IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,

Petitioner,

v.

AVAGO TECHNOLOGIES INTERNATIONAL SALES PTE. LIMITED,

Patent Owner.

Case No. IPR2020-01582

U.S. Patent 8,572,138

PETITIONER'S REPLY

TABLE OF CONTENTS

Virtual machines are mentioned as an afterthought in the '138 specification. The first 31 columns describe deployment of non-virtualized application images. Only the last 5 columns mention virtual machines, applying earlier-described concepts to VMs using off-the-shelf software from VMware. The specification does not identify any purported points of novelty regarding its use of VMs or VM images.

VMware prior art renders the challenged claims obvious. Khandekar and Le —both VMware patents—teach all limitations. PO attempts to defeat obviousness by attacking Khandekar alone. However, PO does not address the *combination* of Khandekar and Le (Ground 2). PO did not dispute the motivation to combine and devoted a single page attacking Le alone, without ever addressing the combination. Therefore, Ground 2 presents entirely unrebutted evidence of obviousness.

PO's arguments rely on a clearly improper claim construction. PO does not identify a single claim term that requires construction. Instead, PO asks the Board to import an entire three-stage procedure into the claims, contradicting bedrock claim construction principles and Federal Circuit case law. PO's construction also contradicts its position in district court, where it declared that "no construction [is] necessary" for the challenged claims.

I. Khandekar and Le Teach the Same VMs as the '138 Patent.

The '138 specification does not purport to invent new VMs or VMMs. Instead, the specification discloses only one VMM, commercial VMware ESX software (Ex-1001, 32:59-64, 33:25-41; Ex-1024, 42:25-43:22, 44:5-11, 84:9-85:3), which it uses to provide VMs (Ex-1001, 33:39-41; Ex-1024, 47:20-48:14, 85:5-17, 48:23-49:9). PO's expert admits that VMs were known and the '138 patent "doesn't disclose how to, for example, design and implement a new kind of virtual machine manager." Ex-1024, 43:23-44:4, 28:16-25, 47:20-48:14, 85:5-17. He further admits the specification does not teach how VMs are created or how images are captured; it only states that it is done using existing VMware software. Ex-1024, 49:11-16, 47:9-48:14, 48:23-49:9, 45:11-46:13, 58:12-60:3.[1]

PO attempts to argue that the '138 patent somehow has a different kind of VM than Khandekar. PO's expert argues that Khandekar's VMM is not capable of creating a VM in the same manner as the '138 patent. Ex-2001 ¶131. However, PO and its expert do not address the fact that ***the '138 patent, Khandekar, and Le all use VMs provided by VMware software***.

[1] The only exemplary VM disclosed by the '138 specification emulates an x86 machine (Ex-1001, 33:11-15), which was known and obvious (Pet., 63-64).

Khandekar and Le are both VMware references that teach VMware software for running VMs. *See* Ex-1004, 13:17-35; Ex-1005, 65:65-66:10, 13:22-30. Khandekar uses existing VMware software to "load and run multiple VMs." Ex-1004, 13:17-35. PO's argument—that the '138 patent purportedly teaches a different type of VM than VMware's own patents—is contradicted by the prior art and the '138 patent itself. Ex-1029 ¶¶3-6.

Since Khandekar and Le both teach VMware software, Petitioner explained the obviousness of using VMware software including ESX. Ex-1003 ¶¶59, 62, 118. PO's expert admits a POSITA would have been able to "use … commercially available software from VMware such as ESX and GSX software." Ex-1024, 33:14-25, 44:5-11; Ex-1003 ¶35. Even if the '138 patent required a special type of VM provided by VMware software, Khandekar and Le teach using that same VMware software.

II. PO's Construction Is Improper.

Claim construction requires identifying a term that requires construction—specifically, it must identify "a textual reference in the actual language of the claim with which to associate a proffered claim construction." *MBO Laboratories, Inc. v. Becton, Dickinson & Co.*, 474 F.3d 1323, 1330-31 (Fed. Cir. 2007); *NTP, Inc. v. Research in Motion, Ltd.*, 418 F.3d 1282, 1310 (Fed. Cir. 2005) ("[A] party wishing to use statements in the written description to confine or

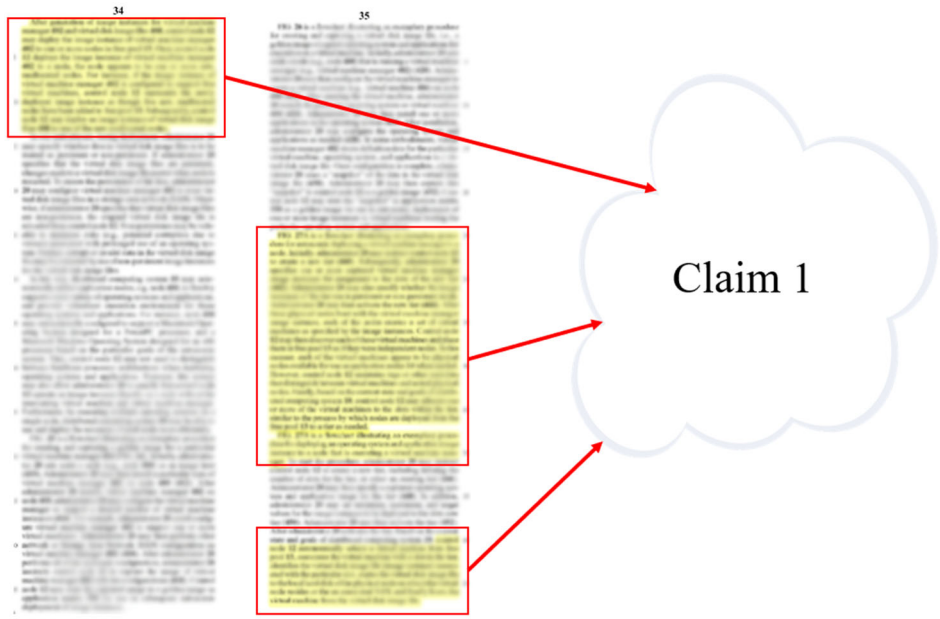
otherwise affect a patent's scope must, at the very least, point to a term or terms in the claim with which to draw in those statements."). As a threshold issue, PO never identified any claim language that requires construction. Therefore, PO's construction should be rejected.

Rather than clarify the meaning of ambiguous claim language, PO asks the Board to literally import an entire three-stage process from the embodiments into the claims, including *system* claim 1. *See* POR, 32 ("Each of these independent Challenged Claims … requires the three-step deployment process … ***described in the specification***.");[2, 3] POR, 25:

---

[2] All emphasis/annotations added.

[3] To the extent PO argues a two-step capture process is required, both experts agree the claims do not require it, and that Figs. 24-26 are exemplary. Ex-2002, 41:5-8, 42:7-9, 56:19-57:1, 57:10-15, 57:21-58:9; Ex-1024, 77:10-18, 53:9-54:4, 67:8-68:16, 69:1-70:8.



This is improper. PO does not point to any words or expressions of manifest exclusion or restriction to justify importing an entire process from the specification. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906-08 (Fed. Cir. 2004); *i4i Ltd. P'ship v. Microsoft Corp.*, 598 F.3d 831, 843-844 (Fed. Cir. 2010), aff'd, 564 U.S. 91, (2011).

Moreover, while PO now insists that the claims must be construed to require a three-stage deployment process, PO took a contrary position in district court, just one month before it filed the POR. PO did not propose any terms for construction—indeed, PO said "***no construction necessary***" for any terms of the '138 patent. Ex-1025, 15-18; Ex-1026, 13-14; Ex-1027, 19-20.

A. PO injects method steps into system claim 1, contrary to the claims and its own admissions.

PO's construction requires, e.g., actual execution of VMM images at application nodes. However, Claim 1 is a system claim that focuses on *capabilities*—specifically, those of the control node. It is not a method claim, does not recite any steps, and deliberately avoids limitations that require actions to be taken by the application nodes. For co-pending IPR2021-00542, the Board held that claim 1 only requires each virtual machine to be "'operable to provide' an environment that emulates a computer platform" but it "do[es] not recite or require that the software is executed and actually creates the emulated computer platform." Paper 11 (PTAB Sep. 7, 2021).

Claim 1 recites "a software image repository … operable to store (i) a plurality of image instances of a virtual machine manager …." The claim only requires those images to be "*executable* on a plurality of application nodes"—not that the images are actually executed. The VMM images need only be capable of providing a plurality of VMs "when executed" (Limitation [1d]). *See Intel Corp. v. U.S. Int'l Trade Comm'n*, 946 F.2d 821, 832 (Fed. Cir. 1991) (because the claim states "whereby *when* said alternate addressing mode is selected[,]" the infringing device "need only be capable of operating in the page mode … actual page mode operation in the accused device is not required.") (emphasis in original).

Similarly, Limitation [1e] requires image instances of software applications to be "executable"—but actual execution is not required. Limitations [1f] and [1g] recite "a control node that comprises an automation infrastructure to provide autonomic deployment"—referring to a capability rather than actual deployment. Ex-1029 ¶¶8-9.

PO's expert admits claim 1 does not require the VMM to be executing. Ex-1024, 41:13-22:

> Q Does claim 1 require the virtual machine manager to actually be executing on an application node?
>
> A ***It doesn't require anything to be executed***. It describes what happens ***when*** the virtual machine manager is executed.

Dr. Rosenblum further admits claim 1 does not require booting the software application images on nodes.[4] Ex-1024, 40:22-41:3, 41:5-11.

Whereas claim 1 recites capabilities, PO's three-step process requires actual deployment and execution (POR, 40), contradicting PO's own briefing:

---

[4] Dr. Rosenblum's testimony contradicts his declaration, which attempted to define an image instance as "a copy of the image *deployed* onto a *particular* virtual machine." Ex-2001 ¶32. Claim 1, however, recites image instances stored *before* deployment in "a software image repository."

PO's PTAB Arguments	PO's District Court Briefing
Claim 1 requires "a virtual machine manager image instance ***to execute*** on an application node" and "***provides the virtual machine*** on the node." POR, 37-38, 25. "Claim 1 requires that the virtual machine manager ***must actually manage one more virtual machines***…" Ex-1024, 75:15-77:4.	The "virtual machine manager" is "'a software program that is ***capable*** of managing one or more virtual machines,' ***not a program that actually does so***." Ex-1027, 19.[5]

PO admits "it is hardly uncommon for an apparatus or system claim, as a claim to ***a product rather than a process (or a forbidden mix)***, to be directed to ***capability, instead of actual operation***." Ex-1027, 19. Claim 1 is directed to a system, rather than a process. By injecting a three-stage process into the system, PO's construction would convert claim 1 into a "forbidden mix" of system and method claims that, by PO's admission, claim 1 seeks to avoid. *See id*.

[5] These arguments arose from a proposed simplification to avoid ambiguity for a jury. Netflix explained that a construction was appropriate with or without the simplification. Ex-1028, n.10.

B. PO's construction is contrary to the intrinsic record.

PO argues it is necessary to import PO's procedure as a "matter of both logic and grammar" to define how software application images are deployed. POR, 2, 33. However, PO does not address the fact that (i) the claim *already provides an express definition* for deploying images, and (ii) PO's supposed "logic" is contrary to claim 1.

First, Claim 1 includes an express definition that leaves no uncertainty as to how deployment is provided: autonomic deployment is provided "by causing the plurality of image instances … to be copied from the software image repository to the application nodes."[6] This "express definitional language in the claim itself is complete and unambiguous, and it cannot be overridden by a description of exemplary embodiments in the Specification." *See GeneriCo, LLC v. Dr. Falk Pharma GmbH*, IPR2016-00297, Paper 55, 10-11 (PTAB May 19, 2017); *Straight Path IP Grp., Inc. v. Sipnet EU S.R.O.*, 806 F.3d 1356, 1361 (Fed. Cir. 2015).

[6] This definitional language ("by causing … to be copied … to the application nodes") is recited for both limitations [1f] and [1g], for providing autonomic deployment of VMM and software application images. It was added in the RCE amendment that led to allowance. Ex-1002, 000034.

Ignoring this express definition, PO relies instead on descriptions of Figures 27A and 27B. These, however, are "exemplary procedure[s]."[7] Ex-1001, 35:25-27, 35:48-51; Ex-1024, 63:23-64:17. PO repeatedly conflates examples in the specification with requirements of the claims, for example citing to Dr. Shenoy's discussion of examples/embodiments to argue for new claim limitations. *See* POR, 28; Ex-2002, 63:10-64:6 ("So in this particular example …"), 71:6-72:6 ("So this is an example procedure … So in this example …"), 73:2-74:3 ("In this example …"). Dr. Shenoy never testified that these examples were required by the claims. Ex-1029 ¶7.

PO relies on exemplary embodiments without pointing to any lexicography or disavowal to override the express definition in the claims. *See Trustees of Columbia Univ. in City of New York v. Symantec Corp.,* 811 F.3d 1359, 1364 (Fed. Cir. 2016) ("Absent implied or explicit lexicography or disavowal, we have recognized that the specification plays a more limited role where claim language has so 'plain a meaning on an issue' that it 'leav[es] no genuine uncertainties on interpretive questions relevant to the case.'").

Moreover, PO asks the Board to import cherry-picked features from Figures 27A and 27B into the claims. PO's expert admitted that "the challenged claims do

---

[7] Figure 24 is also exemplary. Ex-1001, 32:51-55; Ex-1024, 77:10-18.

not require all the steps described for 27B" or 27A. Ex-1024, 66:7-24. While PO argues that a VMM is executing in these examples, most of Figure 27B relates to the creation of tiers and assignment of VMs to slots in tiers. *See* Ex-1001, 35:51-53 (administrator "may" create tiers); Ex-1024, 64:18-66:5 (use of tiers not required). The procedures also include "*cop[ying] the virtual disk image file to the local hard disk* of the physical node" (Ex-1001, 35:59-67) and then "boot[ing] the virtual machine from the virtual disk image file" (Ex-1001, 35:59-67). PO's expert admitted that claim 1 does not require booting the VM.[8] Ex-1024, 41:7-11, 52:7-53:3.

There is no need to guess which aspect of these procedures is required by the claims. Each independent claim defines autonomic deployment as being provided by copying the image to an application node. Ex-1001, 36:42-45, 36:47-49, 38:14-16, 38:19-21, 39:9-11, 39:14-15. That an exemplary procedure of Figure 27B also describes an executing VMM is evidence ***against*** PO's construction because the Applicant knew how to describe the feature but specifically omitted it from the challenged claims. It would be improper to import it now under the guise of claim construction.

---

[8] The specification only discloses one way to load a software application image onto a VM: by booting the image. Ex-1001, 35:59-67, 33:35-41.

Second, PO relies on "logic" and antecedent basis, but its arguments are contrary to express claim language. Claim 1 recites a "software image repository" that stores (i) VMM images and (ii) software application images. The VMM images have the capability, when executed, to "provide a plurality of virtual machines." The repository also stores "image instances of a plurality of software applications that are executable on ***the*** plurality of virtual machines," which in this context simply refers to the VMs that the VMM images are capable of creating.[9] Ex-1029 ¶10.

If "***the*** plurality of virtual machines" referred to specific VMs at application nodes, as PO contends, then the recited software application images cannot exist and cannot be stored in a repository until after VMM images have been deployed and executed on application nodes. PO's arguments impose deployment and execution as prerequisites before the software image repository can store images as claimed. PO cannot be correct—indeed, PO's expert admitted that a "software image repository" can exist, as recited in claim 1, without a control node to provide autonomic deployment of images. Ex-1024, 82:2-17.

---

[9] This is consistent with the specification. Ex-1001, 32:59-64 ("As used herein, a virtual machine manager is a software program that is *capable* of managing one or more virtual machines."); Ex-1027, 19.

Moreover, when applied to claim 1, PO's supposed "logic" creates a paradox. As explained above, under PO's logic the software image repository cannot store images, as recited in claim 1, until after the control node has deployed VMM images to application nodes. But claim 1 also requires the control node to "provide[s] autonomic deployment" of the VMM images by causing them "to be copied ***from the software image repository*** to the application nodes." PO's logic creates self-contradictions within claim 1 and should be rejected for that reason alone. Ex-1029 ¶11.

PO is further contradicted by the specification, which states that images are created before deployment, and that the software images are "bootable on the virtual machines" before deployment. Ex-1001, 33:55-34:5. Dr. Rosenblum admits that "[a]fter generation of … both the image instances … control node may deploy the image instances of the virtual machine manager." Ex-1024, 62:1-63:22.

In short, PO's construction should be rejected because it contradicts the claims and specification. Ex-1029 ¶12.

C. PO's construction is contrary to claims 19 and 26.

Claim 19 shares similar language to claim 1, and many of the concepts explained above apply here. Like claim 1, claim 19 includes an express definition for how autonomic deployment is performed: "autonomically deploy the image

instances … by causing the image instances … to be ***copied to the application nodes***." PO ignores this definition. *See supra* §II.B.

Claim 19 also stores software application images that are "executable on ***the*** plurality of virtual machines." As explained above, PO's "logic" creates paradoxes that are contrary to claims 1 and 19. *See supra* §II.B.

Claim 19 is directed to steps performed at the control node and does not recite any actions taken by the application nodes. By requiring execution of images at application nodes, PO injects the actions that claim 19 seeks to avoid, contradicting the structure and scope of claim 19 as written.

PO's arguments are further contrary to the claims and specification. Claim 19 recites "executing a rules engine" for autonomic deployment. The specification explains that the rules engine uses forward-chaining rules and an application matrix for deploying applications. Ex-1001, 27:10-19. The rules engine may invoke methods for "cloning, capturing, importing, exporting or upgrading software images." Ex-1001, 24:12-21. But claim 19 does not require, and the specification does not mention, the rules engine executing any images at application nodes. *See supra* n.8. There is no mention of a rules engine in the portions of the specification that describes VMMs and VMs (starting col. 32:51), and therefore there is no written description of a rules engine causing a VMM or VM image to be executed/booted at the application nodes.

Applied to claim 19, PO's three-stage deployment process would require the rules engine, running at the control node, to somehow execute VMM images at the application nodes. As explained above, this lacks written description and is contrary to the structure and scope of claim 19.

Claim 26 recites a computer-readable medium executed by a processor to perform the steps in claim 19. Therefore, the same arguments apply to claim 26.

D. "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines"

PO admits that "virtual disk image files" can satisfy this limitation (POR, 43; Ex-1024, 81:8-23, 39:2-20), but argue that Petitioner's construction reads out PO's three-stage deployment process. It is the claim language, not Petitioner, that omits PO's process by requiring that image instances need only be "executable"—not actually executed.

III. PO's Construction Is Satisfied by Grounds 1 and 2.

Even if PO's construction is adopted, it is satisfied by Grounds 1-2.

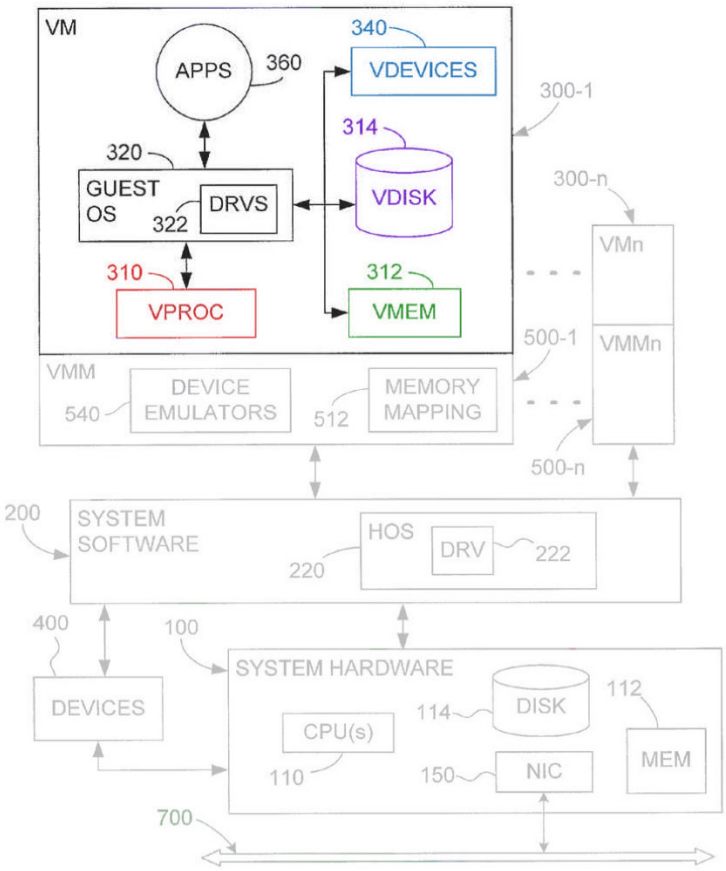
A. Khandekar teaches PO's three-stage deployment process.

PO's construction requires executing VMM images on nodes to provide a plurality of VMs. Khandekar teaches this. Ex-1029 ¶¶13-14. For example, Khandekar teaches deploying VMM images to hosts to allow them to run VMs. Pet. 49-50; Ex-1004, 8:36-44, 13:17-35, 4:66-5:8. PO's expert admits a POSITA would know how to use VMware software to create VMs. Ex-1024, 46:14-47:8,

49:17-50:2. He further admits that, once an image instance of the VMM (VMware ESX) is executed on the node, it creates the VM. Ex-2001 ¶69; Ex-1024, 45:3-10.

Khandekar's Fig. 1 teaches using VMM software to provide VMs containing the same components that PO's expert contend are present in the supposedly bare-metal VMs of the '138 patent: "a virtual processor," "a virtual memory," "virtual devices," and a "virtual disk." Ex-1024, 50:3-52:6; Ex-1004, Fig. 1:

FIG. 1
(Prior Art)



PO's construction requires the VMM image to be executed. Khandekar teaches deploying and executing VMM images at the hosts. Dr. Shenoy explained

how Khandekar teaches that "***[t]he image instances of the VMM are executed on the host computer***, allowing it to run a plurality of VMs" (Ex-1003 ¶¶113, 115), citing and quoting, e.g., Ex-1004, 7:39-47 ("… the VM is structured and *performs* as a complete 'real' computer system, …"), 6:23-43 ("[A] virtual machine (VM)… is a software abstraction—a 'virtualization'—of an actual physical computer system. ***Actual execution of instructions*** for the VM is ultimately carried out on the system hardware 100…"). Dr. Shenoy further explained that "the VMMs manage the execution of the VMs." Ex-1003 ¶92 (citing Ex-1004, 6:62-7:27).

Khandekar teaches deploying VM images on hosts (Pet., 55-59; Ex-1004, 8:64-67); when executed, the booted VM images run on the VMs. For example, Khandekar teaches that "[w]hen the provisioning engine 810 ***deploys*** a VM on a host, such as host 1000, ***the VM is powered on and its guest OS is booted***." Ex-1004, 19:33-45, 13:36-40 ("Once a VM is deployed on a host, it may be started ('powered on') in any conventional manner …."). For images of pre-built and custom-built VMs, Khandekar teaches copying the virtual disk image file to the host and booting the virtual machine with the image. Ex-1029 ¶¶15-16; Ex-1003 ¶160; Ex-1004, 21:25-64:

…

8) Copy the VM's config file and the disk files to the host.

…

11) Call the API in the host ***to power on the VM***. …

…

See also Ex-1004, 15:29-17:29, 18:16-23, 23:28-51.

Similarly, Khandekar teaches deploying images of Instantly-Deployable VMs ("IDVM") by coping and resuming them on the host. Ex-1004, 17:30-50; Pet., 46; Ex-1003 ¶135. "The provisioning server then selects an appropriate host … copies the necessary data on that host, and ***resumes the execution of the VM*** on the new host." Ex-1004, 11:20-35, 21:65-22:13:

…

4) Copy the VM's config file 842, REDO logs 843 for the disks 841 and suspend-to-disk image file 844 to the host.

…

7) Call the API in the host ***to resume the VM***.

…

Pet. 59-60; Ex-1003 ¶162; Ex-1029 ¶17.

In summary, Khandekar teaches deploying VMM images to hosts and using the VMM to execute VM images on the hosts. Ex-1004, 8:36-44, 13:17-35, 4:66-5:8, Fig. 1. PO's expert admitted that VM images cannot execute unless a VMM image is already running and providing the VMs. Ex-1024, 36:12-37:18, 54:5-55:1, 78:25-79:9. He further admitted that for a VM to resume on a node, a VMM must be executing on that node. Ex-1024, 75:1-10. Therefore, Khandekar teaches

deploying VMM images to hosts, executing the VMM images, and then deploying and executing VMs on hosts. Ex-1029 ¶¶18-19.

B. The combination of Khandekar and Le teaches PO's three-stage deployment process.

PO attacks Khandekar alone but did not address the combination of Khandekar with Le. Petitioner and Dr. Shenoy explained how the combination teaches PO's three-stage deployment process by deploying the *physical* image of a host computer (including the VMM) using Le's teachings, booting the host computer, and then deploying the *virtual* image of a VM to the host using Khandekar's teachings. This evidence stands unrebutted.

Ground 2 uses Le's teachings to initialize or reimage physical host computers. Pet., 18, 51-55; Ex-1003 ¶¶58-60, 62, 149-153. Le teaches techniques for writing a physical image onto the host's hard drive, which included the host OS. Ex-1005, 54:14-21, 10:55-62, 2:19-21, Fig. 5; Pet., 18, 52; Ex-1003 ¶¶58, 149. There is no dispute that the OS is the VMM for Type 1 hypervisors such as VMware ESX. Ex-1003 ¶58; Ex-1024, 88:15-90:12, 30:7-31:25, 42:9-24; Ex-1005, 13:12-21, 66:2-10.

The last step of Le's imaging process "*reboots* the destination computer 1500 to allow it to *load the new operating system* deployed from the image 2015." Ex-1005, 27:5-9, 66:53-64 ("[A] physical computer typically reboots and becomes operational once it has been deployed from an image."), Fig. 5; Ex-1003 ¶151. For

the combination of Khandekar and Le, the OS is the VMM. Therefore, the combination teaches deploying a VMM image to a remote host and then executing the VMM on the host by booting the hypervisor OS.

As PO's expert admits, execution of the VMM provides a VM. Ex-1024, 45:3-10, 85:5-17. Therefore, the combination teaches deploying a VMM image to a node and then executing the image at the node to provide VMs. Ex-1029 ¶¶20-21.

After deploying the physical VMM image, Ground 2 used Khandekar's teachings to deploy virtual images of VMs, including virtual disk image files. Ex-1003 ¶¶168, 155-163. The combination uses separate images for VMMs and VMs because VMM images are images of *physical* disks while VM images include images of *virtual* disks. Ex-1003 ¶¶58, 81, 151; Ex-1004, 4:66-5:8, 7:7-38, 13:17-35; Ex-1005, 26:52-27:9, 44:4-17, 66:33-47. VM images were deployed to hosts as a separate, subsequent step because the physical imaging process overwrites the entire hard drive, leaving only the hypervisor OS. Ex-1005, 27:1-5 ("[T]he imaging server … format[s] the destination disk 2230, destroying whatever partitions, file systems, data, etc., that was previously present on it."); Ex-1003 ¶151; Ex-1029 ¶¶22-23. Khandekar teaches copying and then executing VM images on hosts. *See supra* §II.A.

Le's UCMS embodiment further teaches PO's three-stage deployment process. In his opening declaration, Dr. Shenoy explained how Le teaches that VMMs "are ***executed on the host computers***, where the image instances of the each virtual machine monitor and virtual machine manager provide a plurality of virtual machines, with each of the plurality of virtual machine operable to provide an environment that emulates a computer platform." Ex-1003 ¶119 (citing general teachings and UCMS embodiment). Dr. Shenoy further explained the motivation to apply Le's UCMS teachings to Khandekar's provisioning server. Ex-1003 ¶63; Pet., 18, n.3; Ex-1029 ¶24. PO did not dispute or address any of this testimony.

Le teaches that VMMs are running and managing VMs at hosts. For example, Le teaches the server deploying VM images (e.g., template image 4020) to virtual machines. Ex-1005, 44:4-17 (describing deployment steps), 66:33-47. "Once a virtual machine is successfully deployed from a template image 4020, ***the virtual machine manager 6200 may optionally power it on***" or wait to power it on at a later time. Ex-1005, 66:53-64; Pet. 32; Ex-1003 ¶96. Le includes numerous teachings where VMs are powered on. *See* Ex-1005 65:34-50, Fig. 8 (illustrating a typical VM product with a VM being "powered-on"); Ex-1029 ¶¶25-27. "***When a virtual machine is powered on, a virtual machine monitor program 6300 controls it and manages the interactions*** …" Ex-1005, 65:65-66:10.

In short, the combination of Khandekar and Le teaches PO's three-stage deployment process, including the execution of VMM images at host computers. Ex-1029 ¶28.

IV. Grounds 1-2 teach all limitations.

PO attacks Khandekar alone and then devotes a single page to attacking Le alone (POR, 64-65), without ever addressing combination of Khandekar and Le. This is improper because obviousness cannot be overcome by attacking the references individually. *In re Merck & Co*., 800 F.2d 1091, 1097 (Fed. Cir. 1986); *Bradium Techs. LLC v. Iancu*, 923 F.3d 1032, 1050 (Fed. Cir. 2019). PO chose to avoid the combination in its POR and should not be allowed to raise new arguments for the first time on Sur-Reply. The unrebutted evidence of Ground 2 invalidates the challenged claims.

Furthermore, PO's attacks against Khandekar rely on misrepresentation of its teachings. As explained below, PO's arguments have no basis, and therefore Khandekar alone (Ground 1) renders the challenged claims obvious.

A. Grounds 1-2 teach separate VMM images.

The combination of Ground 2 teaches separate VMM and VM images—*physical* disk images for the VMM (to image the physical hard disk of hosts) and *virtual* disk images for the VM. *See supra* §III.B; Ex-1003 ¶¶99-110, 122-138.

PO did not dispute or address this combination; therefore, the unrebutted evidence for Ground 2 teaches separate images.

For Ground 1, Khandekar by itself teaches storing separate images for VMMs and VMs. Khandekar teaches a VMM may be already installed on the host or installed with the VM. That there is a choice to install these together or separately teaches or suggests they are separate images. Ex-1029 ¶33. Khandekar also teaches that one VMM may support multiple VMs (Ex-1004, 7:28-38), which obviously meant the VMM image is separate from the multiple VM images. Petitioner explained the motivation to follow this teaching because it was the norm for VMware software by 2006. Pet., 16-17, 39; Ex-1003 ¶¶52-56, 114. PO did not dispute this.

Khandekar uses virtualization to allow the same VM image to run on a software abstraction—virtualized hardware—so that "[d]eployment of the VM is thereby made *substantially independent of the overall physical hardware configuration*." Ex-1004, 4:66-5:8, 13:17-35, 7:7-27. Dr. Rosenblum admits a VM is an abstraction layer that provides independence from underlying physical hardware. Ex-1024, 22:18-23:2, 29:2-30:6, 44:12-45:2. Therefore, using separate images is consistent with Khandekar's goals and teachings because it facilitates independence between the VM and underlying physical hardware, while a

monolithic image undermines Khandekar's goals by creating dependencies to physical hardware that otherwise would not exist. Ex-1029 ¶¶29-32.

Khandekar's use of separate images is consistent with the teachings of the prior art, where VMM images were stored separately from VMs. Ex-1003 ¶60; Ex-1014, ¶[0062]; Ex-1016, ¶¶[0043], [0049]. PO did not dispute this, and as Dr. Shenoy explains, this is how VM images had been used for years, including for VMware software, which had specific features for creating VM images. Ex-1029 ¶34.[10] PO's expert agrees. Ex-1024, 47:9-49:9. There was no reason why a POSITA would have created a monolithic image with both VM and VMM when a central purpose for using VMs, including Khandekar's use of VMs, was to allow flexibility in running them on different machines. Ex-1029 ¶35.

PO suggests the possibility of creating a monolithic image, and its expert opines that Khandekar requires "including the VMM *within the same image* as the VM." Ex-2001, ¶139. PO offers no evidence that this had ever been done and provides no reason why a POSITA would attempt it. Moreover, PO ignores the fact that the VMM image is a *physical* image while the VM image is a *virtual*

[10] Dr. Shenoy is deeply familiar with the POSITA's knowledge in 2006—he is the co-author of prior art ("B. Urgaonkar et al.") *cited by the applicant* on the face of the '138 patent. Ex-1002, 000233; Ex-1003, App. 1, 000193.

image. Ex-1004, 16:43-63, 17:30-50. PO offers no explanation of how the VMM—an OS or application on the *physical* host computer—could be included in the *virtual* disk image of a VM.

Khandekar does not disclose a monolithic image, and PO's arguments to the contrary mischaracterize Khandekar in three ways. First, PO's expert argues that "the included VMM *depends directly* on the VM." Ex-2001 ¶139. However, what Khandekar actually states is that the VMM must account for "the characteristics ***of the host*** on which the VM is to be installed[.]" Ex-1004, 13:25-28. The VMM image must match the physical hardware of the ***host***, not the VM. Ex-1029 ¶36. That is why it would have been obvious to store a plurality of VMM images to account for the different host hardware. Pet., 27-28.

Second, PO's expert argues that Khandekar's "VMMs and VMs are always to be considered in pairs" (Ex-2001 ¶140), but cites a description offered for "the sake of simplicity" (Ex-1004, 8:36-43). Moreover, Khandekar teaches that one VMM may support multiple VMs (Ex-1004, 7:28-38), which PO did not address or dispute.

Third, PO cites Khandekar's statement that "the VM and VMM can be considered to form a single cooperating system." Ex-1004, 7:40-42. This, however, refers to the execution of a VM on a VMM, explaining how the VMM provides a "transparent" interface to the host. *Id.* It has nothing to do with images.

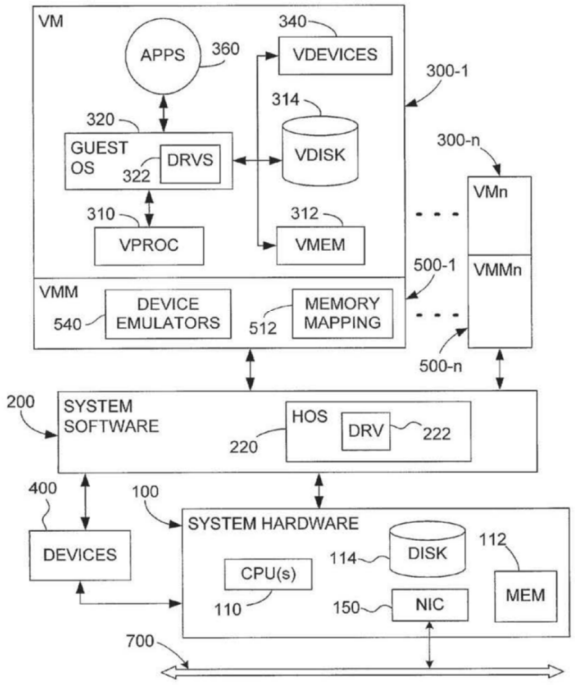
The unrebutted evidence confirms the use of separate images, while PO offers nothing more than mischaracterization.

B. Grounds 1-2 teach software applications images.

Khandekar and Le use the same types of VMs and VM images as the '138 patent—all use commercial VMware software. *See supra* §I. While Khandekar at times uses shorthand to refer to its VM images based on the type of VM they contain—pre-built, custom-built, and instantly-deployable VMs—Khandekar is clear that these are "stored" images of VMs including virtual disk image files. Ex-1004, 15:40-50, 16:43-52, 17:30-50, Figs. 3, 4.

PO argues that Petitioner relies on Khandekar's VMs as both software application images and virtual machines. POR, 60. Not so. For software application images (limitation [1e]), Petitioner relied on Khandekar's images of Pre-Built, Custom, and instantly-deployable VMs. Pet., 41-47. For virtual machines (limitation 1[d]), Petitioner relied on VMs provided by VMMs, including from VMware software. Pet., 37-41; Ex-1004, Fig. 1. Khandekar and Le teach using VMM software to run VMs. Pet., 37-41; Ex-1004, Fig. 1:

FIG. 1
(Prior Art)



Ex-1029 ¶¶4-5.

Khandekar's three types of VM images satisfy the agreed-upon construction of image instances: "a snapshot, or copy, of installed and configured software." Pet., 43-47; Ex-1003 ¶¶123, 126-136. PO's arguments are detached from the challenged claims.

Pre-built VMs. PO argues that images of pre-built VMs are "stored with their operating systems and applications installed" and contradict the '138 patent. POR, 52. However, the '138 patent allows this, as PO's expert admits. Ex-1001, 33:25-30, 33:55-67, 5:12-22; Ex-1024, 38:12-39:1, 39:21-40:21, 55:2-56:25.

Custom-built VMs. PO argues the custom-built VMs include installation files for additional software. POR, 52-53; *see* Ex-1004, 23:28-51. However, a

custom-built VM image has an OS installed (Ex-1004, 15:41-46) and therefore is a snapshot, or copy, of installed and configured software (the OS).

IDVM. PO argues the IDVM is a reallocation of an already-running VM. POR, 52. Khandekar states otherwise, teaching that the IDVM is a *suspended* image of a VM—not already running as PO contends. Ex-1004, 17:30-50. To create the image, the IDVM is powered-on, configured, and then suspended to an image for later deployment to hosts. Ex-1004, 17:30-50. The IDVM image is *copied* and resumed on hosts. Ex-1004, 11:20-35, 21:65-22:13. This is similar to PO's 2-stage image creation process, and nothing in the '138 patent precludes the image instances of software applications from being snapshots of already- or previously-running VMs, as PO's expert admits. Ex-1024, 57:17-58:3.

C. Grounds 1-2 teach known VMMs.

Khandekar's embodiments rely on known VMware software and VMMs. Ex-1004, 6:63-7:55 ("[G]eneral features of VMMs are known in the art and are therefore not discussed in detail"). Petitioner explained why Khandekar's figures and embodiments are meant to be used together. Ex-1003 ¶¶44-50. PO did not dispute any of this; its expert admitted "one could take that description of background IP as part of the teachings." Ex-1024, 91:22-92:13, 34:17-24. Le further teaches using known VMware software, which PO did not dispute. Pet., 33, 36.

PO argues that limitation 1[d] relies on Khandekar Fig. 1, which is prior art. However, "[a] reference must be considered for everything that it teaches, not simply the described invention or a preferred embodiment." *CRFD Research, Inc. v. Matal*, 876 F.3d 1330, 1348-49 (Fed. Cir. 2017) (citing *In re Applied Materials, Inc.*, 692 F.3d 1289, 1298 (Fed. Cir. 2012)); *EWP Corp. v. Reliance Universal Inc.*, 755 F.2d 898, 907 (Fed. Cir. 1985).

V. Conclusion

For the reasons explained above, Petitioner respectfully requests that all challenged claims be held unpatentable.

Date: September 28, 2021

Respectfully,
/*Harper Batts*/
Harper Batts (Reg. No. 56,160)

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Reply to Patent Owner's Response contains 5,595 words and therefore complies with the 5,600 word type-volume limit specified by 37 C.F.R. § 42.24(c)(1). The word count does not include those portions excepted by 37 C.F.R. § 42.24(a)(1) and was calculated by Microsoft Word 365.

Date: September 28, 2021

/*Harper Batts*/
Harper Batts (Reg. No. 56,160)

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on September 28, 2021, a complete copy of the Petitioner's Reply was filed electronically and served via email to all parties to this proceeding at the addresses indicated:

Daniel S. Young
Registration No. 48,277
dyoung@adseroip.com
dyoung@sbiplaw.com

Chad E. King
Registration No. 44,187
chad@adseroip.com
cking@sbiplaw.com

ADSERO IP LLC d/b/a
SWANSON & BRATSCHUN LLC
8210 Southpark Terrace
Littleton, CO 80120

Date: September 28, 2021

/*Harper Batts*/
Harper Batts (Reg. No. 56,160)
Attorney for Petitioner