 B2, issued Mar. 22, 2022 (Ex. 1004, “Burdett”).

³ FortiSandbox – Administration Guide, Version 2.3.2, dated Nov. 9, 2016 (Ex. 1005, “FortiSandbox”). *See* Ex. 1002 ¶ 66 *for an online address*.

⁴ US 9,811,662 B2, issued Nov. 7, 2017 (Ex. 1006, “Sharpe”).

⁵ US 2015/0100357 A1, published Apr. 9, 2015 (Ex. 1007, “Seese”).

⁶ US 2020/0050686 A1, published Feb. 13, 2020 (Ex. 1008, “Kamalapuram”).

⁷ US 10,382,468 B2, issued Aug. 13, 2019 (Ex. 1009, “Dods”).

II. DISCUSSION

A. Level of Ordinary Skill in the Art

Petitioner asserts that a skilled artisan “would have had at least a B.S. in computer science, computer engineering, or electrical engineering (or equivalent experience) and at least two years’ practical experience with computer networking and computer security.” Pet. 15. Patent Owner asserts a skilled artisan “would have had a Bachelor of Science degree in electrical engineering and/or computer science, and two years of work or research experience in the fields of networks and data security, or a Master’s degree in electrical engineering and/or computer science and one year of work or research experience in related fields.” Prelim. Resp. 13 (asserting that the positions in the Preliminary Response are the same under either proposal). The two formulations are similar, differing mainly in the level of experience required. We adopt Patent Owner’s proposal as more consistent with the ’856 patent and prior art, but our analysis would be the same under either party’s proposal.

B. Claim Construction

We interpret claims “using the same claim construction standard that would be used to construe the claims in a civil action [held] under 35 U.S.C. 282(b).” 37 C.F.R. § 42.100(b). Under this standard, we construe claims “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” *Id.* We construe the claims only to the extent necessary to determine whether to institute *inter partes* review. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

The parties assert that claim terms should receive their plain and ordinary meaning, and no claim terms require construction. Pet. 16; Prelim. Resp. 13–14. Having considered the record, we determine that no express claim construction of any claim term is necessary to reach our decision.

C. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103 if differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. 35 U.S.C. § 103; *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). “The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.” *KSR*, 550 U.S. at 416. Similarly, “if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.” *Id.* at 417.

The question of obviousness is resolved based on underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) when in evidence, objective evidence of non-obviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). Neither party presents any objective evidence of non-obviousness.

D. Ground 1: Alleged Obviousness Over Burdett and FortiSandbox

Petitioner asserts claims 1–7, 9–18, 20–26, and 28 are unpatentable under 35 U.S.C. § 103 over Burdett and FortiSandbox. Pet. 2–3, 16–50.

1. *Burdett (Ex. 1004)*

Burdett discloses a system for protecting public cloud services such as Amazon Web Service (AWS) S3 storage from malware and other exploits as illustrated in Figure 3, which is reproduced below. Ex. 1004, 1:28–53.

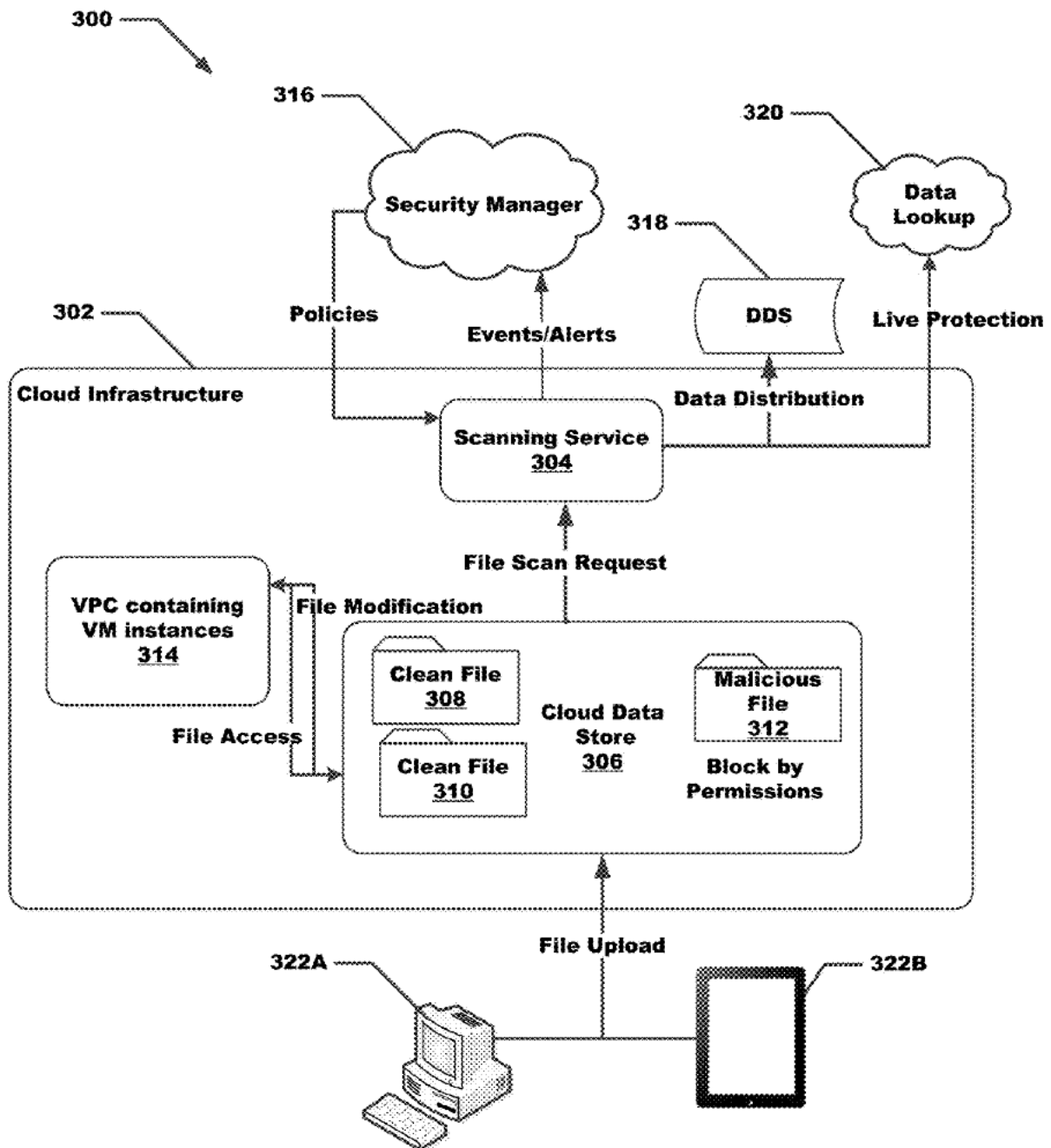


Figure 3 depicts “an agentless scanning service configured for AWS infrastructure [that] uses S3 buckets for file storage, registers with the S3 service to receive notifications of file activity, receives notifications of file activity, and scans files upon receiving the notifications.” Ex. 1004, 2:5–9. Clean files 308, 310 do not require restriction by scanning service 304, but malicious file 312 includes malware or malicious content. *Id.* at 26:49–58.

Scanning service 304 mounts to cloud infrastructure 302 to access cloud data store 304 (Ex. 1004, 23:33–24:10, 27:20–28:22) and is managed and configured by security manager 316 to access files to be scanned and to control file permissions (*id.* at 24:39–25:46). Cloud infrastructure 302 may be provided by AWS and AWS S3 buckets or other “cloud infrastructure services available from Microsoft Azure, CenturyLink Cloud, VMware, Rackspace, Joyent, and Google.” *Id.* at 23:43–56.

Scanning service 304 may be implemented as scanning applications on virtual machines in cloud infrastructure 302 and receive policies from security manager 316 and data updates from Data Distribution Service 318 such as code updates and malicious file definitions. Ex. 1004, 24:32–56. Scanning service 304 accesses files directly from cloud data store 306 and moves files or changes file names to control access. *Id.* at 25:18–28. The scanning service may replace a file with another file that is “clean.” *Id.* at 25:31–33. Scanning service 304 changes file permissions in cloud data store 306 to permit or deny access to files and provide data security with minimal installation and cost. *Id.* at 25:34–46. Scanning service 304 *registers with cloud infrastructure 302 to access files and receive notifications* by setting a role ID, key/password, authentication data, account details, resource rights, secure key store, API credentials, and privileges. *Id.* at 27:24–28:51.

2. FortiSandbox Version 2.3.2 (Ex. 1005)

FortiSandbox discloses a device that “can be configured to sniff traffic from the network, scan files on a network share with a pre-defined schedule, quarantine malicious files, and receive files from FortiGate FortiWeb, FortiMail, and FortiClient.” Ex. 1005, 8. FortiSandbox can scan files stored on a network share and optionally quarantine malicious files. *Id.* at 113.

We reproduce the following instructions from FortiSandbox regarding how to mount a network share to a FortiSandbox device for scanning.

To create a new network share:

1. Go to *Scan Input > Network Share*.
2. Click the + *Create New* button from the toolbar.
3. Configure the following options:

Enabled	Select to enable network share configuration. If network share is not enabled, its scheduled scan will not run.
Network Share Name	Enter the network share name.
Mount Type	Select the mount type from the drop-down list. The following options are available: • CIFS • NFSv2 • NFSv3 • NFSv4

The table reproduced above from FortiSandbox describes creating a new network share by selecting a new network share name and selecting a mount type from a drop-down list. Ex. 1005, 114–115. The following mounting options are available to mount the network share: common Internet file system (CIFS), network file system version 2 (NFSv2), NFSv3, and NFSv4. *Id.* at 115; *see* Pet. 18.

3. *Motivation to Combine*

Petitioner contends that Burdett describes a scanning device that scans an enterprises cloud file store for malware, but Burdett does not detail how the scanning service would access the file store. Pet. 19. Petitioner asserts that FortiSandbox teaches how to configure a device for deployment in an enterprises network infrastructure to scan files on a network share. *Id.*

Petitioner asserts that a skilled artisan would have been motivated to use FortiSandbox's network security device to natively mount the public cloud storage data and implement Burdett's malware scanning services with a reasonable expectation of success because "[c]onfiguring a device like the FortiSandbox system to perform Burdett's malware scans would have enabled an enterprise to preserve the privacy of its files in cloud storage while preventing enterprise users from accessing malicious files." Pet. 19.

4. *Independent Claim 1*

a. *Element 1[pre]*

Petitioner asserts that Burdett discloses "[a] secure data transfer system" including one or more computing devices in a cloud infrastructure that executes a scanning service as a virtual machine to scan files stored in a cloud data store using scanning service 304. Pet. 20–21 (citing Ex. 1004, Abstract, 1:42–53, 23:33–38, 24:19–20, 24:22–25, 24:33–36, 25:18–19, 25:26–28, 25:56–67, 29:42–43, 31:8–14, Fig. 3; Ex. 1002 ¶¶ 71–73). This contention, which Patent Owner does not contest at this stage, is supported sufficiently by evidence of record. Burdett uses scanning service 304 in cloud-based infrastructures like AWS and AWS S3 and other public cloud infrastructures. Ex. 1004, 23:33–56.

b. Elements 1[a], 1[b], and 1[c]

Petitioner asserts that Burdett implements a scanning service in cloud infrastructure service 302 including scanning device 304 with scanning applications running on virtual machines in cloud infrastructure 302 and embodied in a computer readable storage medium to protect an enterprise network and prevent users of that network from accessing malware-infected files in a public cloud data store such as Amazon S3® or Microsoft Azure, CenturyLink Cloud, VMware, Rackspace, Joyent, or Google by accessing untrusted files stored in a first repository of a public cloud file store. Pet. 22–28 (citing Ex. 1004, 2:14–16, 2:19–64, 5:29–31, 7:14–16, 9:29, 11:57–66, 14:18–24, 23:36–38, 23:43–56, 24:22–25, 24:28–31, 24:33–36, 25:18–19, 25:26–28, 25:42–46, 25:55–67, 26:6–28, 26:37–41, 26:44–67, 28:1–5, 28:10–13, 29:31–36, 29:58–61, 31:4–14, 31:24–34, 32:10–67, 34:1–50, Fig. 3; Ex. 1001, 8:8–17; Ex. 1002 ¶¶ 74–83).

Patent Owner argues that Petitioner’s attempt to combine the Figure 1 embodiment of Burdett for an “enterprise network” with Figure 3 for a “public cloud file store” does not show that “Burdett’s Figures 1 and 3 describe the same embodiment (they do not) or show[] that it would be obvious to combine these embodiments.” Prelim. Resp. 27. Patent Owner argues that Petitioner assumes that Burdett’s teachings of preventing users from accessing malware infected files in a cloud repository of Figure 3 apply to Burdett’s enterprise network of Figure 1, but “[n]one of Petitioner’s cited portions of Burdett state that the ‘scanning service 304’ of Figure 3 can be used to restrict access to files by the ‘networked components’ of ‘enterprise facility 102’ of Figure 1.” *Id.* at 28–29. For the reasons that follow, Patent Owner’s arguments are unavailing at this stage.

Figures 1 and 3 depict the same threat management system protecting the enterprise from external threats. Figure 1 depicts the system protecting enterprise users from threats 104, 108 over Internet 154 in “a first particular environment.” *Id.* at 4:55–56, 5:28–31, 5:57–6:43. Figure 3 depicts “an example system 300” and “block diagram of an implementation” protecting enterprise users from threats in cloud 302. *Id.* at 1:28–53, 4:58, 23:33–56. Enterprise administrators manage the threat system settings in both of these exemplary environments in which the enterprise threat management system is used. *Id.* at 2:1–4, 6:25–29, 23:39–42, 23:57–59, 24:41–44, 29:50–52.

c. Element 1[c][ii]

Petitioner contends that Burdett does not describe a “public cloud file store is natively mounted on a network security device that is protecting the enterprise network,” but FortiSandbox natively mounts a network share to a FortiSandbox security system using protocols (specifically including CIFS, NFSv2, NFSv3, and NFSv4) that allow the FortiSandbox to communicate with the file system and network share in the file system’s native language. Petitioner contends a skilled artisan would mount Burdett’s public cloud file store (cloud infrastructure 302) to FortiSandbox’s system using these native mounting file transfer protocols so that the system would scan files stored on Burdett’s public cloud. Pet. 28–29 (citing Ex. 1004, 2:5–17; Ex. 1005, 8, 10, 113–115; Ex. 1001, 11:7–10, 11:66–12:4; Ex. 1002 ¶¶ 85–89). Petitioner contends that “FortiSandbox’s use of mount types including CIFS and NFS protocols, which are native file system protocols, is consistent with the ’856 patent’s discussion of ‘natively mounting’ a file store on a network security device” *using NFS and CIFS transfer protocols*. Pet. 29 (citing Ex. 1001, 11:7–10 (a data store is mounted as NFS, and files are accessed via CIFS).

Patent Owner argues that Petitioner fails to show that FortiSandbox’s “network share” or “remote file share” corresponds to a “*public* cloud file store.” As a result, natively mounting a FortiSandbox to such a remote file share does not disclose “natively mount[ing] a *public* cloud file store” as claimed. Prelim. Resp. 20–21. Patent Owner argues that using CIFS and NFS protocols for mount types in FortiSandbox is not “natively mount[ing]” a “public cloud file store” because the ’856 patent discloses that specific software is used to mount a public cloud file store such as Amazon Web Services (AWS) Simple Storage Service (S3) bucket rather than relying solely on generic protocols such as NFS and CIFS. *Id.* at 21–22.

Patent Owner’s arguments that FortiSandbox’s network share is not a public cloud file store are unavailing as they attack references individually and overlook Petitioner’s contentions for element 1[a] that Burdett mounts scanning service 304 to public cloud file stores (e.g., Amazon S3, Microsoft Azure, CenturyLink Cloud, VMware, Rackspace, Joyent, and Google) that would mount natively to FortiSandbox’s system with NFS and CIFS as they do to scanning service 304. Pet. 22–25, 28–30. The ’856 patent describes public cloud file stores provided by such third-party cloud storage service providers as Amazon Simple Storage Service (Amazon S3)TM, Microsoft AzureTM, Google Cloud PlatformTM, and IBM CloudTM. Ex. 1001, 6:23–28.

As discussed *supra*, Petitioner contends that FortiSandbox natively mounts a public file store on a network security device using CIFS, NFSv2, NFSv3, and NFSv4 so a FortiSandbox system can communicate with the file system in its native language, just as the ’856 patent mounts public cloud file stores to network security devices using NFS and CIFS file protocols. *See* Pet. 28–29 (citing Ex. 1005, 114–115; Ex. 1001, 11:7–10; Ex. 1002 ¶ 85).

These contentions are consistent with the ordinary and customary meanings of “mount” as “[t]o make a physical disk or tape accessible to a computer’s file system” (Ex. 1010, 4 (*Microsoft® Computer Dictionary Fourth Edition*)) and “native” as “the format the application uses internally is its native file format” (*id.* at 5). *See* Pet. 9, 63–64.

They also are consistent with the ’856 patent’s description that files are mounted natively to network security devices via NFS and CIFS. Pet 29 (citing Ex. 1001, 11:7–10). Petitioner asserts that FortiSandbox mounts file shares to network security devices using CIFS or NFSv2–4 so clean files can be copied between repositories in native languages, just as the ’856 patent uses NFS and CIFS to mount a public cloud data store to a network security device and copy clean files between repositories. *Id.*; *see* Ex. 1001, 10:28–33 (clean file transfer module 204 can copy a file to the secondary repository by means of NFS, FTP, CIFS, iSCSI, SAN, and local storage).

Patent Owner’s arguments that these protocols do not natively mount *public* cloud file stores because the ’856 patent describes AWS as providing “specific software” to mount AWS S3 buckets “as opposed to relying solely on generic protocols such as NFS and CIFS” (Prelim. Resp. 21–22) are not availing. *No details* are claimed for natively mounting a public cloud file store to a network security device. Nor is any “specific software” claimed. Moreover, Burdett mounts scanning service 302 and security manager 316 to public cloud store 302 with similar settings as those Patent Owner cites in FortiSandbox 4.2.4. Ex. 1004, 23:33–24:18, 27:24–28:41; Ex. 2003, 74–75, 100–101; Prelim. Resp. 22–23. At this stage, we find Petitioner has shown sufficiently that Burdett and FortiSandbox teach or suggest “the public cloud file store is natively mounted on a network security device.”

d. Element 1[e]

Petitioner asserts that Burdett discloses “a clean file transfer module which, when executed by the one or more processors, makes the clean file accessible to the users by . . . copying the clean file from the first repository to a second repository that is accessible to the users.” Petitioner contends:

Burdett taught or suggested this limitation. After a file was scanned by Burdett’s scanning service and deemed clean, Burdett explained that in “***some implementations,***” the file was made accessible to the user by changing the file permission. (Burdett, 25:64-67.) Burdett, however, further described other interchangeable mechanisms to control access to a file, including changing a file name of the file, or ***changing the storage location of the file***: “In some implementations, ***the scanning service may move files*** or change the name of files ***in order to control access.***” (*Id.*, 25:26-27.)

. . .

In light of Burdett’s disclosure, it would also have been obvious to copy a clean file to a second repository that is accessible to the applications. Additionally, by making a file accessible to one or more applications, a POSITA would have understood that the file would be unavailable to users of those applications. (Franz, ¶96.)

Pet. 32–35 (citing Ex. 1004, 25:26–27, 25:64–67, 26:59–67, 31:4–14, 31:24–34, 32:10–15, Fig. 3; Ex. 1002 ¶¶ 94–97). Dr. Franz testifies that moving a file involves copying it first (Ex. 1002 ¶ 96) (emphasis added):

In light of Burdett’s disclosure of at least moving a file to control access by applications, it would have been obvious to copy a clean file to a second repository that is accessible to the applications. *A POSITA would have found it obvious that “moving” files in many circumstances (such as moving from one physical storage device to another) actually reduces to first copying the file to the new location and then deleting the original.*

Patent Owner argues, “Petitioner ignores the claim language as written and alleges that Burdett’s ‘*mov[ing]*’ of a file somehow meets the ‘copying’ limitation without providing any analysis to justify Petitioner’s departure from the clear language of the claims.” Prelim. Resp. 14. Patent Owner asserts that claims 1, 13, and 21 recite “copying the clean file from the first repository to a second repository,” but claim 12 recites “moving the untrusted file from the first repository to a third repository” for checking. *Id.*

Patent Owner also argues that “copying” and “moving” have different meanings and require different actions. Prelim. Resp. 14. Patent Owner argues, “[t]he plain meaning of ‘copying’ information requires creating at least temporarily [a] second copy of portions of that information [whereas] the plain meaning of ‘moving’ merely requires changing the location of the information independent of any particular mechanism for doing so.” *Id.* at 14–15 (citing Ex. 2001 ¶46). Patent Owner’s declarant, Dr. Black, testifies:

46. In my opinion, Petitioner has not shown that Burdett’s disclosure of “moving” a file discloses or renders obvious “copying” a file. All independent claims of the ’856 patent require “*copying* the clean file from the first repository to a second repository.” In contrast, dependent claims 12 requires “*moving* the untrusted file from the first repository to a third repository” for checking. The terms “copying” and “moving” have different meanings and require that different actions be performed on files—copying vs. moving. “[C]opying” information requires creating, at least temporarily, a second copy of portions of that information. “[M]oving” merely requires changing the location of the information within a file system, regardless of what mechanism is used. A POSITA would understand that, in a file system, moving and copying a file are different actions.

Ex. 2001 ¶ 46.

We find Patent Owner’s argument unavailing. According to the ’856 patent specification, “copying the clean file from the first repository to a second repository that is accessible to the users” can be performed by *sharing* the files using NFS, FPT, CIFS, and other file transfer protocols.

[T]he network security device can *copy the file* to the second repository *by sharing it* by means of any or a combination of network file system (NFS) file transfer protocol (FTP), common Internet file system (CIFS), Internet Small Computer Systems Interface (iSCSI), Storage Area Network (SAN), and local storage.

In another aspect, clean file transfer module 204 can *copy the file* to the second repository *by sharing it* by means of any or a combination of network file system (NFS) file transfer protocol (FTP), common Internet file system (CIFS), Internet Small Computer Systems Interface (iSCSI), Storage Area Network (SAN), and local storage.

Ex. 1001, 6:5–10 (emphasis added); 10:28–33 (emphasis added).

The ’856 patent claims this type of copying, which contradicts Patent Owner’s arguments that copying does not include moving or sharing files. Independent claims 1, 13, and 21 each include a dependent claim that recites this type of copying by *sharing* files. *Id.* at 15:43–49 (claim 5), 16:47–52 (claim 16), 18:4–10 (claim 24). Claim 5 is exemplary and recites:

5. The secure data transfer system of claim 1, *wherein the network security device copies the file to the second repository by sharing it* by means of any or a combination of network file system (NFS), file transfer protocol (FTP), common Internet file system (CIFS), Internet Small Computer Systems Interface (iSCSI), Storage Area Network (SAN), and local storage.

Ex. 1001, 15:43–49 (emphasis added).

Reciting this feature in claims depending from independent claims 1, 13, and 21 means that the scope of claim 1 includes this subject matter and therefore demonstrates that copying a file includes sharing or moving files by file transfer protocols. *See Littelfuse, Inc. v. Mersen USA EP Corp.*, 29 F.4th 1376, 1380 (Fed. Cir. 2022) (“By definition, an independent claim is broader than a claim that depends from it, so if a dependent claim reads on a particular embodiment of the claimed invention, the corresponding independent claim must cover that embodiment as well.”).

The patent ascribes no significance to copying or moving files. It uses both methods and applies the terms “copying,” “sharing,” “transferring,” and “move” *interchangeably*. Ex. 1001, 2:54–57 (*copying* a clean file to second repository), 3:24–36 (*copying* clean files to sanitized storage area), 5:52–57 (*copying* clean file from first to second repository), 6:5–10 (*copying* file by *sharing* via NFS, FTP, CIFS), 6:59–62 (*copying* a clean file from first to second repository), 8:53–57 (*copying and/or transferring* clean file from dirty file store 106 to clean file store 110), 10:28–33 (*copying* file by *sharing* via NFS, FTP, CIFS), 11:64–12:7 (clean file is *moved* to clean bucket).

As a further example, Figure 1A of the ’856 patent describes this process as “Clean Files are *Moved* to Clean File Store” (emphasis added). The description of Figure 1A states, “upon determining that the file at issue is clean, that is, free from malware, network security device 108 can make the clean file accessible to the users by *copying and/or transferring* the clean file from dirty file store 106 to clean file store 110 that is accessible to the users.” Ex. 1001, 8:53–57 (emphasis added); *id.* at Fig. 1B (“Clean Files are *Moved* to Clean File Store” (emphasis added)). Once again, the ’856 patent uses the terms “moving,” “copying,” and “transferring” *interchangeably*.

Burdett discloses this type of copying, as Petitioner asserts. Pet. 33. Burdett’s network security scanning service “may move files or change the name of the files” to control access or may “replace a file with another file that is ‘clean.’” Ex. 1004, 25:26–33; *see* Pet. 33. The scanning service may transfer a file by changing its name or location (e.g., its path in a file system) to control access. Ex. 1004, 25:28–33. Burdett’s moving and transfer of files to control access to the files performs such copying. Ex. 1004, 25:26–33, 26:55–58, 27:24–29; *see id.* at 29:30–41, 30:57–62; *cf.* Prelim. Resp. 15 (changing a file’s location or the file system structure is “moving” the file).

e. Elements 1[c][iii] and 1[d]

Petitioner contends that Burdett limits read access by users to files stored in a first repository of the public cloud file store and also determines whether an untrusted file is a clean file that is free of malicious content by applying security checks (e.g., anti-virus or anti-malware scans) to the untrusted file. Pet. 31–32 (citing Ex. 1004, 1:64–66, 8:1–12, 24:51–56, 25:12–13, 25:26–27, 25:59–67, 29:42–52, 29:58–61, 31:4–14, 31:24–34; Ex. 1002 ¶¶ 92–95). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

f. Other Arguments

Regarding Patent Owner’s assertion that Petitioner has not shown a reasonable expectation of success (Prelim. Resp. 24), “[t]he reasonable expectation of success requirement refers to the likelihood of success in combining references to meet the limitations of the claimed invention.” *Intelligent Bio-Systems, Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1367 (Fed. Cir. 2016). Absolute predictability of success is not required. *See Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007).

Given the broad scope of natively mounted, the disclosure that AWS provides software to natively mount files (Ex. 1001, 5:23–32), and Burdett’s similar native mounting of cloud files to a scanning service (*see* Ex. 1004, 23:33–24:18, 25:34–55, 27:20–29:41), Petitioner has shown a reasonable expectation of success in combining Burdett and FortiSandbox to meet this limitation. *See Uber Techs., Inc. v. X One, Inc.*, 957 F.3d 1334, 1339 (Fed. Cir. 2020) (“The specification of the [patent-at-issue] is entirely silent on how to transmit user locations and maps from a server to a user’s mobile device, suggesting that a person of ordinary skill in the art was more than capable of selecting between the known methods of accomplishing this.”). Descriptions of native mounting in the ’856 patent, FortiSandbox, and Burdett indicate that native mounting can be accomplished with a reasonable expectation of success. *In re Epstein*, 32 F.3d 1559, 1568 (Fed. Cir. 1994) (“[T]he Board’s observation that appellant did not provide the type of detail in his specification that he now argues is necessary in prior art references supports the Board’s finding that one skilled in the art would have known how to implement the features of the references.”).

Regarding the motivation to combine, FortiSandbox natively mounts network shares to security devices to scan suspicious files for malware (Ex. 1005, 113–115; Pet. 18–19), and Petitioner contends that this technique would improve Burdett similarly to perform malware scans of public cloud store files via native communications while preventing users from accessing malicious files (Pet. 19, 29–30). *See KSR*, 550 U.S. at 417.

g. Conclusion – Claim 1

On this record, we find Petitioner has shown a reasonable likelihood that claim 1 would be unpatentable over Burdett and FortiSandbox.

5. *Independent Claims 13 and 21*

Petitioner provides contentions for how the combination of Burdett and FortiSandbox teaches or suggests the limitations of claims 13 and 21, and Petitioner supports its contentions with testimony of Dr. Franz. Pet. 35–38. Patent Owner argues that Petitioner’s contentions do not address “protecting an enterprise network” in claim 13 or “users of an enterprise network are protected by the network security device” in claim 21 and Petitioner’s theories for claims 13 and 21 are analogous to Petitioner’s theories for claim 1. Prelim. Resp. 30–31. As discussed above for claim 1, we find Petitioner’s contentions and evidence sufficient for purposes of institution. We have reviewed Petitioner’s evidence and contentions for claims 13 and 21, and, at this stage, we find them sufficient. Petitioner identifies where Burdett uses an API of the cloud infrastructure service or a cloud data store to send a notification to scanning server when files are uploaded, downloaded, or changed in the cloud data store. Pet. 35–36. On this record, we find that Petitioner has shown a reasonable likelihood that claims 13 and 21 would be unpatentable over Burdett and FortiSandbox.

6. *Dependent Claims 2–7, 9–12, 14–18, 20, 22–26, and 28*

Petitioner provides detailed contentions with a detailed mapping of how Burdett and FortiSandbox teach or suggest the limitations of dependent claims 2–7, 9–12, 14–18, 20, 22–26, and 28 with supporting testimony of Dr. Franz. Pet. 38–50. These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

On this record, we find that Petitioner has shown a reasonable likelihood that claims 2–7, 9–12, 14–18, 20, 22–26, and 28 would have been unpatentable over Burdett and FortiSandbox.

E. Ground 2: Alleged Obviousness Over Sharpe, Seese, and Kamalapuram

Petitioner asserts that claims 1, 2, 4, 5, 7–10, 12–14, 16, 18–22, 24, and 26–28 are unpatentable under 35 U.S.C. § 103 over Sharpe, Seese, and Kamalapuram. Pet. 2–3, 50–79.

1. Sharpe (Ex. 1006)

Sharpe co-locates cloud controller 4208 with a cloud storage system 302 to perform anti-virus scans of data written to the distributed file system as illustrated in Figure 42, which is reproduced below. Ex. 1006, 97:17–20.

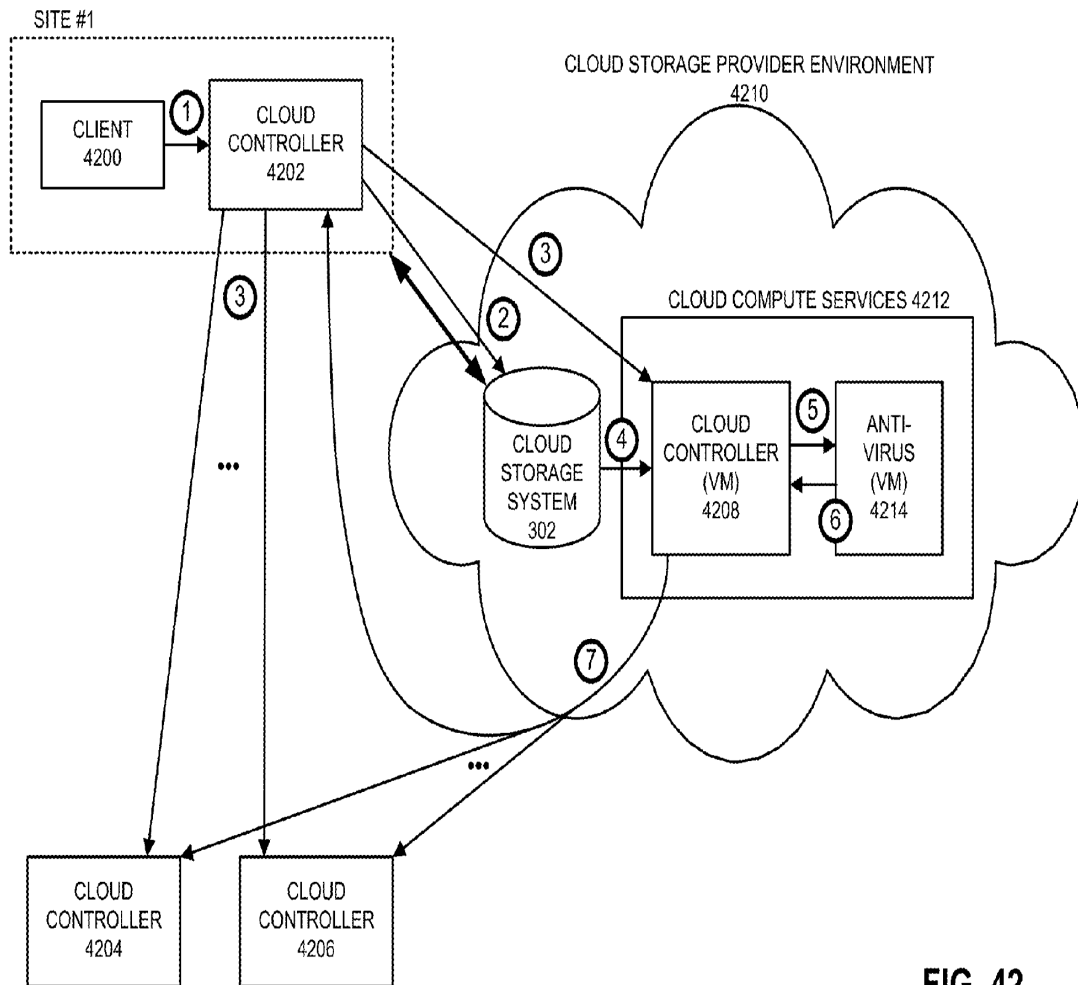


FIG. 42

Figure 42 illustrates cloud controller 4208 configured to access cloud storage system 302 to download files that are not marked as clean and send the files to anti-virus service 4214 to perform anti-virus scans. Ex. 1006, 97:37–44. If a scan indicates the file is clean, the clean field in the metadata is updated to indicate this clean status. *Id.* at 97:56–58. If a file is found to be infected, cloud controller 4208 may remove the file, quarantine the file, and/or mark the file infected and mark the file metadata appropriately. *Id.* at 97:56–62.

2. *Seese (Ex. 1007)*

Seese discloses an integrated storage system that performs scans of files stored on a cloud service 104A on demand or as scheduled periodically (e.g., daily, weekly, biweekly and monthly). Ex. 1007 ¶¶ 63, 64. Data Loss Prevention (DLP) server 404 mounts a first user share in steps 2 and 3 to execute discover scan tasks. *Id.* ¶¶ 66, 67. After mounting the user share, DLP discover server 404 pulls files from the user share on cloud service 104A and inspects the files against configured policies and rules to generate an incidents report, as appropriate. *Id.* ¶ 68. Enforce server 402 and DLP discover server 404 may form part of the computer systems of enterprise 102. *Id.* ¶ 68.

We reproduce Figure 4 of Seese below to illustrate such an integrated storage system that performs on-demand and scheduled scans of files stored by a cloud service that is mounted natively to DLP server 404. *See* Ex. 1007 ¶¶ 63–67.

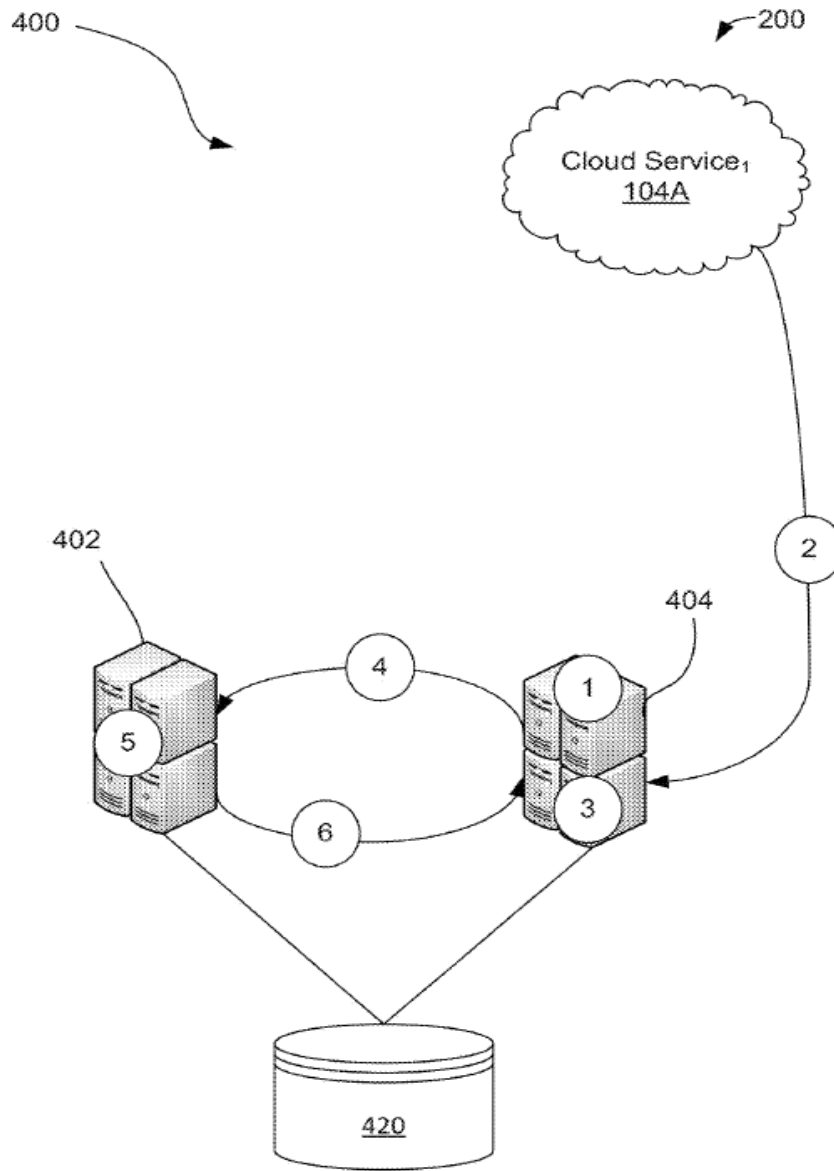


Figure 4 depicts integrated storage system 400 with 104A natively mounted to DLP server 404. Ex. 1007 ¶¶ 63–67.

3. *Kamalapuram (Ex. 1008)*

Kamalapuram discloses a cloud storage system and security analysis system that stores files in a view only mode in an isolated area of network storage until files have been scanned and then moves the files to a general storage area after they have been scanned. Ex. 1008 ¶¶ 121, 122.

4. *Independent Claim 1*

a. *Element 1[pre]*

Petitioner contends that Sharpe discloses a secure data transfer system executing computing devices as a cloud controller that can be implemented as a virtual machine or physical device coupled to an anti-virus service that performs anti-virus scans on every file written to the distributed file system or modified. If a scan indicates a file is clean, a “clean” field in metadata of the file is updated to indicate its clean status so cloud controllers can serve files to a client. Pet. 53–55 (citing Ex. 1006, 2:32–40, 2:53–61, 96:54–66, 97:14–25, 97:32–36, 97:48–62, 97:67–98:2, 98:28, 98:43–46, Fig. 42; Ex. 1002 ¶¶ 141–144). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

b. *Element 1[a]*

Petitioner asserts that Sharpe performs anti-virus scanning operations by executing instructions in a non-transitory computer-readable storage medium for an enterprise network to prevent users of the enterprise network from accessing malware-infected files stored in public cloud file stores. Pet. 55–59 (citing Ex. 1006, 2:19–31, 6:42–7:3, 9:34–48, 96:54–66, 97:17–36, 98:1–8, 98:2–32, 100:60–62, 105:9–10, Fig. 42; Ex. 1007 ¶¶ 16, 64; Ex. 1002 ¶¶ 36, 79, 145–152). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record. Sharpe’s system includes anti-virus capabilities that attempt to prevent infected files from being stored to the system and scan files already stored to the system so that every file written to the system undergoes an anti-virus scan. Ex. 1006, 96:42–61. The metadata for files uses a “clean” field to indicate a file is safe to write to the cloud storage system. *Id.* at 96:61–97:7.

c. Elements 1[b] and 1[c][i]

Petitioner contends that Sharpe's cloud controller is implemented as a virtual machine or physical system using a dedicated or shared processor to execute a software module or code to access an untrusted file on the cloud storage system in a first repository, and Kamalapuram's security analysis system uploads files to an isolated area of the network cloud storage system until it is scanned. Pet. 59–61 (citing Ex. 1006, 6:62–7:3, 96:43–58, 97:32–51, 98:1–8, 98:28–30, 105:9–13, Fig. 42; Ex. 1008 ¶¶ 58, 121, 122; Ex. 1002 ¶¶ 153–158). Petitioner contends it would have been obvious to store files in Sharpe's cloud storage system in a repository to isolate files prior to an anti-virus scan to achieve Sharpe's goal of ensuring that infected files could not be accessed. *Id.* at 61. These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

d. Element 1[c][ii]

Petitioner contends that Sharpe mounts a distributed filesystem to a controller to allow access to contents of the filesystem as if it was an NFS filesystem to enable full, transparent access to data stored in the distributed filesystem, and Seese mounts a public cloud file system to a network security device (data loss prevention (DLP) server) such that a skilled artisan would have natively mounted a public cloud file store of Seese or Sharpe to such a network security device by using NFS, CIFS, or another filesystem protocol to enable the server to use native cloud APIs to detect directory changes in the user share promptly and to control access to files and protect the enterprise against data loss. Pet. 61–64 (citing Ex. 1006, 25:38–42, 61:11–22, Figs. 31, 42; Ex. 1007 ¶¶ 63, 64, 67, 68; Ex. 1002 ¶¶ 159–164).

Patent Owner argues that Sharpe does not natively mount a public cloud file store on a network security device for scanning files because the alleged native mounting in Figure 31 is local storage pool 3108 of file blocks loaded from cloud storage system 302 and cached in storage pool 3108 such that native mounting is for the local storage pool 3108. Prelim. Resp. 34–36 (citing Ex. 2001 ¶ 77). Patent Owner asserts that Seese mounts a cloud service to a network security device, but Petitioner does not establish that the cloud service is a *public* cloud file store. *Id.* at 36–37 (citing Ex. 2001 ¶ 78).

At this stage of the proceeding, we find that Petitioner has accounted sufficiently for this limitation. Sharpe discloses that the NFS mount point 3110 presents the contents of the distributed file system to server 3100 and other external device as if the file system was an NFS file system to enable full, transparent access to data stored in the distributed file system. The blocks requested via this native mount interface 3110 are loaded from cloud storage system 302 and cached in storage pool 3108 so other services can access the distributed file system *and* the data blocks of the distributed file system that are cached 3112 in storage pool 3108 via exported mount point 3110. Ex. 1006, 61:11–32. This disclosure indicates that services executed on server 3100 can access the distributed file system and its files and data blocks that are cached in storage pool 3108 as if these files and data blocks are part of the same NFS native system, i.e., as if they are natively mounted. Even if some files of the natively mounted distributed file system are made available to clients/services on server 3100 via a cache storage pool 3108, such presentation does not negate the disclosure in Sharpe that the natively mounted files also are available for further processing and use such that the files can be considered to be “accessible to the users” as claimed.

Sharpe discloses that each cloud controller 3102 stores and updates metadata indicating the file and directory layout of a distributed file system and location of the data blocks *in* the cloud storage system, which can be provided by an external vendor who may be prevented from accessing the data by encrypting the cloud files. Ex. 1006, 9:22–41. Petitioner reasonably contends that the external vendor provides a *public* cloud file store, which is consistent with *Sharpe’s encryption of data* stored on the external vendor’s cloud storage system to enhance its security. *Id.* at 9:36–48; Pet. 58.

Furthermore, Petitioner and its declarant Dr. Franz contend that Seese natively mounts a cloud service 104A that may be public, private, or hybrid, as all three types of cloud storage systems are “traditional” to perform a DLP process for files stored by the cloud services by mounting user shares of the files stored in the cloud service and scanning the mounted shares against policies and rules. Ex. 1007 ¶¶ 4–6, 63–68, Fig. 4; Ex. 1002 ¶¶ 159–164; Pet. 62–64. Petitioner and Dr. Franz contend that a skilled artisan would have been motivated to natively mount a public cloud file store of Sharpe on a network security device taught by Seese using well-known NFS, CIFS, and other file system protocols to enable the server to use native cloud APIs to detect directory changes in the user share promptly, to protect the enterprise against data loss by applying data loss prevention rules, and to natively communicate with the cloud file store to control access to the files as the references teach. Pet. 63–64; Ex. 1002 ¶¶ 163, 164. On this record, we agree. Using a known technique to improve similar devices predictably in the same or similar way is likely obvious unless its application is beyond the level of ordinary skill in the art. *KSR*, 550 U.S. at 417–18. Patent Owner does not address the rationales provided by Petitioner. Prelim. Resp. 38–40.

Petitioner's rationale improves Sharpe's cloud controllers by using Seese's teachings to natively mount cloud file stores to a network security device such as a DLP processor so the files can be scanned to prevent data loss. Furthermore, the native mounting makes the user share files (public cloud storage system) accessible through the operating system that executes on the DLP server and use native cloud APIs to detect directory changes and control access to files. Pet. 62–64; Ex. 1007 ¶ 67; *cf.* Prelim. Resp. 40.

e. Element 1[c][iii]

Petitioner contends that Sharpe denies read access to files in a cloud storage system until they are scanned by an anti-virus service and a clean metadata field is marked, and Kamalapuram stores files in an isolated area of a network storage system before they are scanned by a security analysis system to access to the files until determined to be clean so this repository would have improved Sharpe by hiding files until scanned and approved for access. Pet. 64–65 (citing Ex. 1006, 96:64–66, 97:32–36; Ex. 1008 ¶ 122; Ex. 1002 ¶¶ 165–166). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

f. Element 1[d]

Petitioner contends that Sharpe checks untrusted files for malicious content and conveys the scan result to a cloud controller to indicate if files are clean or infected, and Seese scans files for policy compliance so that the combined functions would improve data security predictably. Pet. 66–67 (citing Ex. 1006, 2:26–31, 2:56–61, 6:62–7:3, 96:54–58, 97:48–53, 98:1–2, 98:5–8, 98:28–30, 98:40–46, 105:9–13, Fig. 42; Ex. 1007 ¶¶ 67, 68; Ex. 1002 ¶¶ 167–169). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record.

g. Element 1[e]

Petitioner contends that Sharpe makes a clean file available to client systems by modifying its metadata to be “clean,” and Kamalapuram moves a file stored in an isolated area of network storage to a general storage area after it was scanned by a security analysis system such that a skilled artisan would have found it obvious to copy the clean file into a separate repository that is accessible to users as a more efficient way to manage access to files. Pet. 67–68 (citing Ex. 1006, 2:32–40, 6:62–7:3, 58:28, 96:61–66, 97:23–24, 97:56–58, 98:1–8, 98:28–30, 105:9–13; Ex. 1008 ¶ 122; Ex. 1002 ¶ 172).

Patent Owner argues that neither Sharpe nor Kamalapuram teaches “copying” a file but instead they describe “moving” files, and Petitioner fails to show how moving of files renders obvious the claimed copying. Prelim. Resp. 31–32. For the reasons discussed for this limitation in Ground 1, we find that Sharpe’s serving of files to clients and Kamalapuram’s moving of clean files from an isolated area to a general area satisfies this limitation.

On this record, we find that Petitioner has established a reasonable likelihood that claim 1 would have been unpatentable over Sharpe, Seese, and Kamalapuram.

5. Independent Claims 13 and 21

Petitioner provides a detailed mapping to the limitations of claims 13 and 21. Pet. 68–71. These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record. *See* Prelim. Resp. 31–40.

On this record, we find that Petitioner has established a reasonable likelihood that claims 13 and 21 would have been unpatentable over Sharpe, Seese, and Kamalapuram.

6. *Dependent Claims 2, 4, 5, 7–10, 12, 14, 16, 18–20, 22, 24, and 26–28*

Petitioner provides detailed contentions and a detailed mapping of how Sharpe, Seese, and Kamalapuram teach or suggest the limitations of claims 2, 4, 5, 7–10, 12, 14, 16, 18–20, 22, 24, and 26–28 with supporting testimony of Dr. Franz. Pet. 71–79 (citing Ex. 1002 ¶¶ 188–208). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record. On this record, we find that Petitioner has shown a reasonable likelihood that claims 2, 4, 5, 7–10, 12, 14, 16, 18–20, 22, 24, and 26–28 would have been unpatentable over Sharpe, Seese, and Kamalapuram.

F. *Ground 3: Alleged Obviousness Over Sharpe, Seese, Kamalapuram, and Dods*

Petitioner asserts that claims 3, 6, 11, 15, 17, 23, and 25 are unpatentable under 35 U.S.C. § 103 over Sharpe, Seese, Kamalapuram, and Dods. Pet. 2–3, 79–81.

1. *Dods (Ex. 1009)*

Dods discloses a security platform that uses behavior-based analysis to detect malware in files. Ex. 1009, 10:16–26.

2. *Motivation to Combine*

Petitioner contends that a skilled artisan would have been motivated to combine Dods’ teachings with Sharpe, Seese, and Kamalapuram to detect malware in files by using additional mechanisms to detect malware in a file that were taught by Dods, namely a behavior-based analysis, and were not taught by the other references to improve the overall malware detection capabilities of the Sharpe-Seese-Kamalapuram combination. Pet. 79–80.

3. *Dependent Claims 3, 6, 11, 15, 17, 23, and 25*

Petitioner provides detailed contentions with a detailed analysis of how Sharpe, Seese, Kamalapuram, and Dods teach or suggest the limitations of dependent claims 3, 6, 11, 15, 17, 23, and 25 with supporting testimony of Dr. Franz. Pet. 79–81 (citing Ex. 1002 ¶¶ 210–216). These contentions, which Patent Owner does not contest at this stage, are supported sufficiently by evidence of record. On this record, we find that Petitioner has shown a reasonable likelihood that claims 3, 6, 11, 15, 17, 23, and 25 would have been unpatentable over Sharpe, Seese, Kamalapuram, and Dods.

III. CONCLUSION

For the reasons discussed above, we are persuaded that Petitioner has demonstrated a reasonable likelihood of prevailing in establishing that at least one of claims 1–28 of the '856 patent is unpatentable.

IV. ORDER

In consideration of the foregoing, it is:

ORDERED that an *inter partes* review of the '856 patent is hereby instituted as to claims 1–28 on all grounds set forth in the Petition; and

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial on the grounds of unpatentability authorized above; the trial commences on the entry date of this decision.

IPR2023-00456
Patent 11,036,856 B2

PETITIONER:

Thomas Millikan
Babak Tehranchi
Kristen Schunter
PERKINS COIE LLP
millikan-ptab@perkinscoie.com
tehranchi-ptab@perkinscoie.com
schunter-ptab@perkinscoie.com

PATENT OWNER:

James Glass
John McKee
Quincy Lu
Andrew Holmes
Ognjen Zivojnovic
QUINN EMANUEL URQUHART & SULLIVAN LLP
jimglass@quinnemanuel.com
johnmckee@quinnemanuel.com
quincylu@quinnemanuel.com
drewholmes@quinnemanuel.com
ogizivojnovic@quinnemanuel.com